CMS System Identification Based on Improved Algorithm of Apriori

Zhonglin Liu\(^1,2\) and Yan Wang\(^1\)*

\(^1\) Information Security Institute, Sichuan University, Chengdu, 610065, China
\(^2\) Sichuan Key Laboratory of Information Security, 610041, China
Email: yanwang@stu.scu.edu.cn

Abstract. Web applications on the Internet are often attacked by various vulnerabilities. Once various vulnerabilities are published, it becomes more and more important to quickly and accurately locate the affected Web application or system from the list of website resources. In this paper, based on the data and information of similar CMS websites, the improved Apriori algorithm is used to extract the features of CMS websites and their association rules, which can be used to quickly identify whether unknown websites are CMS templates of this type. Through experimental tests, four feature information used to identify WordPress is obtained. According to these features and relevance, the system type website can be identified more accurately.

1. Introduction

With the rapid development of the network information age, the use of the Internet in various industries around the world is becoming more and more common. According to the statistics of IntelnetLiveStats website, from 2016 to 2018, the number of websites worldwide has exceeded 1.7 billion [1]. By the end of 2019, the number of websites that directly use or modify the adjusted CMS (Content Management System) system to build websites has exceeded 1.2 billion, which shows that CMS system is widely used in the construction of websites [2]. CMS is a content management system, and it has much excellent design based on the template, which can accelerate the speed of website development and reduce the cost of development. Its function is not limited to text processing and can also deal with pictures, Flash animation, audio, and video streaming, Image even e-mail files and so on. This paper studied the feature’s identification of specific types of CMS systems, which can be used for similar CMS system identification. In addition, it can be used to collect the global web information, to test the known flaw detection of a specific version, and to make the website information management more convenient.

The CMS has more than 1000 merchants worldwide, which have different templates and different functions [3]. Some are open source, and some are not. It is difficult to obtain the complete version feature of CMS system only from the direct observation and analysis of the response packet's header which comes from the source of each site analysis, not to mention how time-consuming to obtain features of the law from a large number of sites. At present, the general method for identification of CMS system is to check whether it contains the keyword of CMS version name through the regulation match in the source code. However, this method causes a high error rate [4]. Considering lots of the pages do not appear to contain the keyword of CMS version name, this paper presents a CMS system identification method based on an improved Apriori algorithm. We obtain the fingerprint features of the CMS system from CMS site data through the automatically mined of earning algorithm. The fingerprint features can be used to recognize the version of the CMS site.
The rest of the paper is organized as follows. Related work is presented in section 2. In section 3, we give a detailed description of the CMS system identification based on the improved algorithm of Apriori. In section 4, we conduct the experiments and evaluation results. Finally, we summarize our work and discuss further work in section 5.

2. Related Work
With the development of information security technology, in recent years, there are countless attacks on Web systems on the Internet. Almost every day, there are attacks on Web systems or applications in the Web security circle. In order to discover and Alert these affected systems and applications, it becomes very important to quickly identify and locate these affected systems and applications [5]. Wang [6] proposed an identification technology based on traffic analysis. Through the analysis of web page traffic data, mining features, and giving different weights to them, a website fingerprint identification model based on the KNN algorithm is finally established. However, this method is difficult to implement in the actual environment and costs too much on the premise that a large number of ordinary users can obtain online traffic. In the discovery and verification of large-scale Web vulnerabilities proposed by An [7], the identification of special types of websites only uses crawlers and Nmap to obtain server port information and page return header information as features. This method can only have a better identification rate under specific circumstances. Wu [8] proposed that HTTP, SSL, and other protocols may have information disclosure. Fingerprint identification of websites is realized through fingerprint feature extraction based on request/response pairs. This method has considerable limitations in CMS identification. Different CMS systems do not have features at the protocol level, and their features may exist in HTML pages.

In summary, there is still a certain gap between the research and the actual application of the identification method for the website CMS system in terms of effect and cost. Therefore, this paper proposes an identification method with low cost and high accuracy to solve this problem.

2.1. CMS System Identification Method: Overview
The CMS recognition system mainly includes the feature extraction module and the feature recognition module. As shown in the process shown in figure 1, the sample information obtained by the crawler is subjected to feature selection and calculation to obtain the feature information of the CMS system. Finally, the recognition of the CMS is realized. It proceeds as follows:

![Figure 1. Overview.](image-url)
(a) Obtain the response data of the website through the Scrapy crawler framework, use Beautiful Soup to extract the sample of the response data, and through the preliminary screening of the sample, select the sample related to the CMS as the learning sample of the algorithm.

(b) The feature extraction module is mainly used for frequent set analysis of learning samples. The improved Apriori algorithm is used to calculate the samples, and the largest frequent degree that meets the support degree is found as the feature information of this type of CMS.

(c) The feature recognition module’s purpose is to use all the feature information obtained by the feature extraction module to perform CMS identification on uncategorized websites, mark the identified websites, and store the database.

2.2. CMS System Identification Method: Feature Extraction
In general, there are two ways to collect information on the website, one of which is to visit a website through a browser and send a request to the target server, and then the server will return header information in the data package after the response. Generally, through the browser to open the F12 developer toolbar, you can view all requests and responses of header information in the Network Options. The other is to visit the website page through the browser and click the right mouse button to view the parsed source code.

Header information in response generally include: Accept-Range, Connection, Content-Encoding, Content-Length, Content-Type, Date, Server, X-Via, X-Powerd-by, etc. However, we only need to collect information other than Accept-Ranges, Connection, Content-Encoding, Content-Length, Content-Type, Date information. Because they just recorded some browsers interactive communication with the server, format requirements and other information and they are not relevant to the version features of site CMS. However, the collection of source code only collects some of these ‘meta’ tags, ‘a’ tags, and ‘script’ tags after removing the page content of non-frame format. For example, partial results of site information are collected in table 1.

| Table 1. Web information collection (http://www.taylorleopoldphoto.com/). |
|-----------------------------|-----------------------------|-----------------------------|
| Information type | Information source | Information content |
|-------------------|-------------------|-------------------|
| Server | Header | Apache/2.2.27 (Unix) mod_ssl/2.2.27 OpenSSL/1.0.1e-fips DAV/2 mod_bwlimited/1.4 |
| X-Pingback | Header | http://ds1.foliowebhosting.com/~taylorle/xmlrpc.php |
| X-Powered-By | Header | PHP/5.4.30 |
| <meta> | HTML | http-equiv="Content-Type" content="text/html; charset=UTF-8" |
| <script> | HTML | type='text/javascript' src='http://ds1.foliowebhosting.com/~taylorle/wp-includes/js/jquery/jquery.js?ver=1.10.2' |
| <script> | HTML | type='text/javascript' src='http://ds1.foliowebhosting.com/~taylorle/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.2.1' |
| <meta> | HTML | name="generator" content="WordPress 3.8.4" |

From table 1, we can infer that each site after removing the content of the Web page itself, the remaining information such as framework, structure and system still has a lot. Trying to find a complete fingerprint features from the CMS in such a variety of feature information, only by artificial means is very difficult. Each of the information on the site is likely completely different. However, through the observation we can find that the same type of CMS message may have many similarities, such as: in the
home page of CMS which build based on WordPress has a meta tag named “generator”. In most cases the content tag involves a string of “WordPress” and so on.

Since each site’s source label is numerous and difficult to display, we need to screen the data initially and extract just the structure of the site and the relevant portion of the system which was added to the algorithm. In the recognition algorithm, we consider parts of the header which is responded by the target site along with the parts of the HTML source in meta tags, a tag and script tag content as Unary itemset of Category set. After the initial screening of 20 Wordpress website information, we extract the contents of the keywords “WordPress” or “wp-content” which is repeated many times or directly appeared. A total of 15 keywords segments were extracted which numbered 1-15, as shown in table 2. “xxx” in table 2 express different sites have different characters; it can be filtered by regulation match. The feature information in the table will be used as a pretreatment comparison table for subsequent experiments.

### Table 2. WordPress website features information.

| No. | Source | Feature content |
|-----|--------|----------------|
| 1   | Header | Link: rel=shortlink |
| 2   | Server | Apache |
| 3   | HTML   | <script type="text/javascript" src="http://xxx/wp-content/plugins/> |
| 4   | HTML   | <script type='text/javascript' src='http://xxx/wp-content/js/> |
| 5   | HTML   | <meta content=xxx powered by wordpress/> |
| 6   | HTML   | <a href=xxx>xxx Powered by wordpress!</a> |
| 7   | HTML   | <script type="text/javascript" src="http://xxx/wp-content/themes/> |
| 8   | HTML   | <meta property="og:image" content="http://wordpress.com/xxx"/> |
| 9   | HTML   | <script type="text/javascript" src="http://xxx/wp-includes/> |
| 10  | HTML   | <meta name="generator" content="WordPress |
| 11  | Header | X-powered-by:PHP/xxx |
| 12  | Header | X-Pingback: http://xxx/xmlrpc.php |
| 13  | Header | X-Powered-By: PleskLin |
| 14  | HTML   | <meta name="viewport" content="width=device-width |
| 15  | Header | Host:wordpress.xxx |

In this paper, the data will be collected from the URL list of the website through Python crawler program for batch crawling, and the obtained server response header and HTML page content, which may contain useful information, will be extracted through regular expressions, and then stored in the database for subsequent data analysis.

### 3. Apriori Algorithm

#### 3.1. The Original Apriori Algorithm

Apriori algorithm as the most influential algorithm of association rules mining is a set of mining frequent association rule algorithm. The main idea of the algorithm is based on the results of the setting frequency recursive in two stages. In classification, the association rules is one-dimensional, single, Boolean association rules [9]. The so-called association rule is to generate the useful, frequent, associated and the correlation rules from the relational database storage, transaction or other existing databases with large amounts of data between sets. The process generally divided into two steps [10]:

1. Find all frequent item sets;
2. Generate correlation rules by frequent item sets, these rules must satisfy minimum support and minimum confidence;

Set the set of items \( I = \{i_1, i_2, i_3, \ldots, i_m\} \), where ‘m’ is the number of sub items; \( i_j \) (1≤j≤m) is a website information data, such as header information contains and the page source code and other data for each label, saying such information as the data item. That kind of information called data item. \( T_j \in I \) is
A subset of \( I, D = \{ T_1, T_2, \ldots, T_n \} \) is a set of \( T_n \), and \( X \in I, Y \in I, X \cap Y = \emptyset \), recorded as a collection \( X \geq Y \)

Rule D, X and Y are interrelated [11].

Support: If \( X \geq Y \) in the S% T establishment, called \( X \geq Y \) of support for S%, namely:

\[
S\% = \left( \frac{\text{\# of transactions \( t \) contains} \ X, \ Y}{\text{\# of transactions}} \right) \times 100\%
\]

The probability that X and Y sets in D simultaneously.

Confidence: the probability of D appearing itemsets A, item sets B also appeared, namely:

\[
C\% = \left( \frac{\text{\# of transactions \( t \) contains} \ X, \ Y}{\text{\# of transactions \( t \) contains} \ X} \right) \times 100\%
\]

The confidence level indicates the strength of rules. Collection of items called item sets, Item set contains \( k \) entries is called a k-item sets. The frequency of item set is the number of events included item sets, referred to as frequency. If the frequency of item sets is greater than \( \text{min sup} \times D \) in the number of transactions, called the frequent item sets. The most important part of Apriori algorithm is to find frequent item sets from a candidate item sets. Apriori algorithm uses a priori knowledge of the nature of frequent item sets to filter through step by step in an iterative manner. If we can find \( K+1 \) item sets from \( k \) item sets, then can find all frequent item sets from the whole data [12].

Firstly, find a collection of a set, denoted \( L_1 \). \( L_1 \) frequently is used to find set \( L_2 \) of 2-itemset, then find \( L_3 \) by \( L_2 \), in this way, until you cannot find a set of \( k \) frequent items. The main idea in pseudo code are summarized as follows [13]:

\[
\begin{align*}
1) & \quad C_1 = \{ \text{candidate1-itemsets} \}; \\
2) & \quad L_1 = \{ c \in C_1 \mid c.\text{count} \geq \text{minsupport} \}; \\
3) & \quad \text{For } (k=2, L_{k-1} = \emptyset, k++) \quad \text{// Until it can no longer generate the largest project set up} \\
4) & \quad C_k = \text{sc_candidate}(L_{k-1}); \quad \text{// Generate candidate set containing k elements} \\
5) & \quad \text{For all transactions } t \in D \quad \text{// Check-processing} \\
6) & \quad C_t = \text{count_support}(C_k, t); \quad \text{// Candidate projects are included in the set of } t \\
7) & \quad \text{For all candidates } c \in C_i \\
8) & \quad c.\text{count} = c.\text{count} + 1; \\
9) & \quad L_k = \{ c \in C_k \mid c.\text{count} \geq \text{minsupport} \}; \\
10) & \quad \text{Next} \\
11) & \quad \text{Resulttest} = \text{resultset} \cup L_k \\
\end{align*}
\]

where, D represents the data stored in the database; minsupport represents the given minimum support; resultset represents the largest project of all sets.

\( \text{Sc_candidate’s function parameters is } L_{k-1} \), namely: The maximum dimension of all \( k-1 \) project set is returned candidate item set contains \( k \) items \( C_k \). In fact, \( C_k \) is a superset of the largest projects \( k \)-dimensional se by supporting of count_support computing projects, and then generate \( L_k \) [14].

3.2. Apriori Algorithm Improvement Ideas

In Apriori algorithm, the main steps of calculating the maximum set as follows:

First, Statistic frequency occurrence of an element and identify those items set equal to or greater than the minimum support, and screen out the set less than minimum support projects so that can produce a one-dimensional frequent item sets \( L_t \).

Then, let computer handle these projects until no longer to produce a higher number of dimensions of frequent item sets. The progress of the cycle is that according to \( k-1 \) dimensional frequent item sets to generate \( k \)-dimensional candidate set in the first step \( k \).

In this paper, an improved algorithm method is to achieve the counting process of the number of concentrated elements before generating \( k-1 \) dimensional frequent items, so in terms of an element can
be improved as follows: If it is less than the count number \( k-1 \), it can remove the elements to advance exclude all redundant, due to a combination of the elements. If an element wants to be a \( k \)-dimensional elements of a \( k-1 \) frequent item set, its counts must reach \( k-1 \), otherwise it is impossible to generate a set of \( k \)-dimensional project.

Then, we need to test the new \( k \)-dimensional projects which focused on all \( k-1 \) dimensional item set whether or not is included in the \( k-1 \) dimensional frequent item sets which have been obtained according to the Apriori algorithm; if one of them is not included, it is also possible to be deleted and finally get a really useful \( k \)-dimensional frequent item sets.

Apriori algorithm has compressed the candidate sets according to its own original features, but it still requires multiple test inevitably for passes all sets. If the algorithm has more Subkey sets, it will be less efficient. The improved method can optimize the operational efficiency. This improved method can get rid of useless data in advance and greatly reduce the number of Subkey set in order to save computing time when we do large-scale data mining for high-dimensional computation. The main idea in pseudo code are summarized as follows:

1) \( C_1 = \{\text{candidate1-itemsets}\} \);
2) \( L_1 = \{c \in C_1 | c.\text{count} \geq \text{minsupport}\} \);
3) For(k=2, \( L_{k-1} = \emptyset, k++ \)) // Loop until the largest item set can no longer be generated
4) \( C_k = \text{sc_candidate}(L_{k-1}) \); // Generate a set of candidate items with \( k \) elements
5) For all transactions \( t \in D \) // Check-processing
6) \( C_t = \text{count_support}(C_t, t) \); // Candidate projects are included in the set of \( t \)
7) For all candidates \( c \in C_t \)
8) \( c.\text{count} = c.\text{count} + 1 \);
9) Drop(Unsatisfied subset) -> Next; // Remove the unsatisfied subset
10) \( L_k = \{c \in C_k | c.\text{count} \geq \text{minsupport}\} \);
11) Next
12) Resulttest = \text{resultset} \cup L_k

4. Experiment and Evaluation

4.1. Experiment Environment
The experimental test environment is a Windows7 PC, as shown in table 3, through which the computer performs crawling and data processing and storage, and finally outputs the experimental results.

| Project          | Explanation                                      |
|------------------|--------------------------------------------------|
| Operating System | Windows 7 64Bit, i5 CPU, 8G RAM, Network 20M     |
| Python           | 2.7.3                                            |
| MySQL            | 5.6.12                                           |

4.2. Experiment Procedures
In the experiment, the improved Apriori algorithm is applied to the most popular WordPressCMS for identification, in which 20 website lists built by WordPress can be obtained through WordPress website as sample data for machine learning.

In the experimental data collection phase, with the help of Scrappy framework [15] and Beautiful Soup page analysis technology [16], the data of these 20 websites are crawled, including the response header of their servers and the HTML data of the responses. With the help of Beautiful Soup module, all kinds of tag data in HTML are extracted, and finally all the data are stored in Mysql database, and URL is used as one of the fields to distinguish the data of different websites.
According to table 2, sample collection and data processing were performed on the website with ID 1, and 6 places were found to match the information in table 2. As shown in figure 2, the matching numbers were 1, 2, 4, 10, 12, and 14.

Figure 2. ID1 website data.

In the same way as the ID1 website, special investigation samples were collected from the remaining 19 websites. The feature information contained in each website is shown in table 4.

Table 4. Twenty WordPress sites contained characterized No.

| ID | Feature No.  | ID | Feature No.  |
|----|--------------|----|--------------|
| 1  | 1,2,4,10,12,14 | 11 | 4,8,9,10,12  |
| 2  | 4,9,10,12    | 12 | 3,4,10,12   |
| 3  | 2,3,4,9,12   | 13 | 2,4,9,10,12,15 |
| 4  | 2,5,10,12,13 | 14 | 3,4,8,9,12,13 |
| 5  | 4,9,10,12    | 15 | 10,12,15   |
| 6  | 1,4,9,10,12  | 16 | 1,4,9,11,12 |
| 7  | 2,7,9,12     | 17 | 2,9,12,15 |
| 8  | 4,6,7,10,12  | 18 | 2,4,8,9,10,12 |
| 9  | 2,5,8,10,12,13 | 19 | 4,10,12,14 |
| 10 | 2,4,6,8,10,12 | 20 | 2,3,4,7,9,10,12,15 |

Set the value of minimum support minsupport 0.5, calculate the support element of each feature as shown in table 5.

Support less than 0.5 of subitme is removed and will continue calculating the rest of the subitem. Each time excluded a subitem which support is less than 0.5 after calculation until it cannot get more items. Calculation is shown in figure 3.

4.3. Experimental Results and Analysis.

Finally, according to the improved Apriori, we obtained the frequent set for \{4,9,12\}, \{4,10,12\}. Accordingly, in the identification of WordPress CMS, the collection of the tag information and the
header information must contain features 4 and 12, and features 9 or 10 can just have one. If these conditions are met, you can identify the site template system as WordPress. Specific features are shown in table 6.

### Table 5. Support of Each unit itemsets.

| Itemsets | Support | Itemsets | Support |
|----------|---------|----------|---------|
| {1}      | 0.15    | {10}     | 0.75    |
| {2}      | 0.5     | {11}     | 0.05    |
| {3}      | 0.2     | {12}     | 1.0     |
| {4}      | 0.75    | {13}     | 0.15    |
| {5}      | 0.1     | {14}     | 0.1     |
| {6}      | 0.1     | {15}     | 0.2     |
| {7}      | 0.15    |          |         |
| {8}      | 0.25    |          |         |
| {9}      | 0.6     |          |         |

### Figure 3. Frequent calculation procedure table set.

### Table 6. WordPress template recognition feature.

| No. | Identification features mark |
|-----|------------------------------|
| 4   | Script tags in HTML source code have the property of “src”, it content “/wp-content/js/” |
| 12  | HEADER has “X-Pingback:” and it end with “/xmlrpc.php” |
| 9   | Script tags in HTML source code have the property of “src”, it content “/wp-includes/” |
| 10  | Script tags in HTML source code in which name is "generator" and property is “WordPress" |

Because of the unknown nature of the calculation results, the method previously proposed by Ma et al. cannot be used to perform targeted compression on the sample data in advance [17]. In this paper, only the invalid data of known infrequent items can be pruned during the calculation process to improve the calculation efficiency. If no improvement to the Apriori algorithm is taken, the item set \{2,12\} will not be removed in advance when item set L2 generating item set L3. Because element 2 only appears once in item set L2, when generating the next-dimensional item according to the optimization algorithm, all item sets containing element 2 should be deleted first, otherwise the program will generate more three non-frequent itemsets when generating item set L3: \{2,4,12\},\{2,9,12\},\{2,10,12\}. After the program generates these three non-frequent sets, it will still calculate the support of each item set and filter through the minimum support. This will increase the number of program operations and reduce the efficiency of the algorithm.

In order to verify the validity of the template recognition feature information from the WordPress of calculation, we collected 130 sites, 64 of which were WordPress CMS template structure, referred to as Class A sample; the number of non-WordPress CMS website template structure is 66, referred to as type
B samples. The above recognition features of WordPress templates are used to do the tests through the regulation match. The experiment result shows that the number of the WordPress CMS template structure is 61, in which identify one errors. There are three WordPress CMS construction sites cannot match the experimental results because of a larger changes in its personalized WordPress template. The results are shown in table 7.

### Table 7. Website CMS system identification results.

| Project Test Samples | Quantity | Class A identification number | Class B identification number | Classification accuracy | Classification recall | Global accuracy |
|-----------------------|----------|-------------------------------|-------------------------------|-------------------------|----------------------|----------------|
| Class A               | 64       | 61                            | 3                             | 0.983                   | 0.953                | 0.970          |
| Class B               | 66       | 1                             | 65                            | 0.956                   | 0.985                |                |

Classification accuracy is defined as the ratio of correct identification output to total output; Classification recall rate is defined as the ratio of correctly identified test samples to total test samples; and global accuracy is defined as the ratio of correct identification of test output to the total number of the test sample.

As can be seen from the experimental results, the CMS system website’s recognition accuracy and recall rate of is over 95%, which has higher accuracy and value based on improved Apriori algorithm.

### 5. Conclusions

This paper proposed a features mining and recognition scheme of CMS system version based on the improved Apriori algorithm. We can obtain CMS system version of features from learning sample data based on the improved Apriori algorithm. Then we established CMS system identification or filtering rules. This method is fast and easy to implement and will not be affected by the result of a sharp decrease in efficiency due to the growth of data. The method of feature mining of CMS version types given in the text can quickly and easily establish a global website system of feature fingerprint database to identify the site system version, and future development of global sites. The site also can be used to identify the management of site and site data analysis and processing.

### References

[1] Tim B L 2019 Total Number of Websites https://www.internetlivestats.com/total-number-of-websites/.
[2] Nick S 2019 Popular CMS by Market Share https://websitesetup.org/popular-cms/.
[3] Huang S X 2018 Portal system based on CMS content management and web anti-tampering technology Public Communication of Science & Technology 145-146.
[4] Nan S H, Wei W, Wu H Q, Zou J R and Zhao Z W 2018 Web server fingerprint identification technology based on KNN and GBDT Computer Science (8) 141-145.
[5] Xu X T, Li M, Sun Y and Gao Z X 2019 Research on attack and defense technology of deep neural network Proceedings of the 3rd National “Smart Grid” Conference pp 444-447.
[6] Wang W W 2017 Web Identification and Obfuscation Technology Based on Website Fingerprint (Wuhan University).
[7] An X Y 2016 Research on Vulnerability Discovery and Verification Technology for Large-Scale Web Server (Beijing University of Posts and Telecommunications).
[8] Wu J S 2013 Research on Website Fingerprint Identification Attack and Protection Technology (Nanjing University of Science and Technology).
[9] Martin J W and Michael I J 2008 Graphical Models, Exponential Families, and Variational Inference (Now Publishers Inc.).
[10] Sayin Y, Verykios V S and Clifton C 2001 Using unknowns to prevent discovery of association rules *ACM SIGMOD Record* **30** (4).

[11] Peter H 2012 *Machine Learning in Action* (Manning Publications) pp 201-221.

[12] Liu H 2019 Association analysis based on apriori algorithm *China Computer & Communication* 132-133.

[13] Li Z Y and Wang R C 2011 A P2P network traffic identification approach based on machine learning *Journal of Computer Research and Development* 2253-2260.

[14] Zhao C L and Ning H Y 2007 Improvement in Apriori algorithm and using in logistics information mining *Journal of Tianjin University of Technology* 159-164.

[15] Dimitrios K L 2018 *Learning Scrapy* (Beijing: People Post Press) pp 35-45.

[16] Bean P 2019 Detailed usage of *BeautifulSoup Library* https://blog.csdn.net/z714405489/article/details/83245087/.

[17] Ma L Z, Jiang B and Bai H J 2016 Improved apriori algorithm based on selection criterion *Journal of Hexi University* 82-84.