Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 2, February 2018, pp. 300-308
DOI: 10.23940/ijpe.18.02.p11.300308

Intrusion Anomaly Detection based on Sequence

Gangyue Lei*

Hunan College of Information, ChangSha, 410200, China

Abstract

For single event sequences, a new anomaly detection method based on SV-LFSP (Short Variable-Length Frequent Sequence Pattern) is presented in this paper. Considering the structure character of procedure calling sequences generated by computer programs, the method defines SV-LFSP and contains three fundamental elements in the program flow, sequence, iteration and selection. To build the SV-LFSP library, the SV-LFSP generation algorithm is used. Essentially, this algorithm follows the idea of TEIRESIAS, with an additional redundancy controlling mechanism. Event flow chart, which has the capability of describing program behavior accurately, is a visual version of the SV-LFSP library. This new method is superior to previously provided frequent episode pattern matching algorithms for compact detection models, with high detection efficiency and low time delays.

Keywords: web application; network security; intrusion detection

(Submitted on October 29, 2017; Revised on December 1, 2017; Accepted on January 8, 2018)

1. Introduction

Anomaly detection defines the characteristic contours of the system normal behaviors and establishes the normal behavior model of the system [1,6]. When any difference between the current behavior and the normal behavior is detected, the system is considered to have been invaded. Anomaly detection is widely used to detect multiple types of intrusion, which can detect unknown intrusion patterns [11].

Event sequence is a general form to describe the behavior of system objects. The intrusion detection data source is divided into sequence class and attribute set class according to the data pattern. Each item in the sequence class data source has a single attribute. For example, in the system call sequence, the unique attribute of the data item is the system call number. Anomaly detection uses the normal behavior model as the detection model, and high-quality normal behavior modeling is the key to anomaly detection [7,5]. It is rather difficult to carry out the definition and training of the normal behavior model, because the model must present accuracy, guidance and efficiency simultaneously. Firstly, the accuracy of the model directly controls the detection efficiency of the system—that is, whether a higher detection rate can be achieved based on a lower false detection rate and missing rate. Secondly, anomaly detection is used to not only to recognize the intrusion but also to help understand intrusion. Guidance refers to the fact that the model can not only provide the basic function of intrusion recognition but also offer sufficient information in order to facilitate the analysis and control of the intrusion pattern. Thirdly, the efficiency of the model assists in designing the efficient detection algorithm and achieving real-time online detection [12,3].

The sequence is a linear list of unit events, and the abnormal sequence is of great value for the intrusion detection [2,9]. The procedure calling sequence of application constitutes the main source for observing the sequence anomaly analysis. An anomaly detection method based on the short variable-length frequent sequence Pattern (SV-LFSP) is proposed, which further investigates the basic structure of the application procedure calling sequence. Based on the SV-LFSP model mined through the verbose mode control mechanism, a new SV-LFSP event flow chart model is established, which is composed of a sequence, selection and loop. Experiments show that the sequence anomaly detection algorithm, as the detection model of the SV-LFSP event flow chart, is simple and efficient with a low detection cost and false detection rate while maintaining a high detection rate [10, 4, 8].

* Corresponding author.
E-mail address: gangyuelei2017@163.com

2. Frequent short sequence model

The sequence analysis used for anomaly detection usually utilizes the frequent short sequence model in order to find abnormal sequence fragments. The basic idea of the frequent short sequence model is to extract the standard pattern library composed of frequent sequence fragments from normal event sequences, which are used to summarize the behavioral norm of the observing subject. However, the abnormal sequence fragments do not match the pattern in the standard pattern library, thereby deviating from this norm.

The frequent short sequence reflects the local relations among the events in the sequence, fully covering the local behaviors of the observing subject that generates the unit event sequence and providing the appropriate model for behavior modeling of the observing subject. Forrest was the first scholar to consider the system call occurring in the application implementation as the observing event. He tracked the system calling sequence that generated the application implementation, extracted the fixed-length frequent short sequence from sequence sets obtained through sufficient observation, and modeled for the application. The fixed-length frequent short sequence is called the N-LFSP pattern. Frequent means that the short sequence occurs in the training sequence set N-LFSP many times. Supposing there are s training sequences in the training sequence set with the short sequence P included, then the support degree of P is denoted by s, and if s is greater than the specified threshold of SUP, then P is called the frequent short sequence.

3. Short Variable-Length Frequent Sequence Pattern

The anomaly detection algorithm using the N-LFSP pattern is very simple and intuitive. Yet, it is a problem to determine the appropriate N value in different applications. A N-LFSP pattern that is too short may excessively decompose the abnormal sequences, leading to weak abnormal signals and difficult detection, while a N-LFSP pattern that is too long may result in a large pattern library, increased detection costs, and a high false detection rate due to its strictness. Hofmeyr's [12] experimental results show that when N=1, the mismatch rate is less than 7% even if the pattern library is used as the standard library obtained by completely different application training. When N=30, the mismatch rate is 100%. Although many studies have shown that better effects would be achieved when N=6, there is no clear standard to choose the appropriate value of N under different application environments.

In order to extract more suitable frequent short sequences from the training set of the unit event sequence, Wespi et al. proposed a method of anomaly detection by constructing a variable frequent short sequence pattern library. This method first uses a variable length pattern mining algorithm to output the TEIRESIAS candidate gene set, and then selects the optimal subset from the candidate set. The subset of the training set, the smallest subset, covers the unit event sequence. Compared to the fixed length N-LFSP model, the Wespi method can use less variable length patterns (denoted as V-LFSP mode), flexibly covering training sequence sets.

Regardless of complexity of the application, its codes are always composed of three basic structures: order, selection and loop. Various procedures calling sequences, which are obtained by tracking the application implementation, become a snapshot for the combination of these three basic structures. Therefore, the features of the three structures are also reflected in the calling sequence. The detection method based on variable-length pattern can well perform the modeling and detection of calling sequences composed of sequential structures.

However, for sequences with a large number of selection structures, neither N-LFSP nor V-LFSP have adequate ability to describe them. As a result, the short variable-length frequent sequence pattern (SV-LFSP) is obtained by extending SV-LFSP and allowing variable-length patterns to contain finite indeterminate elements.

In the maximum model, “maximum” means that it is impossible for the pattern to be extended as a stronger patter without reducing the support degree. The verbose pattern is a relative concept, whereas verbosity means that under a maximum model library, if a pattern randomly occurs in the training sequence set and all its occurrences are covered by a known stronger pattern in the library, the pattern becomes verbose in the library. In Table 1, the pattern C.DE is the maximum model. No matter how the pattern extends its support degree, the pattern is verbose relative to other two patterns, because AC.DE and C.DEF are sufficient to cover the any occurrence of C.DE in the training sequence set.

Table 1. Maximum mode and redundancy mode

Training sequence set	Max mode (part)	Degree of support	Redundancy mode
ABXCYDEF ACXDEYFG ABDCXDEF ACZDEGHI	C.DE	4	C.DE
	AC.DE	2	
	C.DEF	2	

The SV-LFSP gap density is a key factor in defining the density of the gap. The TEIRESIAS algorithm <L, W> mode method to define SV-LFSP, SV-LFSP must meet the following conditions:

1. The length of the pattern is not fixed, but the starting and ending elements in a pattern must be the determining element—that is, the gap cannot appear on the starting and ending point of the mode;
2. For any pattern in which the elements are identified as the starting and ending elements, if the number of elements determining the number of sub patterns is L, then the length of the sub model cannot be greater than W;
3. The pattern must be frequent—that is, for a given training set, the support of the model must be greater than the threshold of SUP;
4. Must be the maximum mode;
5. Must be a non-redundant mode.

The short sequence satisfying conditions (1) and (2) is called the $< L,W >$ pattern, and Condition (2) is related to the gap density. L, W, and SUP are user-defined integer arguments. According to the definition, the difference between W and L determines the gap density of $< L,W >$; the greater the difference, the more gaps in the pattern. For example, if W and L are 4 and 3 respectively, the short sequence AB.CD is $< L,W >$ patterned; however, AB.C.D does not belong to this pattern, because the sub-sequence B.C.D determines the number of elements of 3 but has a length of 5, exceeding the specified value of W. According to the above five conditions, SV-LFSP can be defined as frequent, the largest and non-verbose $< L,W >$ pattern.

In Table 2, A-F represent six kinds of system calls. The table records all system calling observation sequence modeling that may be generated for a simple application code. If N- LFSP with a length of 3 is used to establish the pattern library, 17 patterns containing all information of the normal mode are needed; if V-LFSP is used, 8 patterns are needed; if SV-LFSP with recognized loops is used, only 2 patterns are needed. Among the three frequent short sequence patterns, it is obvious that SV-LFSP is the actual structure closest to the code. This is because the variable-length features of the SV-LFSP pattern can better describe the sequential structure, the gap can describe the selection structure, and the repeated descriptor can describe the loop structure. The gap in SV-LFSP contains an arbitrary element, the loop body is denoted by square brackets, and * acts on the left side of the loop body, which represents one or more cycles around the loop. The loop body can be nested and corresponded to the flexible loop structure in the program structure.

Table 2. Comparison of three kinds of frequent short sequence pattern library

Code	System call Sequence training set		Frequent short sequence pattern database		
			N-LFSP (n=3)	V-LFSP	SV-LFSP

A;
FOR(…)
}
B;
IF(…) C;
ELSE IF(…) D;
ELSE(…) E;
F;
}
E;
F;
	1	AEF	AEF ABC
			BCF CFE
			FEF ABD
			BDF DFE
			ABE BEF
			EFE CFB
			FBD FBE
			DFB FBC
			EFB
				EF
				ABCF
				ABDF
				ABEF
				BCF
				BDF
				BEF
				EF
					AEF
					A[B.F]*EF

	2	ABCFEF			
	3	ABDFEF			
	4	ABEFEF			
	5	ABCFBDFEF			
	6	ABCFBEFEF			
	7	ABDFBEFEF			
	8	ABCFBCFEF			
	9	ABDFBDFEF			
	10	ABEFBEFEF			
	11	ABCFBDFBEFEF			

In the Table 2, A-F indicate 6 system calls. Table 2 shows the possibility of a simple application code. The results of all system call observation sequence modeling are generated. For the length of 3 N-LFSP to establish the model library, 17 modes are required to contain all normal mode information, while using V-Gram requires 8 modes.

The use of SV-LFSP is to identify the cycle, with only 2 models. It is obvious that SV-LFSP is closest to the actual structure of the code in three frequent short sequence patterns. The variable length property of the SV-LFSP model can better describe the sequential structure, and the gap can describe the selection structure. The gap of SV-LFSP contains an arbitrary element, and the loop body is represented by square brackets. * acts on the left side of the loop body, which represents one or more cycles, and the loop can be nested. Because SV-LFSP can describe the structure of the code to a certain extent, even if the training set is not enough, SV-LFSP can be fitted to a number of normal short sequences to further reduce the false detection rate. Description ability of the N-LFSP model on program behavior is very limited when the training set is insufficient, because while the N-Gram training pattern library is large, the information contained is local and behavior and program fitting degree is not high.

The SV-LFSP model takes into account the variable length structure of the relatively fixed order structure in the procedure call to improve the description of the sequential structure, but it still does not take into account the existence of a large number of options and loop structures.

For all the application calling sequences that can be found, even if the observation sequence is derived from the same code, the difference of the sequence lengths is very large (especially the cyclic structure). The sequence length may reach the magnitude of the difference, and the training and testing are very unfavorable.

A large number of repetitive loops will not bring any new patterns, but they will cause great training cost and detection overhead. More importantly, in the detection stage, the huge difference in the length of the sequence to be measured will make the selection of abnormal index more difficult. The anomaly detection algorithm based on frequent short sequence pattern uses the number of mismatch patterns or the percentage of mismatch patterns as an anomaly; when the abnormal index exceeds a certain threshold, it is judged to be abnormal.

The two indexes are directly related to the length of the sequence, and the huge difference in the length of the sequence will cause the loss of judgment value. For example, suppose that a sequence to be tested is abnormal and contains a large number of repeating loops. If the anomaly of the sequence does not occur in the loop structure, it is likely that the anomaly index will be smaller than the threshold value, so that the abnormal signal cannot be detected in a large number of repeated normal signals.

Therefore, circular duplicate fragments in the sequence are eliminated so that the training sequence and sequence length are measured roughly at the same level, thereby reducing the training and testing costs, allowing for the identification of abnormal sequences, and improving the detection rate.

4. SV-LFSP Pattern mining

4.1. Explicit Cycles

The sequence of loops in an explicit loop is fixed and is easily identified by the sequence of events. If the code contains only two structures, the order and the loop, then the sequence contains only explicit loops. It is very simple to eliminate the explicit cyclic fragment. Table 2 is an example of a system call sequence that can be generated by the code. The elimination of an explicit loop fragment is shown in Table 3. If the sequence contains nested multi-layer loops, the elimination of the loop is an iterative process.

Table 3. Eliminate explicit loop segments

	Sequence	Explicit loop segment
Initial sequence	ABCFBDFBDFBFBFBEFBEFBDFBFGH	
Identify explicit loop segments	ABCF BDFBDF BCFBCF BEFBEF BDFGH	BDF, BCF, BEF
Eliminate explicit loop segments	ABCF BDF BCF BEF BDFGH	

4.2. Basic Model

It is necessary to dig out all the basic patterns from the training sequence by eliminating the explicit cycle. The next step is to get the foundation of SV-LFSP. Assuming that the parameters L, W, and SUP are specified, the basic pattern is to determine the number of elements that are just $< L,W >$.

According to this definition, the length of the basic mode Len is between L and W, and the middle position of the model contains up to W-L gaps.

The Pratt algorithm is an exhaustive algorithm based on depth-first traversal, which can mine all frequent $< L,W >$ patterns in the training sequence set. The algorithm constructs an exhaustive pattern tree, in which each complete branch of a pattern tree represents a frequent $< L,W >$ pattern, and the known patterns are extended by exhausting all possible pattern elements (determining elements or gaps). If the extended pattern is still frequent, then continue to extend or return to the previous branch and extend until all the frequent patterns are exhausted.

The exhaustive search strategy can be frequent $< L,W >$ patterns of arbitrary length. However, it needs to calculate the

support degree of a large number of unknown extension patterns, greatly increasing the scanning number of the training sequence set. In order to enhance the efficiency, the fixed-length pattern insertion method is adopted to respectively construct the pattern tree for each pattern with length between L and W. Therefore, it only needs to calculate the support degree of actual pattern, reducing the scanning number of the training sequence set. Suppose it is necessary to construct the pattern tree of the basic pattern of the long Len, then the algorithm steps are as follows:

(1) Establish the root nodes of the tree pattern tree T, denoted as NULL;
(2) Slide on all training sequences at the fixed-length Len window, and calculate the support degree of the fixed-length short sequence that falls into the window. If the support degree is greater than the specified threshold SUP, a fixed-length frequent short sequence P is obtained;
(3) The basic pattern may contain M=WL gaps, and compare P with all branches in T. Suppose the mismatch number D (i.e., Hamming distance between P and P') of branching pattern P' and P in T is the smallest: if D is not greater than M, then P is merged into the P' branch, the mismatch becomes the candidate gap, and all optional elements in the candidate gap are recorded; otherwise, a new branch P is established at the root node NULL;
(4) Continue to slide the window until all training sequences are traversed.

Figure 1 and Figure 2 are the system call sequences with Table 2 as the training set, specifying two basic pattern trees obtained by L=2, W=3, and SUP=1. The basic pattern of the gap free length is 2, and the basic mode of the single candidate gap has a length of 3. In the pattern tree that contains the candidate gap, the optional element is in the candidate gap.

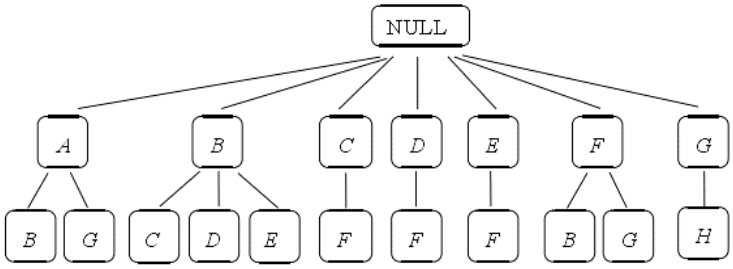

Figure 1. Non-gap basic pattern tree (L=2, W=3, SUP=1)

When the threshold value is K, the candidate gap is transformed into a gap, which can match any element. Figure 1 does not contain the candidate gap. All modes are the basic mode. Hypothesis K=2 is shown in Figure 2. Only mode B[CDE]F is converted to the basic mode B.F. The clearance of the other modes is at the starting and ending points of the mode, which does not conform to the requirements of <L and W> mode.

The basic patterns of different lengths are excavated separately, while the longer basic modes contain more gaps. Therefore, it may be the case that the shorter mode is relatively large in the case of a relatively large pattern. These shorter patterns do not contribute to the next generation of stronger patterns through connection and should be excluded from the basic mode.

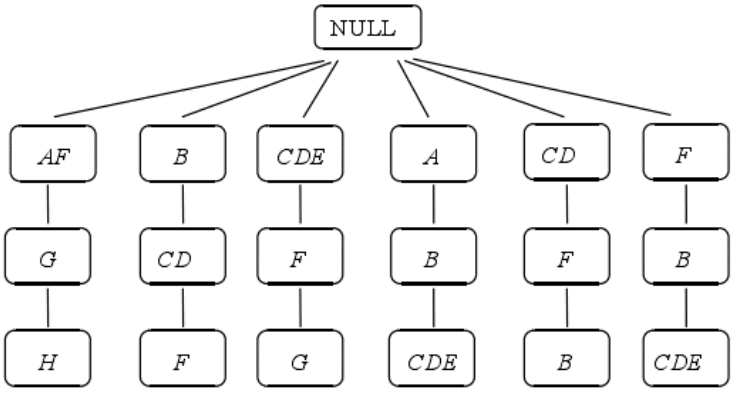

Figure 2. Single candidate gap basic pattern tree (L=2, W=3, SUP=1)

The presence of a single gap mode of 3 is B.F. In Figure 1, the six mode BC/BD/BE/CF/DF/EF with a length of 2 is not the largest or most redundant. Therefore, the final basic mode before the next step to generate the SV-LFSP mode is only 6: AB/AG/GH/FB/FG/B.F.

4.3. SV-LFSP Pattern generation algorithm

The TEIRESIAS algorithm takes the basic pattern as input and outputs all the frequent $< L,W >$ patterns in the training sequence set. The basic operation of the algorithm extension pattern is joined end to end: suppose there are patterns P and Q, if the last L-1 element of P is exactly the same as the first L-1 element of Q, then P and Q are connected to form a stronger extension pattern (R); if R is still a frequent pattern, then a new frequent $< L,W >$ pattern is obtained. Taking patterns B.F and FG as the example, they can be connected to form a new frequent $< L,W >$ pattern B.FG, and B.F and FB can also be connected to form a new pattern B.FB. The TEIRESIAS algorithm continuously connects the known patterns until a new pattern cannot be found. The basic steps are as follows:

1. The basic mode of the input is sorted according to a certain rule and all discharged into the queue;
2. Take a pattern P from the queue, called the current mode;
3. Find all of the P from the queue that can be connected to the right from the operation of the mode Q, connect P and Q to get the extended mode R, and check the expansion mode R. If the R is not frequent, discard; otherwise, the R is the current model. Recursive call this step to achieve the depth of the first mode expansion.

According to the training sequence set in Table 2, Table 4 describes the connection between the right side of the basic mode FB and the B.F. Since all of the items of the schema FB match list are covered by the extended pattern FB.F, the coverage of all the matches after the update is set, so that FB is redundant and discards the FB.

Table 4. The operation of the matching table

Pattern	Matching list
P = FB(Before update)	(5,4,0), (6,4,0), (7,4,0), (8,4,0), (9,4,0), (10,4,0), (11,4,0), (11,7,0)
Q = B.F	(2,2,0), (3,2,0), (4,2,0), (5,2,0), (5,5,0), (6,2,0), (6,5,0), (7,2,0), (7,5,0), (8,2,0), (8,5,0), (9,2,0), (9,5,0), (10,2,0), (10,5,0), (11,2,0), (11,5,0), (11,8,0)
R = FB.F	(5,4,0), (6,4,0), (7,4,0), (8,4,0), (9,4,0), (10,4,0), (11,4,0), (11,7,0)
P =FB(After update)	(5,4,0), (6,4,0), (7,4,0), (8,4,0), (9,4,0), (10,4,0), (11,4,0), (11,7,0)

5. Event Flow Diagram

Each branch of the model is used as a node to identify the loop fragments in the model as the loop. From the point of view of the SV-LFSP pattern, these loops are explicit. Iteration of each layer is identified from the innermost layer. Each iteration is used to identify a loop until there are no duplicate segments on the branch. In order to reduce the storage overhead and reduce the cost of detection, it is necessary to establish a new branch in different places, so as to keep the tree structure of event flow graph. Figure 3 is an event flow diagram corresponding to the SV-LFSP pattern collection in Table 2.

The event flow graph is a tree with a loop, each of which represents a separate SV-LFSP pattern from the root node NULL. A loop containing a descendant node to an ancestor node shows the cyclic structure.

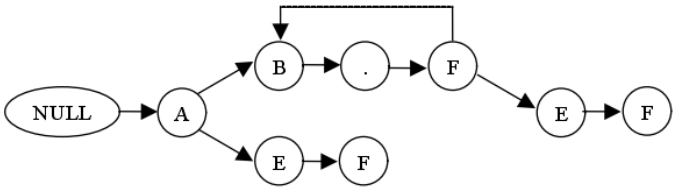

Figure 3. Event Flow Diagram

6. Experimental Analysis and Results

In this paper, we propose an event flow graph detection algorithm based on SV-LFSP and two kinds of event sequence anomaly detection algorithm. The STIDE algorithm is based on the N-LFSP model and the V-LFSP model is based on the variable frequent sequence pattern matching algorithm. The results show that the three algorithms of the detection model scale, the false detection rate, detection rate and detection overhead performance test.

When Forrest first proposed to use the system calling sequence as the observation data source of the host anomaly detection, he adopted two sets of data obtained by tracking the system calling sequence of send-mail applications, namely, the UNM synthetic data sets and CERT send-mail synthetic data sets. Each data set contains more than 1.5 million system calls. Here, synthesis means to build a comprehensive application scenario for send-mail, in order to track the system calling sequence in as much detail as possible, including the normal application scenarios and intrusion scenarios. Therefore, the two sets of data also contain the normal sequence and the intrusion sequence. The data set consists of a series of files, and each file corresponds to an application scenario, recording the system call tracked in the scenario. The data are divided into two columns, respectively representing the process ID and the sequence number of the system call. It extracts sequence numbers of the system call in the same process ID by category to form a sequence, so that each scenario corresponds to a sequence set, which is called a category. Three frequent short sequence pattern libraries are created separately for each category of normal application scenario, and the size of each pattern library is shown in Table 5. When the pattern library is created, two lengths of 6 and 10 are taken for the N-LFSP pattern, the values for the L and W parameters selected for the SV-LFSP pattern are 3 and 4 respectively, and the maximum length of the V-LFSP and SV-LFSP patterns is limited to 30. Additionally, since the training sequence can be guaranteed to originate from the normal application scenario, the threshold of support degree SUP is set to 1.

The SV-LFSP pattern library after loop recognition is the most compact. The reasons are as follows: firstly, the event loop body is recognized in two phases in the establishment of the event flow chart, the explicit loops are directly eliminated against the training sequence in the first stage, and the potential implicit loops are recognized from the SV-LFSP pattern with gaps in the second stage. Secondly, according to the definition of the SV-LFSP pattern, the pattern mining algorithm tends to output a longer strong pattern, and the pattern library excludes any verbosity. It is shown in Table 5.

Table 5. Comparison of three kinds of frequent short sequence patterns

Normal system call sequence set		Mode base size			
		N-LFSP		V-LFSP	SV-LFSP
Data set	category	N=6	N=10		L=3, W=4
UNM	bounce1	31	58	21	9
	Bounce2	57	97	27	16
	bounce	79	121	35	19
	plus	164	231	42	20
	queue	114	174	29	11
	sendmail.daemon	362	598	89	52
	sendmail.log	131	209	32	21
CERT	sendmail.daemon	217	335	61	44
	sendmail	71	127	32	27

The false detection rate of the algorithm can be measured by calculating the abnormal index of the normal sequence. To compare the false detection rate of the three patterns in inadequate training, the UNM and CERT send-mail normal system calling sequences are divided into two groups by three different scales. One group is used to obtain the pattern library through training, and the other group is used to calculate the abnormal index of normal behaviors in the detection set. For comparison, the detection algorithm based on the N-LFSP pattern library uses the maximum Hamming distance of the mismatched pattern in each sequence as the abnormal value of the sequences, while the algorithm based on V-LFSP and SV-LFSP is handled similarly. For the abnormal value of the sequence, the longest consecutive abnormal calling sub-sequence is taken, and then the average of the abnormal values of all the sequences in the test set is taken as the abnormal index. The three algorithms and abnormal index of normal sequences are shown in the absence of training in Table 6. Even in the case of severely inadequate training, SV-LFSP still maintains a high matching degree for the normal sequence. The V-LFSP abnormal index in the CERT data set is even greater than N-LFSP, because the training set coverage is oriented in the selection of V-LFSP pattern library and the completeness of pattern library depends greatly on the size of the training set.

Table 6. Mean comparison of abnormal index of normal sequence (insufficient training)

Normal system call sequence set			Mode base size			
			N-LFSP		V-LFSP	SV-LFSP
Data set	Training set	Detection set	N=6	N=10		
UNM	90%	10%	1.67	2.16	0.63	0.43
	60%	40%	1.78	2.87	0.81	0.75
	30%	70%	2.65	4.08	1.87	1.43
CERT	90%	10%	1.67	2.98	1.12	0.32
	60%	40%	1.80	2.71	2.45	0.54
	30%	70%	2.67	3.89	4.02	0.94

The UNM and CERT two sets of data sets contain a full set of system call sequences in the intrusion scenario, which can be used to test the detection rate of the algorithm. There are several sets of data for each kind of intrusion scene. Each group

of data represents the whole process of tracking the system call sequence generated by a single intrusion. The average anomaly index of each sequence in each scenario represents the anomaly index. Three kinds of algorithms for a variety of intrusion scenarios are shown in Table 7.

Table 7. Comparison of anomaly index of send-mail intrusion scenarios

Normal system call sequence set		Mode base size			
		N-LFSP		V-LFSP	SV-LFSP
Data set	category	N=6	N=10		
UNM	decode	3.2	5.3	11.6	7.4
	fwd_loops	4.1	6.8	13.3	11.5
	sscp	5.2	7.4	19.6	12.6
CERT	sm5x	4.5	9.3	13.6	9.6
	sm565a	3.8	7.4	12.6	11.6
	syslog-local	4.6	9.6	17.8	13.7
	syslog-remote	4.7	8.5	10.1	9.5

From the perspective of abnormal signal intensity, the flow detection algorithm of SV-LFSP mode can be combined with the existence of the gap. Reducing the false detection rate ensures normal sequence matching and may also lead to the decomposition of the abnormal subsequence and weakening of the abnormal signal. On the other hand, the SV-LFSP model is a long strong mode, and this weakening effect is very limited. According to the experimental data, the anomaly index based on SV-LFSP model is slightly lower than the V-LFSP's algorithm; however, it is still sufficient to detect intrusions.

Figure 4 shows the average detection cost of three detection algorithms for all sequences in the CERT data set. The detection algorithm based on the V-LFSP pattern has a large detection cost, because the algorithm must perform a forward pre-matching process each time it selects a matching pattern from the V-LFSP pattern library and select one optimal pattern for sequences to be detected within a limited future scope. There is a large computational cost in the pre-matching process. The cost of the detection algorithm based on N-LFSP comes from the large pattern library. The larger the pattern library, the greater the cost of each matching. In addition, the step size for the sliding window is 1. Each event is repeatedly detected for N time. When the event flow chart is used for the model detection, each event only needs to be detected once, and the pattern library has a smaller size. Therefore, the detection cost is lower, satisfying the requirements of real-time detection.

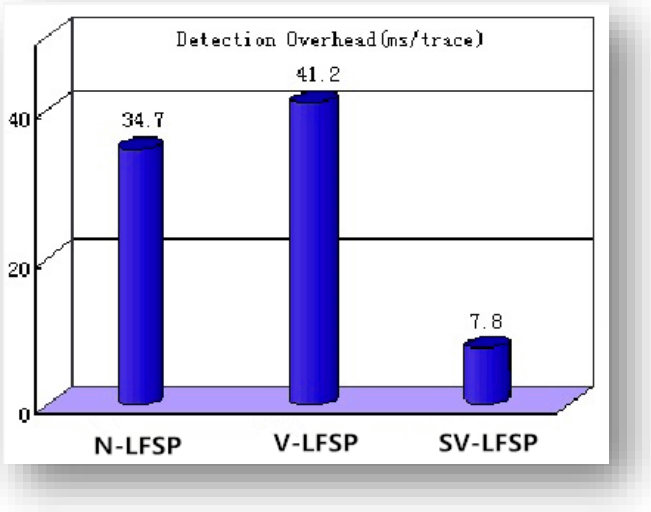

Figure 4. Three frequent short sequence patterns for each sequence detection overhead

The detection algorithm based on V-LFSP is called variable frequent short sequence matching. The detection algorithm based on SV-LFSP is called event flow graph matching. It is shown in Table 8. Table 8 shows the training time cost and training algorithm complexity of the three detection models. When the STIDE algorithm is used to create the N-LFSP pattern library model, it only needs to scan all training sequence sets once. The method is featured by low time cost, large model size and higher detection cost. During training of the V-LFSP pattern library, the frequent elements set is obtained by scanning the training sequence set using the TEIRESIAS algorithm. The SV-LFSP pattern library is trained in three stages: first, scanning the training sequence set, removing the explicit loop fragments and obtaining the frequent elements set; then, creating the basic pattern library that contains uncertain elements with frequent elements set as the input; and, finally, mining the SV-LFSP pattern library with the basic pattern library as the input by using the TEIRESIAS algorithm with verbose control. It

should be noted that the problem size independent variable of the training algorithm complexity is related to algorithms at all stages in the training process, namely, the sizes of the training sequence set, the frequent elements set and the basic pattern library.

Table 8. Unit event sequence anomaly detection algorithm training overhead

Detection model	Training cost			
	Training process and algorithm	Algorithm complexity		Time overhead
		Scale independent variable	Asymptotic complexity	
N-LFSP (N=6)	Mining N-LFSP pattern library	Training set	O(n)	14.5
V-LFSP	Get frequent sets of elements	Training set	O(n)	4.2
	Mining SV-LFSP schema library (TEIRESIAS)	Frequent element set	$O(n^M)$	26.5
SV-LFSP	Elimination of explicit cyclic fragments and obtaining frequent element sets	Training set	O(n)	8.7
	Basic pattern library	Frequent element set	$O(n^2)$	17.9
	Mining SV-LFSP pattern library	Basic pattern library	$O(n^M)$	25.4

7. Conclusions

The SV-LFSP model proposed in this paper takes into account the three basic structures of the procedure call and can describe the program flow more efficiently. The event flow chart of the SV-LFSP pattern matching algorithm maintains a relatively high detection rate and low false alarm rate while also having a smaller scale detection model and low detection cost, making it suitable for use in online event sequence anomaly detection units.

References

1. Saihua Cai, “Research on Component Security Anomaly Detection Method Based on Monitoring Log Mining”, *Jiangsu University*, 2016
2. Jing Du, Yuanyuan Chen, “Anomaly Detection Based on Hidden Markov Model (HMM)”, Journal *of Taiyuan University of Science and Technology*, vol.9, pp. 16-19, 2008.
3. A. Hofmeyr, A. Somayaji, and S. Forrest, “Intrusion Detection System Using Sequences of System Calls”, *Journal of Computer Security*, vol.6, no.3, pp.151-180, 1998
4. Yu Ji, “Study on the Key Problems in the Process of Sequential Pattern Discovery”, HeFei *University of Technology*, 2008
5. Guoyuan Lin, “Research on Anomaly Detection Based on Host Behavior”, *Nanjing University*, 2011
6. Hongli Li, “Research on Behavior Matching and Evaluation of Time Series”, *The PLA Information Engineering University*, 2014
7. Shangzhe Shi, “Anomaly Detection Based on Hidden Markov Model”, Yangzhou *University*, 2012
8. Ying Sun, “Research and Implementation of the Key Problems in the Process of Sequential Pattern Discovery”, *HeFei University of Technology*, 2005
9. Kai Xiong, “Research on Frequent Sequence and Closed Sequence Mining Method Based on Minimum Position”, *Northeastern University*, 2012
10. Jifeng Yu, “Anomaly Detection Research of Web Application Based on Data Mining”, *Huazhong University of Science and Technology*, 2011
11. Yang Yang, “Research on Intrusion Detection Technology Based on Linux Process Behavior”, *University of Electronic Science and technology of China*, 2014
12. Jing Zhao, “Research and Application of Network Protocol Anomaly Detection Model”, *Beijing Jiaotong University*, 2010

Gangyue Lei received his M.S degree from the Central South University. He is an associate professor at the Hunan College of Information. His research interests include Extract-Transform-Load for Big Data.