The Pennsylvania State University
The Graduate School
College of Information Sciences and Technology

A DATA TRIAGE RETRIEVAL SYSTEM
FOR CYBER SECURITY OPERATIONS CENTER

A Thesis in
Information Sciences and Technology

by
Tao Lin

© 2018 Tao Lin

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2018
The thesis of Tao Lin was reviewed and approved* by the following:

Peng Liu  
Professor of Information Sciences and Technology  
Thesis Advisor

Nan Zhang  
Professor of Information Sciences and Technology

Sencun Zhu  
Associate Professor of Computer Science and Engineering  
& Information Sciences and Technology

David Fusco  
Director, Masters Programs and Assistant Teaching Professor of Information Sciences and Technology

Andrea Tapia  
Director, Graduate Programs and Associate Professor of Information Sciences and Technology

*Signatures are on file in the Graduate School
ABSTRACT

Triage analysis is a fundamental stage in cyber operations in Security Operations Centers (SOCs). The massive data sources generate great demands on cyber security analysts' capability of information processing and analytical reasoning. Furthermore, most junior security analysts perform much less efficiently than senior analysts in deciding what data triage operations to perform. To help analysts perform better, retrieval methods need to be proposed to facilitate data triaging through retrieval of the relevant historical data triage operations of senior security analysts. This thesis conducts a research of retrieval methods based on recurrent neural network, including rule-based retrieval and context-based retrieval of data triage operations. It further discusses the new directions in solving the data triage operation retrieval problem.

The present situation is that most novice analysts who are responsible for performing data triage tasks suffer a great deal from the complexity and intensity of their tasks. To fill the gap, we propose to provide novice analysts with on-the-job suggestions by presenting the relevant data triage operations conducted by senior analysts in a previous task. A tracing method has been developed to track an analyst's data triage operations. This thesis mainly presents a data triage operation retrieval system that (1) models the context of a data triage analytic process, (2) uses recurrent neural network to compare matching contexts, and (3) presents the matched traces to the novice analysts as suggestions. We have implemented and evaluated the performance of the system through both automated testing and human evaluation. The results show that the proposed retrieval system can effectively identify the relevant traces based on an analyst's current analytic process.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................................................................. vi

LIST OF TABLES ................................................................................................................... vii

Acknowledgements .................................................................................................................. viii

Chapter 1 Introduction ........................................................................................................... 1

Chapter 2 Triage Analysis in SOCs ......................................................................................... 4
  2.1 Data Triage for Cyber SA ............................................................................................. 5
  2.2 Multi-Source Data in SOCs .......................................................................................... 6
  2.3 Data Triage Operation ................................................................................................ 7

Chapter 3 Problem Overview .................................................................................................. 8
  3.1 Difficulties in Data Triage Tasks .................................................................................. 8
  3.2 Experts' Knowledge of Data Triage ............................................................................. 9
  3.3 A Framework for Data Triage Knowledge Retrieval System Designs ....................... 9
  3.4 Challenges in Developing Effective Data Triage Knowledge Retrieval Systems ..... 10

Chapter 4 Deep Learning based Retrieval of Triage Operations ........................................ 12
  4.1 Properties in Recurrent Neural Networks .................................................................. 12
  4.2 Data Triage Operation Retrieval based on Recurrent Neural Networks .................... 13
  4.3 Data Triage Model ..................................................................................................... 14
  4.4 Triage Operations through Time ................................................................................ 16
  4.5 Data Triage Operation and Characteristic Constraint ............................................... 17
  4.6 Trace and Context ...................................................................................................... 17
  4.7 Insights: Context-Driven and Efficient ...................................................................... 19
  4.8 Challenges in Using Machine Learning for Data Triage Operation Retrieval .......... 20

Chapter 5 Evaluation .............................................................................................................. 21
  5.1 Test Cases Generation ............................................................................................... 22
  5.2 Ground Truth .............................................................................................................. 23
  5.3 Random Selection ...................................................................................................... 24
  5.4 Performance Measurement ......................................................................................... 24
  5.5 Human Evaluation ..................................................................................................... 27
    5.5.1 Evaluation Protocol ............................................................................................ 27
    5.5.2 Sequence Length ............................................................................................... 28
    5.5.3 Result ................................................................................................................. 29
    5.5.4 Case Study ....................................................................................................... 29

Chapter 6 Related Work ........................................................................................................... 31
LIST OF FIGURES

Figure 1 Experiment measurement ................................................................. 26
Figure 2 The precision-recall curve ............................................................... 26
Figure 3 Data analysis processes in SOCs ...................................................... 32
Figure 4 The critical features in an attack graph ............................................. 34
Figure 5 Experience relaxation levels .......................................................... 36
Figure 6 Hierarchical experience networks .................................................. 37
Figure 7 The architecture of the rule-based knowledge retrieval system ....... 38
Figure 8 An E-Tree example ....................................................................... 40
Figure 9 An example of the logical relationships between data triage operations ... 44
LIST OF TABLES

Table 1 Examples of data triage operations ................................................................. 15
Table 2 Experiment results .......................................................................................... 25
Table 3 Satisfaction rate for sequence length ......................................................... 28
Acknowledgements

I sincerely thank my advisor Prof. Peng Liu for his guidance and support throughout the time of my study at Penn State. He always encouraging us to make difference. These words give so much positive power to me. Compared to Newton’s famous expression, there is a similar Chinese verse--if one wants to see far, they need stand higher. As a student, I am a great pleasure to work with him. As another old Chinese saying goes, a good teacher not only is excellent in teaching skills, but also has a good reputation in daily life. In my opinion, Prof. Liu is such a teacher. He is always hard-working, which is the model to all of our lab student

I would like to thank Prof. Sencun Zhu as my committee members for advising my research on this thesis. He always guided me on the research and gave kind support and advice for my career. Prof. Zhu’s valuable suggestions and insightful comments help to improve the readability and reduce ambiguity of this thesis. Many thanks to him for always offering me constructive help.

I would like to thank Prof. Nan Zhang as my committee members for advising my research on this thesis. I learned so many research methods from Prof. Zhang’s course, which are not easy to obtain from any computer programming books. Prof. Zhang always encouraging us from his own experiences, such as the story of “To edit wiki to get citation”.

Finally, I would like to express my thanks to all friends I met at Penn State. They gave me a lot of support in my life and research.
Dedication

To my love,

it is difficult to be water for one who has seen the great seas,

and difficult to be clouds for one who has seen the high mountains.

献给挚爱，

曾经沧海难为水，

除却巫山不是云。
Chapter 1

Introduction

There are colossal, complex and undetermined threats in the cyber world. As cyber-attacks are happening on a daily basis and could be launched against an enterprise network at any moment, more and more organizations have established Security Operations Center (SOCs) to coordinate the defenses against cyber-attacks [1].

When a security incident happens, the top three questions a SOC seeks to answer are: What attack has happened? Why did it happen? What action should be done? While a variety of software tools (e.g., security information management system, host-based security systems) and hardware equipment (e.g., network intrusion detection systems) have been deployed in today's enterprise networks to detect and correlate security-related events [2], real-world SOCs still rely on security analysts (and watch officers) to make decisions on "What should I do?". Due to several critical limitations (e.g., high false positive rates) of the deployed software tools and hardware equipment, autonomous intrusion response is not yet being adopted by SOCs [3].

From the perspective of “data to decisions,” the intrusion response decisions made by a SOC can be viewed as the main output of a particular human-in-loop data triage system [4]. Not surprisingly, how soon the right intrusion response decisions can be made heavily depends on the efficiency (i.e., avoid performing useless data triage operations) of the system's data triage operations [5]. Since there are a large variety of “sensors” monitoring an enterprise network, the enterprise's SOC will gather a huge amount of heterogeneous data coming from different types of data sources. Accordingly, a critical challenge faced by the SOC is that the massive data sources generate great demands on security analysts' capability of information processing and analytical reasoning.
To address this critical challenge, SOCs have been putting in a lot of effort to recruit and train security analysts. However, it is widely observed that the amount of time and effort required to train a security analyst is overwhelming. It usually takes a newly hired security analyst several years to complete his or her training and become an experienced analyst. Moreover, during the long on-job training process, it is observed that most inexperienced (junior) security analysts perform much less efficiently than senior analysts in deciding what data triage operations to perform.

To address these training challenges, several retrieval methods have been proposed to facilitate the data triage of inexperienced security analysts through retrieval of the relevant past data triage operations of experienced (senior) analysts. These research works have shown that data triage operation retrieval could help an inexperienced security analyst a lot in reducing the number of useless triage operations during his or her data triage processes.

In this thesis, we first conduct a review of the existing retrieval methods, including experience-based retrieval and context-driven retrieval of data triage operations. We then discuss the new directions (e.g., apply machine learning techniques) in solving the data triage operation retrieval problem.

The remainder of this paper is organized as follows. In Section 2, we present an overview of data triage in SOCs. In Section 3, we give an overview of data triage operation retrieval systems. In Section 4, we discuss the main challenges in developing effective triage operation retrieval systems. In Section 5, we conduct an evaluation of the data triage operation retrieval methods, namely, experience-based retrieval and context-driven retrieval of triage operations. In Section 6, some related work will be discussed. In Section 7, we direct some future directions in building better triage operation retrieval systems. We conclude the paper in Section 8.
Chapter 2

Triage Analysis in SOCs

We define the triage analysis as a dynamic Cyber-Human System (CHS) evolving over time in this section. We mainly describe the details of the input data sources and the analysts' operations performed by analysts in the process of triage analysis and explain the challenges faced by the analysts.

The definition of triage analysis lays the base for understanding the work of developing the knowledge retrieval systems described in the following sections.

Although Security Operations Centers (SOCs) are increasingly rely on automatic approaches [7], the complicated network incidents cannot be discovered and evaluated without the efforts of cyber security analysts. In this section, we first introduce the triage analysis performed by analysts in SOCs and point out the need for leveraging the human experience to improve incident detection to achieve cyber situational awareness.

Unfortunately, there are few researches on how to leverage human skills to retrieve SOCs data and improve automatic methods' accuracy.

Why SOC still is human in the loop? Because the cyber analysts are the most important decision-maker among SOCs [8]. Analysts collect, discover, analysis, and respond all kinds of potential attacks [9]. The problem is that although senior analysts can accomplish their work successfully, these analysts may not explain all details precisely. While, these details are significant to train junior analysts.

We did a series of researches on SOCs data retrieval based on following reasons. The benefits of the retrieval system are two-fold. First of all, given the retrieved operations performed by other senior analysts, a junior analyst can learn what could be an effective data triage operation
to conduct as a next step in the current context. Secondly, the retrieved operations can provide valuable insights into how to interpret the abnormal network events and their relationships so that the junior analyst can gain a deeper understanding of the current context. In the current SOCs, junior analysts can obtain such assistance by directly communicating with the senior analysts or even by working under the supervision of senior analysts. However, it is challenging for senior analysts to find sufficient time to work together with junior analysts to offer timely assistance. Therefore, the retrieval system is developed to offer junior analysts immediate assistance in a more cost-efficient way. We have found little prior work specific to the information retrieval on data triage operations to assist analysts. However, we noted several areas of related work that are of interest in our work. Generally, these works can be classified into two categories: graph-based information retrieval approaches and cyber security operations centers.

2.1 Data Triage for Cyber SA

We demonstrate the human-in-the-loop process of the triage analysis in a SOC. The goal of the cyber security analysts is to detect the potential attack chains. Given the data sources collected by multiple sensors, an analyst conducts a series of data triage operations to rule out the false alerts or unrelated reports [10]. Therefore, we define the data triage process as a dynamic Cyber-Human System (CHS), which includes the following components: (1) the attack chains, (2) the network monitoring data collected from multiple sources, (3) a collection of incident reports which concludes the analysts' findings, (4) a collection of domain knowledge and experience knowledge, (5) the data triage operations performed by the analysts for accomplishing data triage, and (6) the hypotheses generated by analysts based on the existing findings about the potential attack chains (i.e., the mental model of analysts). Next, we explain the data sources and analysts' data triage operations in details.
2.2 Multi-Source Data in SOCs

SOCs usually deploy multiple cyber security defense technologies to protect an organization's network (such as intrusion detection systems (IDS) and firewall) [2]. The network connection activities are being monitored and controlled by these defense technologies over time. These network monitoring data collected from multiple sources usually have a high noise-to-signal ratio and are changing rapidly in the dynamic network environment. The common data sources include the alerts generated from intrusion detection/prevention systems (IDS/IPS), firewall logs, server logs, network status reports, vulnerability scanning reports, anti-virus reports, traffic packages, and so on.

Going through the automatic data cleaning, aggregation, and correlation, the data sources will be further provided to the analysts to identify the key evidence of potential cyber-attacks so that they can reason about the potential attack chains [11]. Therefore, such multi-source data are the input of the data analysis process of human analysts.

The multi-source data collected from the cyber defense technologies can be represented by a collection of network connection events. These events can be further ordered according to their occurrence time. Therefore, the multi-source data can be represented as a sequence of network connection events, part of which are indicators of the ongoing attack activities and the remaining are the benign network activities.

Each network connection event can be defined by a vector that specifies the attributes of a network connection:

\[ e = \langle t, type, ip_s, port_s, ip_d, port_d, protocol, source, severity, conf, msg \rangle \quad (1) \]

where \( t \) is the occurrence time of the event; \( type \) is the type of network connection (e.g., built, teardown and deny); \( ip_s \) and \( port_s \) are the IP address and port of the source, respectively; \( ip_d \) and \( port_d \) are the IP address and port of the destination, respectively; \( protocol \) is the network
protocol; source is the data source; severity and conf specify the level of severity and confidence of the event, respectively; msg specifies other important characteristics of the event, determined by the sensor

### 2.3 Data Triage Operation

The data triage of the network monitoring data refers to the process where an analyst conducts a sequence of data triage operations to filter and correlate the suspicious network connection events. To accomplish a data triage task, an analyst needs to iteratively search and identify the suspicious events from the raw data, to interpret the suspicious events, and to generate hypotheses about potential attack chains based on the existing observation, and to search for supporting/denying evidence if a hypothesis needs to be further investigated [12]. There are in general three types of operations performed during data triage:

- **FILTER**: filtering based on a condition.
- **SELECT**: identifying a subset of suspicious events.
- **SEARCH**: searching according to keywords.

As a result, the data triage analysts conclude his/her hypotheses about the possible attack chains with the evidence found in the raw data sources in the incident reports. Therefore, one main output of the triage analysis is the updates of the collection of incident reports.
Chapter 3

Problem Overview

3.1 Difficulties in Data Triage Tasks

The primary challenge faced by most SOCs is the gap between increasing data collected by cyber defense technologies and the limited resources of expert analysts.

Security analysts face several major difficulties in conducting their data triage tasks. First of all, the raw data from multiple sources has a large volume and very high noise-to-signal ratio. It has been impossible for analysts to go through all of them in details. Besides, considering the time pressure, analysts need to be highly concentrated on the task. Analysts need to decide whether or not a cyber event is suspicious or benign in minutes. Even worse, more and more cyber-attacks have multiple steps to achieve their ultimate goal, which make detection harder.

Last but not the least, the training of analysts always requires long-time on-the-job training. It is usually found that experts may not be able to explain the practical knowledge and their strategies precisely, although they are able to accomplish the tasks.

3.2 Experts' Knowledge of Data Triage

Analysts' experience and domain knowledge play a critical role in accomplishing data triage tasks. There have been several cognitive task analysis (CTA) studies conducted to investigate the working procedure of triage analysis. D'Amico et al. studied the main data sources and workflow of triage analysis. Analysts are good at interpreting data, comprehending contexts,
generating hypotheses and drawing conclusions through a complicated analytical reasoning process. Therefore, it is desirable to elicit experts' knowledge from their past data triage operations.

3.3 A Framework for Data Triage Knowledge Retrieval System Designs

This section presents a framework for data triage knowledge retrieval system. The system maintains a triage operation trace collection which manages all the data triage operations performed by experts for solving previous data triage tasks. A novice analyst is working on the triage of the incoming data sources. The analyst can directly create a query based on his/her attention of interest. Otherwise, his/her operations can be tracked in order to automatically construct a query based on the current context. Given a query, the operation retrieval engine will search for relevant operation traces in the trace collection and rank the results according to the relevance. The relevance can be determined by the similarity of the contexts. The retrieval result will then be presented to the analyst as a next-step suggestion.

The benefits of a retrieval system can be two-fold. First of all, a junior analyst can learn what could be effective data triage operations to conduct in the current context, if he/she is provided with the retrieved operations performed by other senior analysts in similar situations.

Secondly, the junior analyst can learn how to interpret the suspicious network events and how to generate the valuable hypotheses for further investigation.

Considering that most junior analysts are currently working under the supervision of senior analysts for guidance, a retrieval system can offer immediate and relevant suggestions in a more cost-efficient way. We have found little prior work specific to the information retrieval on data triage operations to assist analysts. However, we noted several areas of related work that are of interest in this work, which will be described in the next section.
3.4 Challenges in Developing Effective Data Triage Knowledge Retrieval Systems

The unique characteristics of how a SOC operates lead to several notable challenges in developing effective data triage operation retrieval systems. These challenges are as follows.

1. The nature of data triage operation retrieval is Knowledge Retrieval, not Information Retrieval. Knowledge representation plays an essential role in triage operation retrieval, but not in standard information retrieval systems. Accordingly, the existing information retrieval techniques, including text retrieval and web (page) retrieval techniques, could not be directly applied to solve the data triage operation retrieval problem.

The subject of the data triage operation retrieval is the practical knowledge gained by analysts from experience. Such tacit knowledge has been represented in an explicit format that a system can manage.

A good representation of such knowledge needs to incorporate the key components in analysts' analytical reasoning processes. For example, a conceptual AOH model of an analyst's analytical reasoning process includes: (A) actions performed by the analyst to filter and correlate the provided data sources; (O) observations of suspicious network events gained by performing actions; (H) hypotheses of the potential attack chains generated based on the existing observations.

2. The specific knowledge representation needed by data triage operation retrieval cannot be directly handled by existing knowledge retrieval systems. First, one unique characteristic of how a SOC operates is that there are a large variety of data sources (e.g., over 100 log files are collected from each host) are involved in data triage. Such amount of heterogeneity is usually not assumed in existing knowledge retrieval systems. For example, rule-based logic representations are generally used to represent knowledge, but the highly formalized structure makes this kind of representation limited to handle the aforementioned heterogeneity. Second, the data triage
knowledge representation in a SOC has domain-specific characteristics which cannot be handled by generic knowledge retrieval systems.

3. Data triage knowledge inherently covers a large amount of analytical reasoning conducted by security analysts, and the analytical reasoning in a SOC has domain-specific characteristics. Given that a common challenge of developing a knowledge retrieval system is to make the system domain-specific, data triage operation retrieval systems face the same challenge. This challenge will affect both knowledge representation and the retrieval algorithms. It is necessary to develop retrieval systems that can handle both the task operation information (i.e. actions and observations) and the analyst's mental processing (i.e., hypothesis).

4. A new challenge which is faced by a SOC but is not addressed in other knowledge retrieval systems is that data triage operations are being retrieved in adversarial settings. That is, the attacker may purposely obfuscate their attack actions in such a way that the accuracy of triage operation retrieval could be significantly reduced. How to make the retrieval system resilient to such adversarial obfuscation is a new challenge. Since keyword-based retrieval is usually not really resilient, it is important to incorporate semantics in triage operation retrieval.
Chapter 4

**Deep Learning based Retrieval of Triage Operations**

Due to the following observations, machine learning could play an essential role in developing better data triage operation retrieval systems.

First, the methods we have discussed in the previous sections make use of pre-determined similarity measurements when checking which historical data triage operations are most relevant to the current cyber situation. However, there is no guarantee that the pre-determined similarity metrics are the most suitable. Machine learning could be leveraged to help learn the most suitable similarity metrics [13].

Second, data triage operation retrieval systems must be able to handle a variety of uncertainties such as the uncertainty introduced by false positives, false negatives, and incomplete information.

Machine learning could be leveraged to increase retrieval systems' capability in dealing with the uncertainties. Machine learning, especially neural networks, is a potential approach, which can be used for data triage operation retrieval in a SOC. There are a variety of artificial neural networks, such as convolutional neural networks, long short-term memory, and deep belief networks [14]. Instead of providing a comparative viewpoint, below we only discuss the potential application of recurrent neural networks.

### 4.1 Properties in Recurrent Neural Networks

RNNs are learning machines that recursively compute new states by applying transfer functions to previous states and inputs [15]. Typical transfer functions are composed by an affine
transformation followed by a nonlinear function [16], which are chosen depending on the nature of the particular problem at hand [17]. RNNs possess the so-called universal approximation property [18], that is, they are capable of approximating arbitrary nonlinear dynamical systems (under loose regularity conditions) with arbitrary precision [19], by realizing complex mappings from input sequences to output sequences [20]. However, the particular architecture of an RNN determines how information flows between different neurons and its correct design is crucial for the realization of a robust learning system [21]. In the context of prediction, an RNN is trained on input temporal data $x(t)$ in order to reproduce a desired temporal output $y(t)$. $y(t)$ can be any time series related to the input and even a temporal shift of $x(t)$ itself [22]. The most common training procedures are gradient-based, but other techniques also have been proposed [23].

### 4.2 Data Triage Operation Retrieval based on Recurrent Neural Networks

For data triage operation retrieval, the most promising neural networks approach seems to be recurrent neural networks (RNN), mainly because this type of neural network is good at dealing with sequence data. One of the most notable features in data triage operations is that security-related events are sequential.

The fundamental philosophy behind RNN models is that rather than rewriting all information, each element in an RNN model updates the current state by adding new information [24]. Accordingly, when an RNN is trained to classify the newly arrived data triage operations, the RNN can be incrementally maintained to incorporate substantial new data triaging knowledge.

But, before training and deploying any RNNs in a SOCs, the SOC should cautiously consider the potential adversaries.

A new challenge which is faced by a SOC but is not addressed in other knowledge retrieval systems is that data triage operations are being retrieved in adversarial settings. That is, the attacker
may purposely obfuscate their attack actions in such a way that the accuracy of triage operation retrieval could be significantly reduced.

Recently, substantial research work has shown that most existing machine learning classifiers are highly vulnerable to adversarial examples.

The RNNs deployed in a SOC should be resilient to adversarial examples.

### 4.3 Data Triage Model

Cyber security data triage is targeted at determining whether the incoming data sources are worth of further investigation in a timely and quick manner. To achieve this goal, security analysts usually conduct a sequence of data triage operations to filter malicious network events and to group them according to the potential attack chains. Therefore, the unit of data triage analysis is a network event. Network events are the data reported by various network monitoring sensors, including SIEM tools and human intelligence agents,

A *network event* can be abstracted as a multi-tuple of its characteristics,

\[
e = \langle t_{\text{occu}}, t_{\text{detect}}, \text{type}, \text{attack}_{\text{prior}}, \text{sensor}, \text{protocol}, \text{ip}_{\text{src}}, \text{port}_{\text{src}}, \text{ip}_{\text{dst}}, \text{port}_{\text{dst}}, \text{severity}, \text{confidence}, \text{msg} \rangle,
\]

where \( t_{\text{occu}} \) is the time the event occurred; \( t_{\text{detect}} \) is the time the event first being detected; \( \text{type} \) is the type of network connection activity (e.g., Built, Teardown or Deny); \( \text{attack}_{\text{prior}} \) is the attack type of the event being detected by a sensor/agent based on prior knowledge; \( \text{sensor} \) is the sensor/agent who detected this event; \( \text{protocol} \) is the network protocol; \( \text{ip}_{\text{src}}, \text{port}_{\text{src}}, \text{ip}_{\text{dst}}, \text{port}_{\text{dst}} \) are respectively the source IP, source port, destination IP, and destination port; \( \text{severity} \) and \( \text{confidence} \) specify the level of severity and confidence of the event, respectively; \( \text{msg} \) specifies other characteristics of the event, which depends on the sensor.
An example of a data triage process is an analyst performs a sequence of data triage operations to identify suspicious network events. Each data triage operation specifies a constraint for the events to narrows down the original data set. As the examples shown in Table 1, there are mainly three types of data triage operations:

- **FILTER** \((D, C)\): to filter a set of events \((D)\) based on a constraint \((C)\).
- **SEARCH** \((D, C)\): to search a keyword \((C)\) in an event set \((D)\).
- **SELECT** \((D, C)\): to select a subset of events with a common feature \(C\) from a set \((D)\).

**Table 1 Examples of data triage operations**

<table>
<thead>
<tr>
<th>Data Triage Operations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILTER</td>
<td>a set of network events (i.e., IDS Alerts) based on a condition ((\text{SrcPort}=6667))</td>
</tr>
<tr>
<td>SELECT</td>
<td>a set of network events (i.e., the underlined firewall log entries) with a common characteristics ((\text{DstPort}=6667))</td>
</tr>
<tr>
<td>SEARCH</td>
<td>in the Firewall log based on the condition ((\text{SrcIP} = 172.23.233.52 \text{ OR DstIP} = 172.23.233.52)).</td>
</tr>
</tbody>
</table>

All these operations result in a subset of events satisfying a constraint. The constraint is defined as follows.
The analyst specified several criteria of the suspicious or correlated network events based on the domain knowledge and experience. Each criterion specifies a constraint on the network event characteristics, so that a data triage operation can select and correlated network events.

### 4.4 Triage Operations through Time

Gradient-based learning requires a closed-form relation between the model parameters and the loss function. This relation allows to propagate the gradient information calculated on the loss function back to the model parameters, in order to modify them accordingly. While this operation is straightforward in models represented by a directed acyclic graph, such as a Feedforward Neural Network (FFNN), some caution must be taken when this reasoning is applied to RNNs, whose corresponding graph is cyclic. Indeed, in order to find a direct relation between the loss function and the network weights, the RNN has to be represented as an equivalent infinite, acyclic, and directed graph. The procedure is called unfolding and consists of replicating the network’s hidden layer structure for each time interval, obtaining a particular kind of FFNN. The key difference of an unfolded RNN with respect to a standard FFNN is that the weight matrices are constrained to assume the same values in all replicas of the layers, since they represent the recursive application of the same operation.

Training a neural network commonly consists of modifying its parameters through a gradient descent optimization, which minimizes a given loss function that quantifies the accuracy of the network in performing the desired task. The gradient descent procedure consists of repeating two basic steps until convergence is reached. First, the loss function $L_k$ is evaluated on the RNN configured with weights $W_k$, when a set of input data $X_k$ are processed (forward pass). Note that with $W_k$ we refer to all network parameters, while the index $k$ identifies their values at epoch $k$, as they are updated during the optimization procedure. In the second step, the gradient
\( \frac{\partial L_k}{\partial W_k} \) is backpropagated through the network in order to update its parameters (backward pass).

### 4.5 Data Triage Operation and Characteristic Constraint

An atomic constraint predicates the value of an event characteristic/attribute,

\[ T_i = R_i(char, val), \]  

(3)

Where \( R_i = \{=, \lt, >, \leq, \geq\} \).

Considering the fact that an event possesses multiple attributes, the constraint can be multidimensional and be represented by a predicate in disjunctive normal form, named *Characteristic Constraint*,

\[ C = \{ \lor (\land T_i) \}_{i \in \mathbb{N}} \]  

(4)

Given the definition of characteristic constraint, a *data triage operation* is defined as follows.

\[ O_i = < D, t, C >, \]  

(5)

where \( D \) is a set of network events; \( t \) is the time of being performed; \( C \) is the characteristic constraint specified for data filtering.

### 4.6 Trace and Context

A *trace* consists of a sequence of data triage operations performed by an analyst in accomplishing a data triage task. It can be represented by \( T = (O_i)_{i \in \mathbb{N}} \), where \( O_i (1 \leq i \leq n) \) is a data triage operation.
Given a data triage operation $O_i$ in a trace $\mathcal{T}$, the context of $O_i$, denoted by $\mathcal{C}(O_i)$, is defined by the sequence of the data triage operations that precede $O$ and their relationships. The relationships between data triage operations consist of temporal and logic relationships, which are defined as follows.

Let $O_1 = (D_1, t_1, C_1)$ and $O_2 = (D_2, t_2, C_2)$ be two different data triage operations.

The temporal relationship between them is determined by $t_1$ and $t_2$. We have "happen-before" or "happen-after" relationships, denoted by $\prec_t$ and $\succ_t$ respectively.

\[
\prec_t (O_1, O_2) \iff t_1 < t_2, \quad \succ_t (O_1, O_2) \iff t_1 > t_2 \tag{6}
\]

The logic relationship between $O_1$ and $O_2$ is determined by their characteristic constraints. Three types of logical relationships, “is-equal-to”, “is-subsumed-by”, and “is-complementary-with”, can be defined as follows.

\[
isEql(O_1, O_2) \iff C_1 \leftrightarrow C_2, \tag{7}
\]
\[
isSub(O_1, O_2) \iff C_1 \rightarrow C_2, \tag{8}
\]
\[
isCom(O_1, O_2) \iff C_1 \rightarrow \neg C_2 \text{ and } C_2 \rightarrow \neg C_1, \tag{9}
\]

where $\rightarrow$ means “implies”. $isEql(\cdot, \cdot)$ and $isCom(\cdot, \cdot)$ are bidirectional relationships but $isSub(\cdot, \cdot)$ is a unidirectional relationship. For example, if $\mathcal{C}_1$ is “SrcPort = 21” and $\mathcal{C}_2$ is “SrcPort $\neq$ 6667”, then we have $isSub(O_1, O_2)$. If $\mathcal{C}_3$ is “SrcPort = 22” and $\mathcal{C}_4$ is “SrcPort = 6667”, we have $isCom(O_1, O_3)$ and $isCom(O_1, O_4)$.

Therefore, the context of the data triage operation $O_i$ can be defined as,

\[\mathcal{C}(O_i) = (O_j), \{R_T(O_j)\}, \{R_L(O_j)\}, j < i \tag{10}\]
where \((O_j)_{j<i}\) is the set of data triage operations conducted earlier than \(O_i\), \(R_T\) and \(R_L\) refer to the temporal and logic relationships among \((O_j)_{j<i}\) respectively.

An analyst makes decisions on what data triage operations to perform mainly on the context.

The current context of an analyst's data triage process refers to the context of the latest data triage operation, which changes dynamically as long as the analyst performs new operation. Therefore, it is critical to take the context into consideration in order to retrieve the relevant traces.

4.7 Insights: Context-Driven and Efficient

Our goal is to retrieve the relevant traces based on the current context of an analyst's data triage process and to quickly update the results along with the context change. Our approach is developed based on two main insights.

Insight 1: Recurrent neural network can be used to represent the context information because it can perfectly capture the course of data triage operations and the dependencies between these operations.

Insight 2: The retrieval results need to be updated dynamically along with the changes of the current context. It requires the graph-based approach to be efficient enough for timely updates. Therefore, we need to avoid graph isomorphism analysis.

Inspired by these two insights, we adopted a deep learning approach. The concept of neural network originally comes from biological brain, which composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.
4.8 Challenges in Using Machine Learning for Data Triage Operation Retrieval

Machine learning has been playing an increasingly important role in performing various tasks in SOCs [25]. For example, network intrusion detection systems [26] and malware classification systems are leveraging more and more automation achieved through machine learning [27, 28].

However, although machine learning is good at (dealing with) average cases [29, 30], it is not easy to implement any machine learning methods for data triage operation retrieval systems [31], since data triage operation retrieval systems are related to worst cases [32, 33]. It is possible to bypass a machine learning based content filter through malicious manipulations in adversarial settings [34]. The attacker could combine malicious samples with benign events to evade several retrieval classifiers [35, 36]. For example, some very small manipulations [37-39] in events logs can lead to distinct opposite results in data triage operation retrieval systems [40]. It is not an easy task to guarantee accuracy and sensitivity simultaneously. In data triage operation retrieval, because of the inherent temporal relationships between events, the adversary has the possibility to infer the similarity metrics to bypass the retrieval system [41].
Chapter 5

Evaluation

We first prepared a trace collection that comes from a previous experiment. In that experiment, we recruited thirty professional analysts and asked them to complete a cyber data triage task. Each analyst's data triage operations are tracked automatically. It has been shown that these collected traces can represent the analysts' analytical reasoning process.

We built a Python [42] and Java prototype to retrieve relevant traces from the collection traces. Given the input of a current context (i.e., a sequence of trace operations), the system outputs the matched traces in the trace collection.

Our method has been evaluated by answering the following questions. (E1) Can the retrieval system find the matched traces for cyber security analysts? (E2) What factors may influence the performance of the retrieval methods? (E3) Do cyber security analysts find the retrieval output useful?

A collection of data triage traces (n=50) have been collected in a previous experiment involving cyber security professionals. To evaluate the retrieval method, we first generate testing cases by conducting subsampling on the data triage trace collection. The ground truth of the matched traces is decided through a two-round manual trace analysis conducted by two researchers.

Several measures are used, including precision, recall, accuracy rate and F-1 measure, to compare results generated by our approach with the ground truth. We further investigate what

In order to prepare the ground truth, we simplify the problem to a binary classification. Then, two analysts annotate traces which are match or not independently.
We use subsampling to generate 330 experiment cases. For these 330 pairs, two analysts labeled each pair whether or not are matched, respectively. The results of 304 pairs are Consistent between the two analysts and these pairs are the ground truth for the experiments.

After that, our approach labeled the pairs automatically. To evaluate the results, we will detail precision, recall and F-score, etc.

5.1 Test Cases Generation

Although the subjects are professional analysts, their task performance turned out to be quite different.

In order to eliminate the interference of the quality of the traces on the performance of the retrieval system, the traces were evaluated carefully with the consideration of task performance, trace quality and the diversity of traces: (1) all the selected traces come from the analysts who had successfully revealed the attack events in the task; (2) each analyst had conducted a series of data triage operations during their exploration that are sufficient for understanding the analysis strategies of the analyst; and (3) these traces embody various analysis strategies used by the analysts. At last, we selected five traces in this way. The average number of data triage operations in the selected traces is 23.2.

Considering the richness of a single selected trace, we use subsampling to generate a large set of representative trace slices. To maintain the important temporal relationships between the data triage operations, we set a sliding window and extract a sequence of successive data triage operations as a slice, and then move the window to a next \( l \) operation (\( l \) is called the hop length). Let \( m \) be the window length, \( l \) be the hop length, given a trace with \( n \) data triage operations (\( n>m \)), \((n-m+1)/l\) slices can be generated.
To generate trace slices, we set the window length 8 and the hop length 3 in consideration of three aspects: (1) a trace slice should contain sufficient number of DT operations and most slides involving multiple DT relationships have a length no less than 8; (2) however, a trace slice can't be too complicated for analysts to get a controversial understanding of the real analytic process; and (3) most frequently, 3 successive DT operations can be highly related and similar, and therefore we use 3 hops to increase the distance between the slices.

Altogether we extracted 29 trace slices. 330 pairs were further generated by combining each two slices. But an exclusion is that the slices extracted from a same trace cannot be paired as it will be meaningless to consider the similarity of the slices extracted from the same trace. 330 pairs in total are considered as the testing cases. Although the quantity of the testing cases is not large, the testing set overall contains sufficient information because each testing case was an informative and self-contained piece implying a data triage process.

We selected 5 traces collected from the analysts who had high-level task performance. These traces also contain the most data triage operations among the trace collection (the average number of data triage operations is 23.2).

5.2 Ground Truth

The ground truth refers to the fact that whether two slices matches. To determine the ground truth, we have two experts to manually label the testing cases, named Coder A and Coder B. The experts were asked to make decisions based on their interpretation of the data triage processes. To guarantee the accuracy of their decisions, we made sure that the two experts have sufficient expertise in cyber security analysis and were familiar with the trace representation and the data triage task ground truth. Besides, two rounds of manual analysis have been conducted to ensure the quality of their decisions. During each round of analysis, Coder A and B analyzed and labeled the
pairs independently. At the end of the first round of analysis, Coder A labeled 129 pairs as positive (i.e., matched) and 202 pair as negative (i.e., not matched). Coder B labeled 163 positive pairs and 168 negative pairs. They agreed on 205 pairs among the 330 pairs. During the second round of analysis, Coder A and B went through the inconsistent pairs independently without any information of their previous labels or the other's labels. At the end of the second round, Analyst A and B agreed on 304 pairs in total, including 126 positive ones and 178 negative ones. The inconsistent pairs were discarded.

5.3 Random Selection

To ensure the randomness, we shuffled the positive and negative pairs, and randomly select 100 pairs respectively from each set (positive pairs and negative pairs). In this way, we can construct a balanced testing set with 100 positive pairs and 100 negative pairs. In the following testing, we repeated the random selection for 10 times and evaluated the average performance of our method.

5.4 Performance Measurement

To answer the first evaluation question (E1), we first evaluate the performance of the retrieval system. We counted the True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), and used them to determine the following measurements: (1) Precision, the percentage of the pairs predicated as positive that actually are matched; (2) Recall, the percentage of the matched pairs that are predicated as positive; (3) Accuracy (ACC), the percentage of correct predictions; and (4) F-1 Score, the harmonic mean of precision and recall. They are calculated as follows.
\[ \text{Precision} = \frac{TP}{TP + FP} \]  
(11)

\[ \text{Recall} = \frac{TP}{TP + FN} \]  
(12)

\[ \text{ACC} = \frac{TP + FN}{TP + TN + FP + FN} \]  
(13)

\[ F_i = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \]  
(14)

The results are shown in Table 2 and Figure 1.

Table 2: Experiment results

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.91</td>
<td>0.68</td>
<td>0.78</td>
</tr>
<tr>
<td>2</td>
<td>0.88</td>
<td>0.71</td>
<td>0.79</td>
</tr>
<tr>
<td>3</td>
<td>0.85</td>
<td>0.75</td>
<td>0.80</td>
</tr>
<tr>
<td>4</td>
<td>0.82</td>
<td>0.76</td>
<td>0.79</td>
</tr>
<tr>
<td>5</td>
<td>0.79</td>
<td>0.81</td>
<td>0.80</td>
</tr>
<tr>
<td>6</td>
<td>0.75</td>
<td>0.83</td>
<td>0.79</td>
</tr>
<tr>
<td>7</td>
<td>0.74</td>
<td>0.86</td>
<td>0.80</td>
</tr>
<tr>
<td>8</td>
<td>0.72</td>
<td>0.85</td>
<td>0.78</td>
</tr>
<tr>
<td>9</td>
<td>0.71</td>
<td>0.89</td>
<td>0.79</td>
</tr>
<tr>
<td>10</td>
<td>0.66</td>
<td>0.9</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Average 0.78 0.80 0.79
To further investigate the trade-off between the precision and recall, we show the RP-Curve in Figure 2. Recall, as shown on the x-axis, is identical with sensitivity of our algorithm, and precision, as shown on the y-axis, is identical with the positive predictive rate. We notice that precision remains relatively stable when recall is in the range roughly between 0.7 to 0.8.
5.5 Human Evaluation

Comparing the outputs with the ground truth, we have shown that the RNN method can be tuned to get a relatively high precision or recall rates. To further evaluate the usefulness of the retrieval results, we asked two human judges to manually rate the outputs during the process of data triage.

5.5.1 Evaluation Protocol

The ideal experiment is to have the human subjects perform a real data triage task with the assistant of the retrieval system and rate the retrieval outputs in each trial. However, it requires the subjects' domain knowledge and expertise. Besides, this step-by-step evaluation takes a lot time per trial (about 30-60 minutes are needed to complete a data triage task). However, professional analysts with the expertise cannot afford such a long time. To make the human evaluation feasible, we made two adjustments. Due to the limited access to professional analysts, we invited two graduate students who has the expertise of cyber security analysis and had participated in our previous experiment as the judges. To reduce the evaluation time in each trial, we prepared a data triage context for the judges in each case so they could situate themselves in a scenario and make a decision without performing the task from the beginning. The details are described below.

For each trial, we randomly selected one slice (i.e., a sequence of data triage operations) from the trace collection to be the current context and presented it to a judge. The judge first read through the slice to interpret the current context.

The retrieval system took the current context as a query and suggested the retrieved slices. The judge needed to manually go through each retrieved slice to make a decision on the usefulness
of the suggestion. A 5-point Likert scale was used: 1 (strongly negative), 2 (negative), 3 (neutral), 4 (positive), 5 (strongly positive).

5.5.2 Sequence Length

There are some tradeoffs on sequence length. We considered that if the sequences are too short, then there will be too many sequences similar to the original sequences. While, if the sequences are too long, then on the one hand, no sequences may be similar to the original sequences. On the other hand, judges do not have enough time to check long sequences. Based on the above reasons, we try to choose sequences including about four to ten data operations, respectively. There are 1000 traces for each specific length.

As shown in Table 3, Slices with length 8 is better than other slices.

Table 3 Satisfaction rate for sequence length

<table>
<thead>
<tr>
<th>Sequence Length</th>
<th>Satisfaction Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>52.1%</td>
</tr>
<tr>
<td>5</td>
<td>70.6%</td>
</tr>
<tr>
<td>6</td>
<td>81.5%</td>
</tr>
<tr>
<td>7</td>
<td>85.1%</td>
</tr>
<tr>
<td>8</td>
<td>91.2%</td>
</tr>
<tr>
<td>9</td>
<td>83.8%</td>
</tr>
<tr>
<td>10</td>
<td>43.2%</td>
</tr>
</tbody>
</table>
5.5.3 Result

In total, we collected 1934 responses from the 2 judges. The average time used by the judges for making a decision per trial are respectively 24.36 seconds (Judge 1) and 56.16 seconds (Judge 2). Besides, the two judges were consistent with their responses: we compared the judge's response in the repeated cases; we found all the responses of both judges were consistent in terms of positive and negative ratings, and the average of the rating difference is 0.328 and 0.317 respectively.

The statistical description of the responses is shown that the majority responses are positive (the overall mean is 4.1 and median is 4).

5.5.4 Case Study

This section demonstrates two cases: one rated positive (i.e., Case 1) and the other rated negative (i.e., Case 2) by the judges. In Case 1, the query slice indicates that the analyst first identified a suspicious host and paid attention to the network connections targeting this host via port 6667. After that, he switched his attention to other suspicious network connections via port 21. After conducting the filtering based on port 21, he ruled out the network connections via port 6667 and 80, indicating he was searching for other suspicious activities. Comparing with the query slice, the retrieved slice on the right also indicated the analysts had conducted data filtering based on the port 21 and 6667 and detected the same set of network connections. Similarly, the analyst later filtered out these network connections and then detected a set of new network connections via port 22 targeting to the IPs in the range of 10.32.5.*. Based on our understanding of these two slices, it is worth considering the retrieved slice for an analyst who has been conducting the operations in the query slice but has no idea what to do next.
In terms of the negative Case 2, the query slice indicates the analyst first detected the network connections targeting to the IPs in the range of 10/32.5.* via port 21. After that, he identified one suspicious host 172.23.233.57 and further switched his attention back to the connection via port 6667. On the other hand, the retrieved slice suggested that another analyst also filtered the network connections via port 6667. However, the analyst switched his attention to the connections via port 445 back and forth instead of exploring more suspicious activities via port 6667. Therefore, presenting the retrieved slice can hardly help the current analyst.

The results of the case study provide insight into the improvement of our deep learning based retrieval method. We found in both cases the retrieved slices are relevant to the query slice. The difference between the positive case and the negative case lies in whether the retrieved slice matched the analyst's current focus of attention. Therefore, the analyst's focus of attention is a critical factor for consideration when matching traces. In fact, this factor has been implicitly incorporated in terms of calculation: we placed more weight on the operations that are more recent when calculating the `isSub` subgraph. Such weight allocation can be tuned by a system user to achieve a performance that satisfies the user's needs.
Chapter 6

Related Work

In this section, we review several data triage knowledge retrieval systems that were constructed under the retrieval framework proposed in previous section: a rule-based retrieval system and a context-based retrieval system. According to the challenges discussed in the previous section, we will mainly introduce the knowledge representation and matching algorithms of the knowledge retrieval systems.

6.1 Rule-Based Data Triage Retrieval System

Similarity matching is an important topic in many fields [43]. Chen et al. developed a knowledge-based intrusion detection approach, which using Horn rules to illustrate experts' experience. As shown in Figure 3, a large number of data filtered by intrusion detection systems. Coordination agents will determine the events with the potential relationship. Inference agents will decide the related events with specific rules. Most works focus on the layers rely on the results of data triage analysis. This work represented analysts' data triage knowledge of as logic rules and invented a rule relaxation approach to gain flexibility.
These evaluation criteria can be used to invent more robust retrieval models and to provide standardized benchmarks of systems' performance in data triage operation retrieval. Chen et al. provided an experience-driven framework.

In fact, it is not easy to monitor cyber events and discover attacks. Experiences are important in discovering attacks. Expert guidance is critical for monitoring network events. The challenge is how to represent experiences, specifically in SOC.

We will later introduce a systematic method to leverage analysts' experiences to improve cyber situation recognition and discover attacks.

6.2 Knowledge Representation

According to the retrieval framework (Section 2), analysts’ data triage knowledge is managed in the triage Operation trace collection. Each knowledge piece is represented by logic rules. An event-alert system $S$ is formalized as a 4-tuple $(E, A, C, T)$, where $E = E_1, \ldots, E_m$ is a finite set consisting of Event Types, $A = A_1, \ldots, A_2$ is a finite set consisting of Alert Types, $C = C_1, \ldots, C_o$ is the causality relationship hyper edges between Event Types. $T = T_1, \ldots, T_2$ is links about Event Type to Alert Type. A partially observable event-alert system $S$ is a system where all
alert events are observable but may be hidden from the users. These hidden events can still be indirectly observed through context. Because alerts are observable, while events are unobservable, the runtime information is an alert sequence.

Given a partially observable event-alert system $S = (E, A, C, T)$, there is an alert sequence $q = <a_1, \ldots, a_n>$ being generated at run-time. Each instance of the alert $a_i = [7 tE ]$ contains these information:

- $T_A$: the alert instance’s type;
- $t_A$: the time stamp of this alert becoming available;
- $T_E$: the type of the event;
- $t_E$: the time stamp of the hidden event occurs.

### 6.3 An Example of Rule-Based Representation

In this section, to help understand experiences in data triage operation retrieval, an example is provided to illustrate the core idea of hierarchical experience representation. An attack graph can be constructed to show its vulnerabilities and their dependencies. Figure 4 explains the important features from an attack graph. Firstly, the upper part of the graph is a list of alert types. These alert types are observable to analysts. In addition, each alert contains information about its triggering event. In this survey, we use dashed line to represent this relationship. The events are often hidden from the analysts. Lastly, several events are linked by their causal relationships. These causal relationships infer a typical temporal order of alerts.
Figure 4 The critical features in an attack graph.

6.4 Knowledge Capturing

Before retrieving analyst's experience, it is necessary to capture experience to construct the knowledge base. Chen et al. identified the following important properties of cyber situation recognition:

- Events type;
- Events temporal relationships;
- Alert correlation information.

Based on them, Chen et al. use forward-changing rules stemming from Horn logic to illustrate experience patterns.

There are two patterns for each experience: event pattern and alert pattern.

Hidden events are important clues for data triage operation retrieval. Event pattern captures the hidden events, and the temporal orders among the hidden events indicate the causal relationship. Alert pattern captures the observable alerts. They are the clues discovered by analysts. Experiences with specific detail at a specific moment will be useful.
6.5 Knowledge Matching and Rule Relaxation

Given the rule-based representation, a past incident can be described by a rule condition, which includes every specific detail at that moment, such as the time slot and the geographical location of the events. Therefore, an experience will not repeat itself with each same single details.

As shown in the retrieval framework, the current context will be searched in the knowledge base. However, the rule matching requires every single detail of the rules to be matched, which may limit the usefulness of the retrieval results.

The problem is: how can we make a limited number of experiences useful for assisting to detect similar events? An experience will not be useful if we do not abstract the particular details. It is significant to retrieve the key parts of an experience and to relax the experience by choosing the portions, which are not too specific.

To make the rule matching more flexible, Chen et al. proposed rule relaxation based on the Horn clause representation.

In regard to rule-based representation, researchers can relax the constraints by removing conditions from antecedents of that rule.

Conditions with lower priorities can be relaxed. Experience includes specific details, such as the exact time and location of the incident. Therefore, an experience will not repeat itself with each single details remaining the same. It is desirable to retrieve the important parts of an experience and to relax the experience by trimming those portions that are too specific.

The higher the degree to which an experience can be relaxed, the higher the possibility exists that it can be matched against a new situation. Figure 5 shows that the knowledge generated by relaxation form a hierarchy: the most specific knowledge at the bottom while the top is the most relaxed ones.
Overall, upper-level experiences have better precision. While lower level experiences provide broader coverage. The entire experience hierarchy is formed through a consistent process, where each level of relaxation is defined with a specification guideline (i.e., how a higher-level experience should be relaxed into lower-level ones). All experiences on the same level will have a consistent specificity. According to Figure 6, rule matching is performed on each piece of knowledge in the network. Rule relaxation enables a larger set of matching candidates. Meanwhile, it may influence the precision of the results.
6.6 Case Study

We have discussed a systemic model to represent, capture, and relax experiences. In this section, for evaluating the concepts of experience-based cyber security analysis, Chen et al. implemented a cyber situation recognizer.

The rule-based retrieval system has been implemented and evaluated in a case study.

Figure 7 demonstrates the architecture of the system: the experience base is the collection of knowledge; the cyber security adapter takes in the network data (alerts); the recognizer performs the rule matching and rule relaxation by consulting the knowledge base and the rule system, and the matched results will be suggested to the user.
The cyber security adapter is responsible for receiving alerts and correlation sequences from external information sources. Then, these alerts are transformed into internal data structures. The reorganized information will be delivered to the recognizer, which will perform cyber situation recognition. Recognizer will consult the knowledge base, including experience base and rule processing system. After that, recommendations will be delivered to the end user, if an intrusion is discovered.

In the case study, Chen et al. evaluated the performance of the retrieval system by comparing the recommendations against the ground truth of a simulated data set. It showed that the rule representation made knowledge capturing possible. Besides, the rule relaxation makes the retrieval system more flexible and the analysts can adjust the coverage or precision of the matching results based on their needs.

This approach including two aspects. The first aspect is to capture experiences, relax experiences, and integrate newly captured and relaxed experiences into the hierarchical experiences. The second aspect is to perform situation recognition by feeding alerts and correlation sequences. The agent generated recommendations will be compared against the ground truth.

Overall, Chen et al. leverage a logic-based method to implement systematic capture of analyst's experiences. In addition, for utilizing few human experiences, the authors use experience relaxation to enhance the coverage of a set of patterns.
6.7 Context-Based Data Triage Knowledge Retrieval System

A context-based data triage knowledge retrieval system represents analysts' analytical reasoning processes in a tree structure. Given the structure-based knowledge representation, the context of an analytical reasoning process was further defined so that the similarity between two contexts can be measured. The retrieval results were ranked according to the similarity between them with the current context.

6.8 Knowledge Representation

According to the conceptual AOH model, an analyst's analytical reasoning process in data triage contains three types of components: actions, observations, and hypotheses: an action may trigger an observation; gaining an observation may let the analyst generate a hypothesis; the further investigation of the hypothesis requires further actions. Based on the conceptual model, a tree structure, named Experience Tree (E-Tree), represents actions, observations, hypotheses, and their relationships.

The nodes of an E-Tree are the instances of actions, observations, and hypotheses, and the edges are the relationships between them. The root of an E-Tree is the initial action or observation in the analytical process.

The context of a hypothesis is defined by the path in the E-Tree from the root to this hypothesis.

Figure 8 demonstrates an example of E-Tree: ''EU'' refers to a pair of action and its resulting observation. According to the context definition, the context of ''H4'' consists of ''Root EU1'', ''H1'', and ''EU2'. 
6.9 Knowledge Matching

Given the definition of context, the similarity measure was proposed to determine whether two pieces of knowledge (E-Tree) matches or not. Both base matching and weighted matching are used to calculate similarity.

Base Matching is the minimum criteria. For instance, Two E-Trees should come from the same data source. Weighted Matching is based on Base Matching. We can calculate the degree of matching through Weighted matching. A Match Propagation (MP) algorithm can efficiently rank E-Trees by similarity.

In summary, an AOH model to retrieves data triage operation. After representing analysts' experience as an experience tree, there are several approaches to retrieve data triage operations. For example, this work constructs indexes for retrieving efficiently.
Chapter 7

Discussion and Future Work

The existing studies introduced in the above section has demonstrated promising results for future studies. In this section, we propose several research directions for developing data triage knowledge retrieval systems.

Based on our observations of the two methods reviewed in the previous section, we propose three future research directions in this section.

1. Researchers may design more efficient retrieval algorithms specific for SOC's data.
2. Researchers may leverage machine learning methods to retrieve the data in SOC.
3. Researchers may enhance some state-of-the-art information retrieval approaches, such as metapath, to be used in SOC settings.

7.1 Graph-Based Data Triage Knowledge Retrieval System

According to the conceptual AOH model, an action is a data triage operation performed by an analyst to filter or to correlate network events, which usually specify a condition on the network events to narrows down the dataset. It is through conducting a series of data triage operations enables an analyst to find the critical "indicators" of potential attack chains [44]. Therefore, the analytical reasoning strategies used by an analyst are embedded in the relationships (both logical and temporal relationships) among the data triage operations. With this insight, a graph-based data triage knowledge retrieval system can be developed that represents and retrieves not only the analytical reasoning process but also the underlying logic and reasoning strategies used by analysts [45] [46].
This section presents a novel idea to implement retrieving systems in SOC. There are many algorithms for graph-based retrieval systems [47]. Before implementing these algorithms in SOC settings, the challenge is to design a knowledge representation model specific for data triage operation.

### 7.2 Knowledge Representation

Security operations centers are focused on whether or not the incoming data are worthy of further analysis. To accomplish this goal, analysts conduct a serious of data triage operations to discover malicious network events and to summarize these raw data to the potential attack chains. The data triage analysis is based on network events, which are from a variety of IDS, SIEM and human agents.

A network event can be represented as a multi-tuple:

\[
e = <t_{occur}, t_{detect}, type, attack_{prior}, sensor, protocol, ip_{src}, port_{src}, ip_{dst}, port_{dst}, severity, confidence, msg>,
\]

In the multi-tuple, \( t_{occur} \) is the event occurred time; \( t_{detect} \) is the event being detected time; type is the network connection activity's type; \( attack_{prior} \) is the attack type being detected; sensor is detection agent; protocol is the network's protocol; \( ip_{src}, port_{src}, ip_{dst}, port_{dst} \) are the source IP, source port, destination IP and destination port, respectively; severity is the level of severity; confidence is the level of confidence; msg is additional information for the event.

Recall that there are three types of data triage operations in SOC:

- FILTER \((D, C)\): to filter a set of events based on a constraint.
- SEARCH \((D, C)\): to search a keyword in an events group.
- SELECT \((D, C)\): to select a subset of events with a specific feature.
All these operations are performed to obtain a subset satisfying a specific constraint. Therefore, a constraint is a critical component in a data triage operation. A constraint specifies the characteristics of network events, indicating the analyst's focus of attention. The constraint can be multidimensional if multiple characteristics are specified. Therefore, a constraint can be represented by a predicate in disjunctive normal form.

The relationships between data triage operations include both temporal and logical relationships. An analyst performs data triage operations in a temporal sequence: one operation precedes the next one. The logical relationships between data triage operations are defined by the constraints specified in the operations. Let \( C_1 \) and \( C_2 \) be two constraints of operation \( O_1 \) and \( O_2 \) respectively, we have:

- if \( C_1 \leftrightarrow C_2 \), \( O_1 \) “is-equal-to” \( O_2 \);
- if \( C_1 \rightarrow C_2 \), \( O_1 \) “is-subsumed-by” \( O_2 \);
- if \( C_2 \rightarrow \neg C_2 \) and \( C_2 \rightarrow \neg C_1 \), \( O_1 \) “is-complementary-with” \( O_2 \);

The examples of the "is-subsumed-by" and "is-complementary-with" relationships are demonstrated in Figure 9. The nodes are the constraints that specify the characteristics of network events (i.e., \( C_1 \), \( C_2 \), \( C_3 \), and \( C_4 \)). \( C_2 \) is subsumed by \( C_1 \), and \( C_3 \) is subsumed by \( C_2 \). \( C_1 \) and \( C_4 \) don't have overlap so that they are complementary with each other.
To discover an analyst's analytical reasoning process, both temporal and logical relationships need to be considered. More specifically, we are mainly interested in learning how a data triage operation is related to the previous operations. Therefore, given all the operations performed by an analyst, we identify the logical relationships between each operation and all its preceding operations and represent them in a graph structure.

7.3 Knowledge Matching and Challenges

The context of a data triage operation can be defined as all its preceding operations and their temporal and logical relationships [48]. Given the graph structure, the context of a data triage operation is a graph. Therefore, the matching problem becomes a graph matching problem: we need to search in the knowledge base (i.e., a collection of graphs) to find the graphs/subgraphs that matches the current context of the user of the retrieval system.

The time efficiency is the main challenge for graph matching [49]. Graph isomorphism analysis is usually time-consuming. In order to improve the time performance, it worth considering the similarity calculation based on graphs: first, to develop a method for calculating the graph;
second, to develop a similarity measure to compare two graphs; and then match the graphs based on similarity.

To conduct a research on data triage operation, we need to define atomic operation as a trace. A trace includes a sequence of data triage operations performed by analysts. The trace can be illustrated as $T = (O_n)_{n \in \mathbb{N}}$, where $O_i (1 \leq i \leq n)$ is a data triage operation; Where $O_i (1 \leq i \leq n)$ is a data triage operation.

The context of $O_n$ is defined by the sequence of the data triage operations that in previous of $O$. This relationship between data triage operations including temporal and logic relationships.

If $O_1 = (D_1, t_1, C_1)$ and $O_2 = (D_2, t_2, C_2)$ are two data triage operations, the sequential relationship between them is defined by $t_1$ and $t_2$.

There are two kinds of relationships: "happen-before", And "happen-after".

The logic relationship between these two relationships is defined by their characteristic constraints. There are three kinds of logical relationships: "is-equal-to", "is subsumed-by", And "is-complementary-with".

Based on above discussion, the context of the data triage operation $O_i$ is

$$C(O_i) = \langle (O_j)_{j \leq i}, \{R_T(O_j)\}, \{R_L(O_j)\} \rangle, j < i$$

where $(O_j)_{j < i}$ is the group of data triage operations analyzed previous than $O_i$; $R_T$ and $R_L$ is the temporal and logic relationships among $(O_j)_{j < i}$.

The goal is to retrieve specific relevant traces based on the current context of analysts work.

Based on Triage Graph Model and Characteristic Constraint Graph, several algorithms may be implemented for data triage operation retrieval.
7.4 Ontology-based Data Triage Operation Retrieval

Ontology-based retrieval is widely used in semantic web (data) search [50]. Researchers may apply this approach to solving several relevant triage operation retrieval problems (e.g., semantics-aware retrieval of triage operations).

In order to apply this approach, researchers need to map data triage operations into an ontological knowledge base. To achieve this goal, the main hurdle is the ontological annotations. After the ontological annotations are obtained, the next step of data triage operation retrieval seems to `embed" semantic features into the retrieval process.
Chapter 8

Conclusion

A major challenge of data triage in SOCs is the inefficient performance of junior security analysts caused by the lack of experience.

It can be effectively addressed through retrieval of the relevant past data triage operations performed by the senior analysts. We conducted a novel research on data triage knowledge retrieval methods and discussed the new directions in solving the retrieval problem in this field.

Acknowledgment

This work was supported by ARO W911NF-13-1-0421 (MURI) and ARO W911NF-15-1-0576.
Bibliography

[10] Zhang, Y. and Zhai, C. A Sequential Decision Formulation of the Interface Card Model for Interactive IR. ACM.


