The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

CONTEXT-DRIVEN SIMILARITY-BASED RETRIEVAL OF CYBER ANALYST EXPERIENCES FOR MULTI-STEP ATTACK ANALYSIS

A Thesis in

Computer Science and Engineering

by

Deepak Samuel Kirubakaran

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2013

The thesis of Deepak Samuel Kirubakaran was reviewed and approved* by the following:

John Yen
Professor, College of Information Science and Technology
Thesis Advisor

Sencun Zhu
Associate Professor, Department of Computer Science and Engineering

Peng Liu
Professor, College of Information Science and Technology

Lee Coraor
Associate Professor, Department of Computer Science and Engineering
Head of the Graduate Program

*Signatures are on file in the Graduate School

ABSTRACT

Multi-step cyber-attacks on enterprise networks are gaining popularity due to their subtlety and longevity that make them hard to detect. Cyber analysis of these attacks has always been challenging due to the noise-abundant monitoring data and increasing complexity of the reasoning tasks requiring high analytical reasoning capability. Senior/expert analysts often leverage their past experiences from prior analyses during the human reasoning process of current multi-step attack analysis.

Existing techniques including alert correlation, attack graphs do not take into account the human reasoning process during analysis as they fail to capture and share such analysts' experiences while aiding them during analysis. Our experience-based reasoning support system automatically captures experts' experiences where experience is modeled as an interactive process involving actions, observations and hypotheses.

Retrieving useful experiences from the experience base is critical to the usefulness of our system in helping novice analyst during analysis. In this paper, we propose an experience-retrieval approach for our experience-based reasoning support system that helps guide novice analysts in a step-by-step manner by retrieving "relevant" experiences from the experience base using the context of current analysis. We evaluate the scalability and performance of our retrieval approach based on precision and recall of the results with respect to ground truth for varying contexts and experiences in the experience base.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my adviser, Prof. John Yen and my co-adviser, Prof. Peng Liu for their continuous support and guidance they have shown me during my time here at Penn State. I am deeply grateful for them for giving me the opportunity to pursue research in the field of Human-centric Cyber Situational Awareness.

I am also very grateful to my colleagues Chen Zhong and Sooyoung Oh, for working with me on this research project and helping me understand my research area better. I would also like to thank Chen Zhong for developing the framework for the tool with interactive user interface and experience modeling.

I would also like to express my deepest gratitude to my parents and my sister for their constant support, encouragement and love throughout my life, without which I would not have had a chance to be at Penn State.

I would like to use this opportunity to acknowledge all my friends for their support and helping me stay focused on my goals through these years.

Chapter 1

Introduction

With increasing popularity of the Internet, multi-step cyber-attacks are getting common in enterprise networks involving compromise of multiple hosts in the network. Detection and analysis of this type of attacks have become a major challenge for cyber security community due to their complexity and longevity. High false positive rate of Intrusion Detection Systems (IDSs), need for analysts to leverage their expertise during analysis and overwhelming amount of information to be analyzed from multiple sources including system logs, antivirus reports, etc. make analysis of these attacks very challenging for novice analysts.

Analysts need to monitor data generated by IDS, anti-virus and other network monitoring tools in order to detect malicious activities in the network. Often cyber analysts leverage their past experiences during the human reasoning process of current attack analysis. Current approaches including alert correlation, attack graphs, etc., are more machine-centric and are not helpful for multi-step attack analysis due to their inability to take into account the human reasoning process involved during analysis. Since the nature of the problem of analysis is not well defined and due to the lack of a unique pre-defined solution to the problem, it is not feasible to adopt AI techniques including case based reasoning approach. Since human reasoning process is subjective, different analysts could analyze the same situation in different ways before arriving at a conclusion. The incremental analytical reasoning process involved is as important as the result of analysis and hence case-based reasoning becomes less effective owing to the semi-structured nature of our problem of cyber analysis due to constantly evolving new threats and zero day exploits on enterprise networks.

Successful multi-step attack analysis using these techniques depends on the expertise of security analysts in analyzing overwhelming amount of data in order to identify the false positive alerts from the true positives. Existing techniques fail at detecting attacks with unknown signatures (Zero day attacks) that are not detected by signature-based Intrusion Detection Systems (IDSs) like Snort. We aim at addressing these issues by leveraging the human reasoning process during the attack analysis process.

In order to leverage the captured experiences of expert/senior analyst to facilitate cyber analysis by novice analyst, our approach contributes, ability to guide novice analysts in a step-by-step manner during the reasoning process using efficient experience retrieval. Our main motivation behind our approach was to (1) leverage expert analyst's human reasoning process during analysis and to (2) train novice analyst from valuable captured experiences of experts from the past thereby exploiting knowledge gained from the analytical reasoning process of experts.

One of the key challenges our experience-based reasoning support system described by Zhong et al [18] is to identify the useful experiences from the repository of experiences from the experience base. Useful experiences help novice analyst to analyze current situation by guiding them through the analysis in a step-by-step manner. The two key metrics related to our retrieval approach are (1) Relevance of the results with respect to the actual attack scenario (or ground truth) of analysis and (2) Scalability of the approach with increasing number of captured experiences in the experience base. Our retrieval framework has been designed by taking these metrics into account.

Our experience retrieval based on similarity measure, retrieves the most relevant experience from the past, based on the current context of analysis. We evaluate the performance of our approach for experience retrieval based on precision and recall of the results with respect to ground truth for varying contexts and experiences in the experience base in the following sections.

Multi-Step Attack Analysis

Multi-Step attack analysis involves identifying malicious activity in the network by monitoring various alert and information sources including IDS alerts, network traffic, etc. This task usually involves identifying the true positive alerts from IDS using indicators from various other sources including server logs, antivirus reports, etc. Expert analysts tend to more successful during analysis of these attacks due to their past experiences and ability to leverage them for the current analysis unlike novice analysts.

In detecting multi-step attack chains, the trajectory of security analysts' reasoning is in most cases wiggly rather than straight, but this wiggly aspect, though a critical viewpoint to gain insights on "why an analyst make mistakes in doing the job "and "how to improve the job performance of analysts", is not reflected in attack signatures or heuristics used by software tools that help detection. The experiences of analysts including positive and negative experiences provide insights on the analysts’ observations and corresponding hypotheses generated from those observations. These experiences reflect the mental model of different analysts on how they performed their analysis. Due to the wiggly nature of human analytic reasoning, the current context of analysis involves a series of observations that analysts looked at, leading them to generate a new hypothesis.

Challenges in Multi-Step Attack Analysis

Challenges of novice analysts performing multi-step attack analysis are depicted in Figure 1-2.

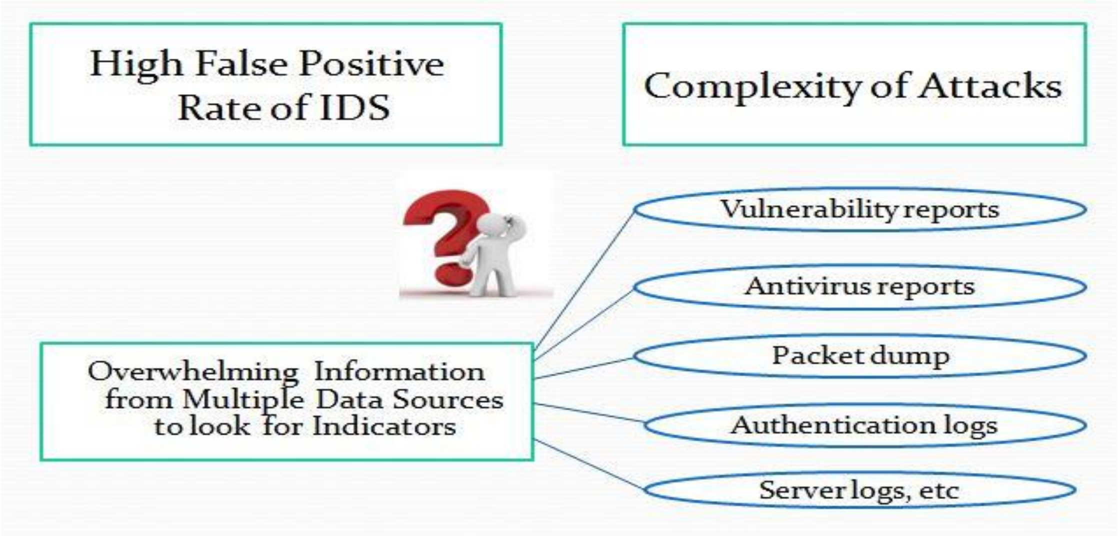

High False Positive Rate of IDS
Complexity of Attacks
Vulnerability reports
Antivirus reports
Packet dump
Authentication logs
Server logs, etc
Overwhelming Information from Multiple Data Sources to look for Indicators

Figure 1-1. Challenges in Multi-Step Attack Analysis.

In an attempt to address and overcome these challenges, we propose the experience-retrieval approach for our experience-based reasoning support system [18]. The related work from the literature and detailed description of our proposed approach and its evaluation are discussed in the following sections.

Chapter 2

Related Work

The related work from the literature addressing multi-step attack analysis includes the AI techniques like case-based reasoning, vulnerability and exploit detection approaches including IDS, attack graphs, etc. and theory of sense making. They are discussed in detail in this section.

Experience in Analytical Reasoning

Due to the overwhelming amount of information available, visual analytical tools that facilitate sense-making has been used. Based on cognitive theory, human beings have limited memory to process data, thereby making the task harder in the absence of visualization tools. Sense making involves information seeking, analyzing observations, deriving insights and retrieving results from insights. Srinivasan et al. [8] highlighted the importance of knowledge built from analysts' expertise and experience using Pirolli and Card's sense-making framework [7]. The theory of sense-making provides the mental model that results from reasoning of analyzing information obtained from the information seeking phase. But, experience has not been defined in the literature clearly.

Relaxation of logic patterns have been proved to be useful for experience matching in cyber situational awareness and using causal relationship between events in the past experience have been useful to detect missed or delayed alerts in current analysis [5, 6]

However existing approaches fail to capture these experiences from analysis of multi-step attacks. In the remainder of this section, we discuss the existing approaches that help in the analysis of multi-step attacks and their shortcomings.

Alert Correlation

Alert correlation techniques aim at aggregating several elementary alerts based on their similarity (e.g. IP address, vulnerabilities, patterns, etc.). They aim at providing synthesized information by correlating aggregated alerts in order to help the administrator and senior analyst with useful information. Alerts are modeled as facts and the attacks are modeled with specifications that include pre-condition, post-condition, scenarios, etc. The alert correlator leverages these specifications to generate attack plans and predict the next steps in the attack.

It includes forming attack scenarios from alerts using pre-conditions and consequences from previously known attacks [1] or by using data mining techniques (LAMBDA) [9] or logic-based techniques [2, 3]. These approaches were limited to identifying known attack scenarios. A variation of this uses a consequence mechanism to specify what type of attack follows a specific attack [10]. Some approaches used partial matches in pre-requisites (pre-condition) and consequences (post-condition) by representing relationship between predicates using ontology rules [11]. But these approaches are not human centric and are not efficient against new attack scenarios that are similar to attacks of past. Also, their inability to capture and leverage experiences from each analysis of the past makes them less scalable and less robust.

Alert correlation techniques have also been used to detect all possible attacks that could happen against a vulnerable network system [12]. But we are more interested in *what has happened to a system* than *what may happen.* Hyper alert correlation graphs can be generated as a result of correlation that defines the set of pre-requisites and consequences at various stages in the attack [13, 14]. Other approaches include query optimization techniques for efficient alert correlation [15].

Attack correlation techniques are more suitable for real time network monitoring by expert analysts. But they do not have the capability to detect zero-day attacks that are not detected by Intrusion Detection Systems (IDSs).

Attack Graph

Attack Graphs help in detecting vulnerabilities in the network based on which the analyst can make defenses for vulnerable hosts. It provides a common representation to abstract multiple attack patterns. Graph generation can be time consuming and logical formalisms can be used to represent them as they are usually large and complex.

Reasoning engines used modeling languages to represent network configuration and vulnerabilities as facts and applied rules to detect attack path [16]. Scalable attack graphs could be generated from these reasoning engines [4]. But attack graphs are difficult to use and understand by humans. Due to large and complex nature of attack graphs, the information they present is not easily comprehended by novice analysts making them less effective for multi-step attack analysis.

Case Based Reasoning

Case Based Reasoning (CBR) leverages the past cases to solve a current case where the cases are independent of each other [17]. It provides a way for learning via reasoning in contrast to rule based systems. They are easy to maintain and are used in detecting vulnerability associations and Intrusion Detection Systems (IDSs) in networks. In cognitive science, they can be used to model human experiences, but they lack formal representation of human experience and efficient approaches for retrieval.

Novice analysts can learn from the past experiences of experts based on the theory of sense making [7]. Visualization tools have been used to capture attack analysis based on the theory of sense making [8]. We propose an experience-based human centric approach that captures the human reasoning process of experts during attack analysis and leverages them in order to help novice analyst during analysis.

But case-based reasoning is effective for well-defined problems like medical diagnosis where for a given set of systems there could only be one probable result of diagnosis. Moreover the problem is generally static and does not grow incrementally or dynamically. Hence they are not suitable for multi-step attack analysis which is less well defined and semi-structured given the increasing number of threats and exploits in networks. Since previous solutions cannot be applied directly for the current problem, case-based reasoning does not provide an effective solution against multi-step attack analysis.

Information Retrieval

Information retrieval has been studied extensively based on ranking using keyword-based matching, most of them being text-based. But retrieval based of captured experiences based on context has not been proposed with regard to cyber-attack analysis. This is due to the lack of existing tools that leverage captured experiences involving human reasoning process. Our approach uses fuzzy matching to retrieve relevant experiences from experience base unlike existing techniques. Precision and Recall have been used extensively as the evaluation metric for retrieval systems.

Chapter 3

Experience-based Human Centric Tool

In this section, we provide an overview of our tool that provides an experience-aided reasoning support system described by Zhong et al [18], in addition to experience representation, relevance definition, and criteria for matching, observation types, relevance measure, and its properties.

Tasks performed by analysts during his analytical reasoning process while analyzing multi-step attacks can be modeled as an action-observation-hypothesis cycle shown in the figure 3-1.

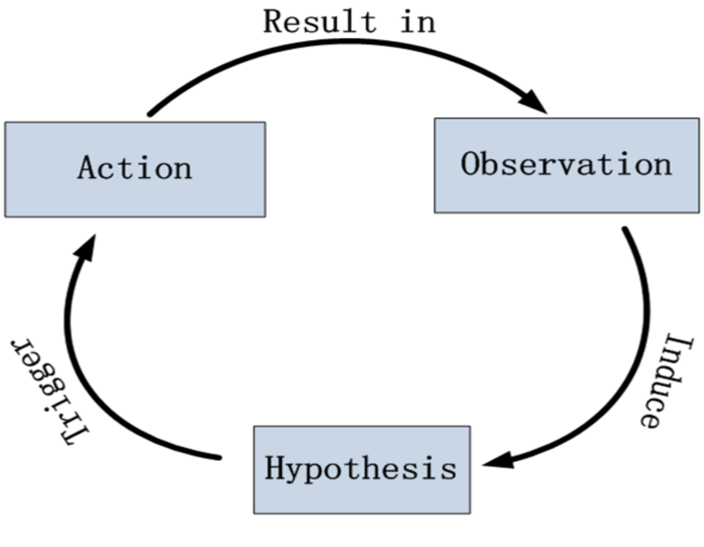


Figure 3-1. Action-Observation-Hypothesis Cycle.

E-tree and H-tree

Experience Tree (ET) is used to represent a single instance of experience. Hypothesis Tree (HT) represents flow of thoughts of the analysts during analysis. Experience tree (E-tree) consists of a collection of Experience Units (EUs) as nodes and corresponding hypotheses resulting from them as links. Each Experience Unit (EU) consists of a set of actions and a set of observations resulting from those actions. One or more hypothesis results from each EU, each of which could in turn trigger new EUs. The resulting structure is an n-ary experience tree.

Experience is represented by the N-ary tree as shown in Figure 3-2 [18], where each node is represented by an experience unit and hypothesis representing edges. Each experience unit consists of a set of actions and a set of observations resulting from the actions.

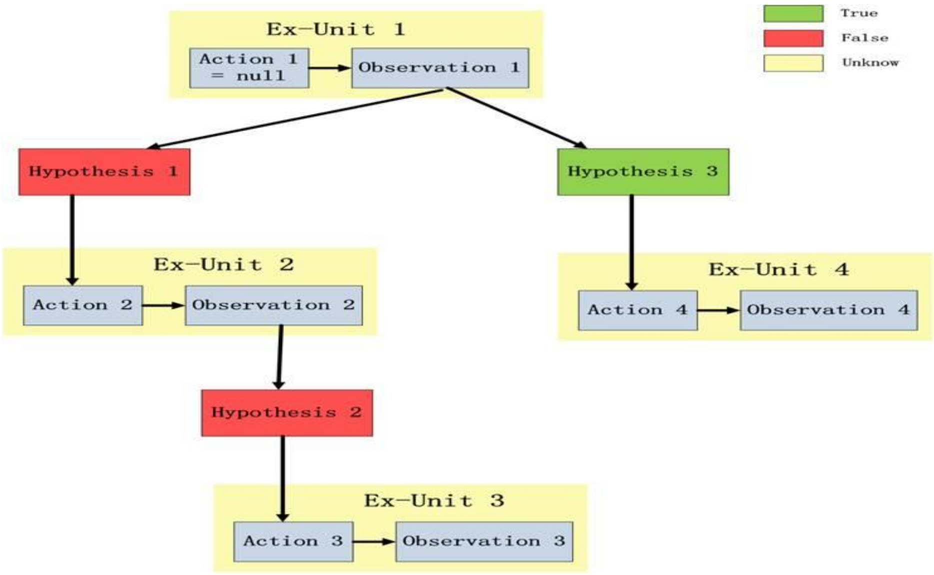


Figure 3-2. Experience Tree (ET).

The initial trigger is the result of a set of observations that represents root of the tree. The subsequent nodes are triggered as a result of hypothesis from prior observations. The ancestral experience units includes all the observations that form the current context with respect to a given child experience unit. Each experience unit could result in a positive or negative hypothesis as shown in the figure.

An example of ET for analysis involving DNS cache poisoning attack on DNS server by an attacker is shown in Figure 3-3.

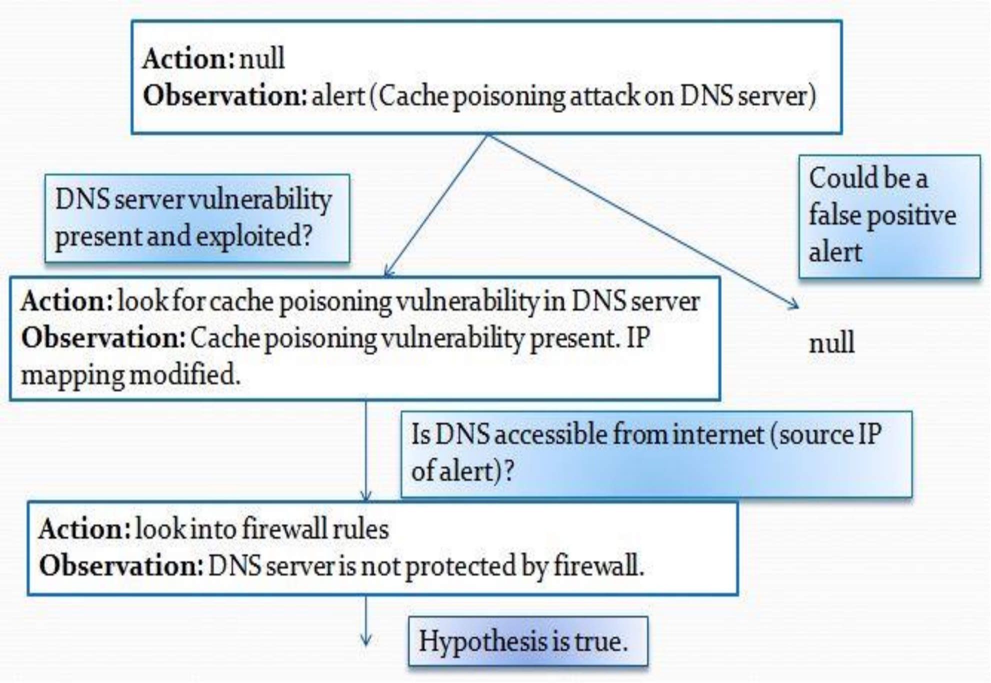

Action: null
Observation: alert (Cache poisoning attack on DNS server)
DNS server vulnerability present and exploited?
Could be a false positive alert
Action: look for cache poisoning vulnerability in DNS server
Observation: Cache poisoning vulnerability present. IP mapping modified.
null
Is DNS accessible from internet (source IP of alert)?
Action: look into firewall rules
Observation: DNS server is not protected by firewall.
Hypothesis is true.

Figure 3-3. Example of ET.

Hypothesis tree (H-tree) is used to capture the current flow of thoughts of the analyst during analysis. For example, the analyst might observe a set of alerts and generate any number of hypotheses based on those observations.

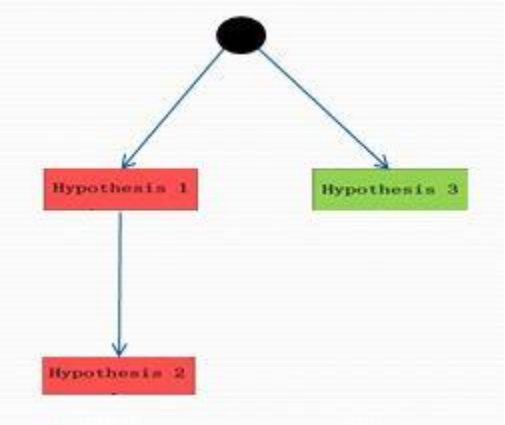


Figure 3-4. Hypothesis Tree (ET).

Each of these hypotheses could trigger more observations. The context with respect to current hypothesis is obtained from the H-tree and can be defined as the set of all observations along the path to the root that led to current hypothesis.

Figure 3-4 shows H-tree corresponding to the experience tree shown in Figure 3-2 [18]. Hypothesis 1 and hypothesis 2 represent negative hypotheses and hypothesis 3 represents positive hypothesis. Both hypothesis 1 and hypothesis 3 are triggered by initial set of observations. An example of HT for analysis involving DNS cache poisoning attack on DNS server by an attacker is shown in Figure 3-5.

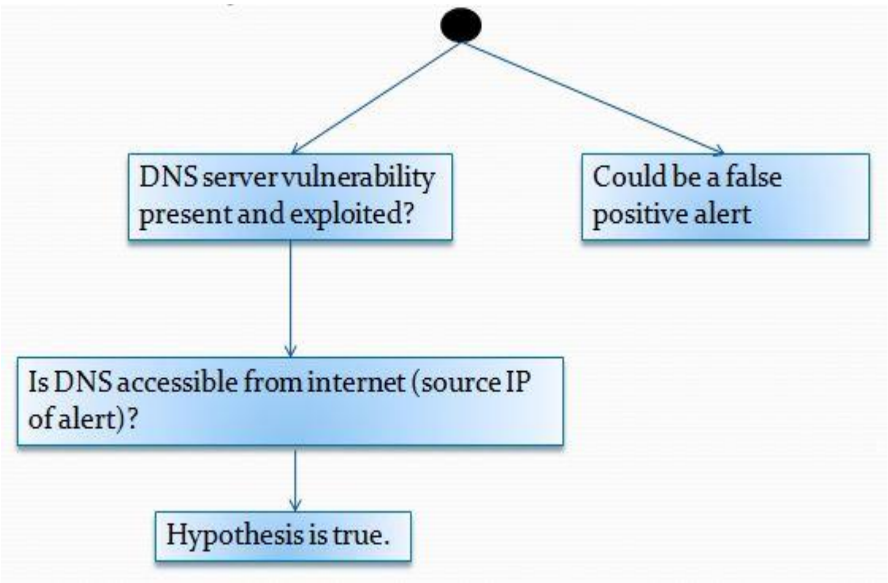

DNS server vulnerability present and exploited?
Could be a false positive alert
Is DNS accessible from internet (source IP of alert)?
Hypothesis is true.

Figure 3-5. Example of HT.

The problem involved in experience retrieval is to detect the most relevant experience from the experience base using the current context. In the following sections we define "relevance" and the relevance measure used for ranking results in our approach and its properties.

Defining "Relevance"

Relevance is used with respect to a context. During multi-step attack analysis, context with respect to a hypothesis can be defined as the set of all the observations that triggered the hypothesis. In our E-tree representation of experience, this can be defined as the set of all observations from the current experience unit through the root experience unit of the experience tree.

Conceptually, relevance of experience tree (ET) with respect to current context is defined by, Relevance (ET, Observations (context)) which measures the degree of relevance between an

E-tree (ET) and observations in a given context. An ET includes multiple paths; each can be compared with the observations in the given context.

Relevance (ET, Observations (context)) is a fuzzy disjunctive operator applied to Relevance(Observations (p), Observations (context)) for all paths p in ET. Relevance(Observations (p), Observations (context)) is a weighted sum of the (fuzzy) match between observations O_i in Observations (context) and observations O_j in p. The relevance measure/score reflects the degree of relevance between an E-tree and a given context.

Criteria for Matching

The following criteria need to be satisfied for an observation in the current context to match an observation from an experience unit from ET in experience base.

For O_i from current context to match with O_j from an EU from ET,

- O_i and O_j should be of the same observation type (IDS alert, packet dump entry, etc)
- Depending on the type, there should be a minimal match in fields between O_i and O_j.

Observation Types

The minimal matching fields associated with individual data source are shown in the table 3-1 along with other fields for each data source. Each field is associated with a weight and each data source is associated with a base weight obtained as a result of minimal match. The details on specific weights are discussed in the evaluation section of the paper. The value of the fuzzy match between an observation from the context and observation from EU of an ET falls in the range [0, 1]. These fields are considered important for each data source and they must match in order for the EU to match the current context. These minimal matching fields are used as composite keys for hash tables in order to achieve fast retrieval of matched EUs from the experience base.

Table 3-1. Minimal-matching Fields for Individual Data Source.

Data Source or Observation Type	**Minimal matching fields and other fields**
Snort IDS	**Snort ID (Attack Signature)** Other: source IP, destination IP, source host type, destination host type.
Vulnerability List	**Vulnerability ID** Other: IP address, Host type
Packet Dump	**Protocol, source host type, destination host type.** Other: source IP, destination IP
Port Information	**Port number, status, host type** Other: IP address
Web server log	**Host type** Other: IP address

Authentication log	**Host type** Other: IP address
Anti-virus log	**Virus name, host type** Other: IP address
Accessibility	**Is accessible?, source host type, destination host type** Other: source IP, destination IP

Relevance Measure

The degree of match between an observation O from current context and observation O_{EU} from experience unit is given by the following,

- $\mathbf{Match\ (O, O_{EU}) = Base\text{-}match_{O} + \sum w_{i} * match\ (Field_{i}^{O}\ , Field_{i}^{EU})}$,

  Base-match is match degree that result from matching of minimal-matching fields for corresponding observation type.

- Each Observation in the current context is matched with each of the set of observations from matched EU to determine the overall match of current context with respect to matched EU.

Property of Relevance Measure

Relevance of an E-tree with a context C increases monotonically as the context extends with additional observations. Since the multi-step attack analysis provides an incremental solution, we use relevance measure with similar properties.

Theorem: Relevance (ET, O(C1)) <= Relevance (ET, O(C2)) if O(C1) is a subset of O(C2)

Chapter 4

Approach for Experience Retrieval

Experience retrieval involves ranking the E-Trees in the experience base using the current context and retrieving the top-K ranked paths from these E-Trees. This involves 3 steps including match propagation, retrieval and update propagation, discussed in this section.

Design Objectives

The main design objective in the experience retrieval approach is to achieve fast experience retrieval. This includes,

- Time-efficient Retrieval of Matched EUs.
    - Using Hash Tables (O(1) retrieval). Every observation type is associated with a hash table for fast retrieval of matched EUs.
- Time-efficient Ranking of E-Trees
    - Efficient Match Propagation algorithm of O(number of matches * average length of matched path)
- Time-efficient Retrieval of Ranked E-Trees
    - Efficient Retrieval and Update Propagation algorithm of O (average length of matched path)

Architecture

The architecture of our experience retrieval approach is shown in Figure 4-1. It consists of uses the current context observations captured from analyst actions and experience base containing past experiences in the form of E-Trees.

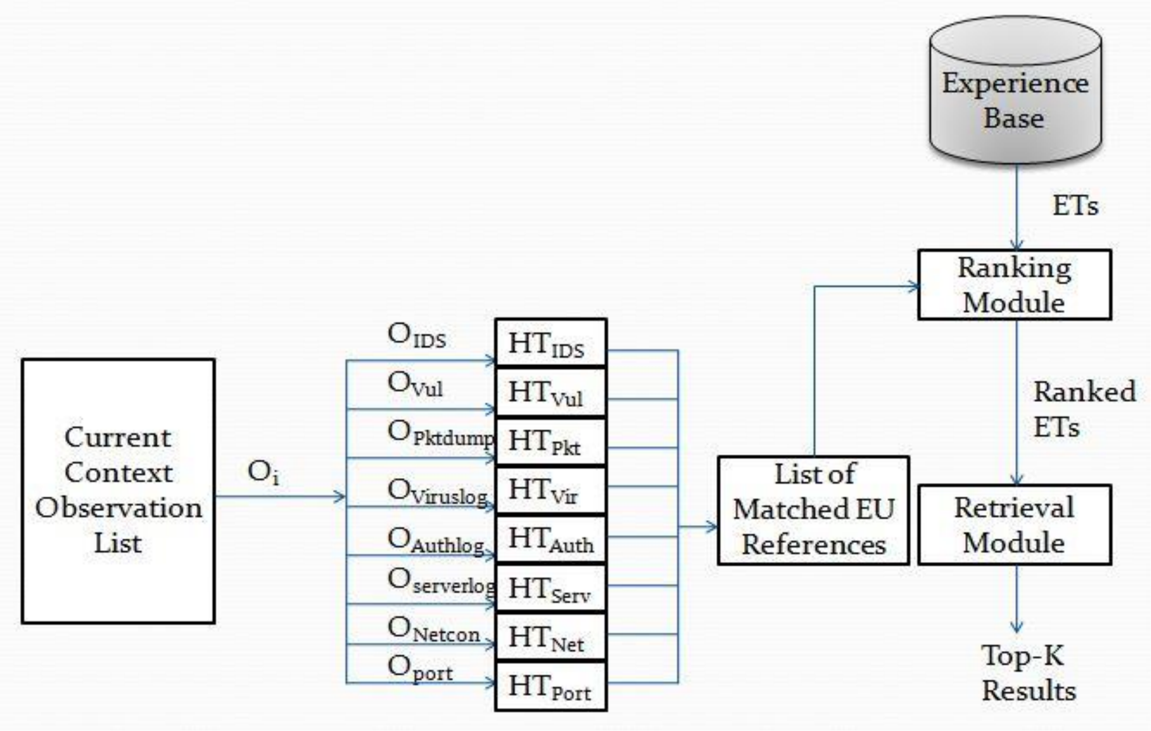


Figure 4-1. Architecture of Experience Retrieval Approach.

Hash Tables

Each observation type is associated with a hash table that hashes on the minimal matched fields associated. The minimal matching fields are used as a composite key for hash table. The hash tables map these keys to list of matching Experience Units (EUs) from the E-Trees (ET).

The ranking module uses the list of references and the E-Trees (ETs) and returns the list of ranked E-Trees (ETs).

Ranked and Unranked E-Trees

The match propagation algorithm is used by the ranking module that ranks the E-Trees (ETs) from the experience base. Figure 4-2 shows the representation of ranked E-Tree ET1.

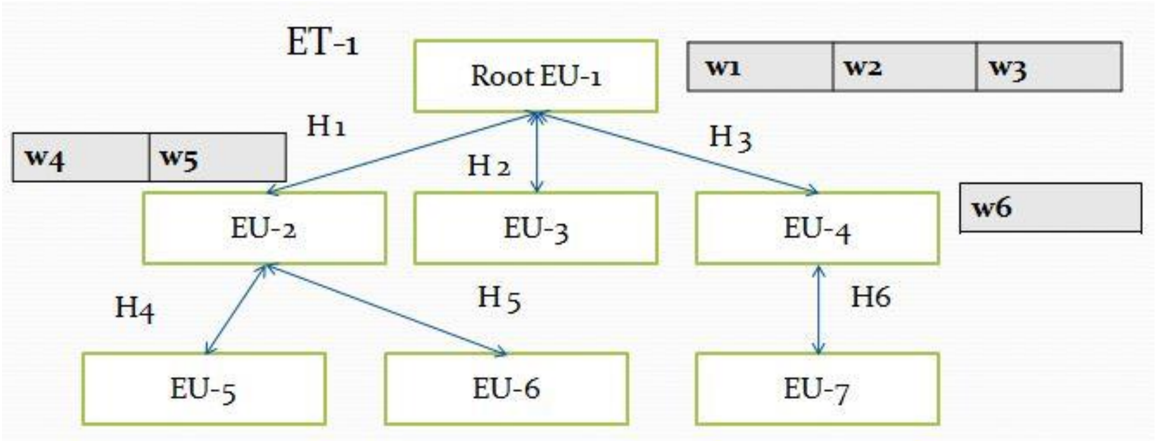


Figure 4-2. Ranked E-Tree ET1.

Each E-Tree has a unique E-Tree ID.

Each EU has a unique EUID and a matching degree (initially 0).

Each parent EU has a list of Child EU references named Child-EU list. For example, Child-EU list of Root EU-1 is {EU-2, EU-3, EU-4}

Each EU has a reference to its parent EU.

Each parent EU also contains a list containing the matching degree of its sub-trees (initially 0). For example, matching degree of sub-tree list of EU-1 is {w1, w2, w3}, where w1, w2 and w3 represent the matching degree of the sub-trees of EU-1 respectively.

Figure 4-3 shows the structure of E-Tree before ranking with each EU containing matching degree sub-tree list initialized to 0.

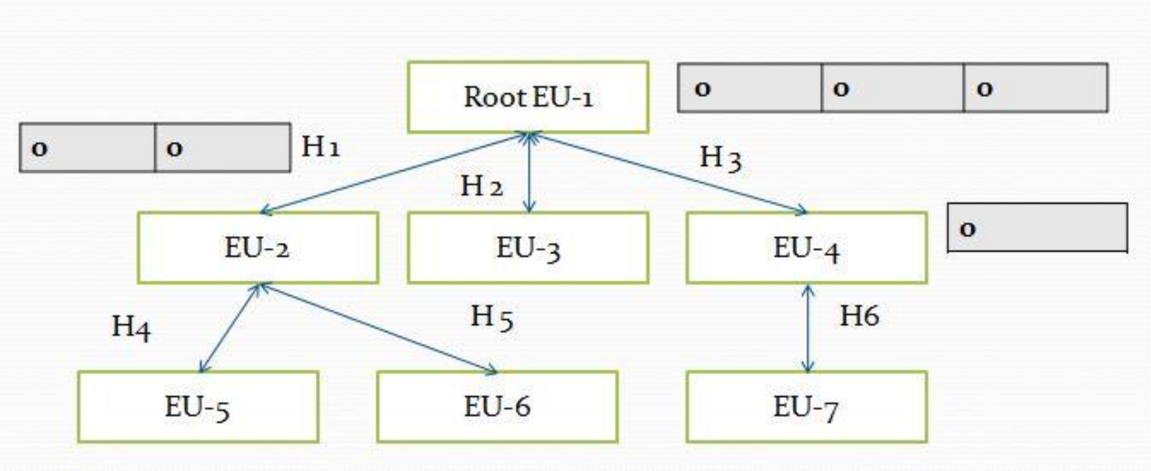


Figure 4-3. Unranked E-Tree ET1.

Match Propagation Algorithm

If an observation in the current context matches EU5 of ET1 with a matching degree of 0.6, the view of ET1 after matching is shown in Figure 4-4.

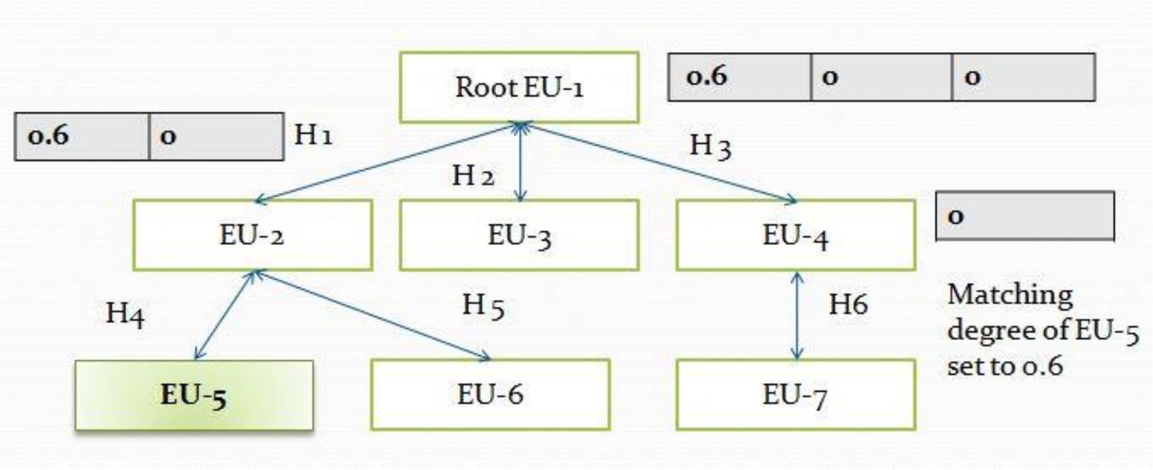


Figure 4-4. E-tree ET1 after EU5 Match.

If EU-6 matches with an observation in current context with a matching degree = 0.5, the view of ET1 is shown in Figure 4-5. It can be seen that the match is propagated till the root EU (EU-1) and the matching degree of sub-tree list is updated along the path to the root.

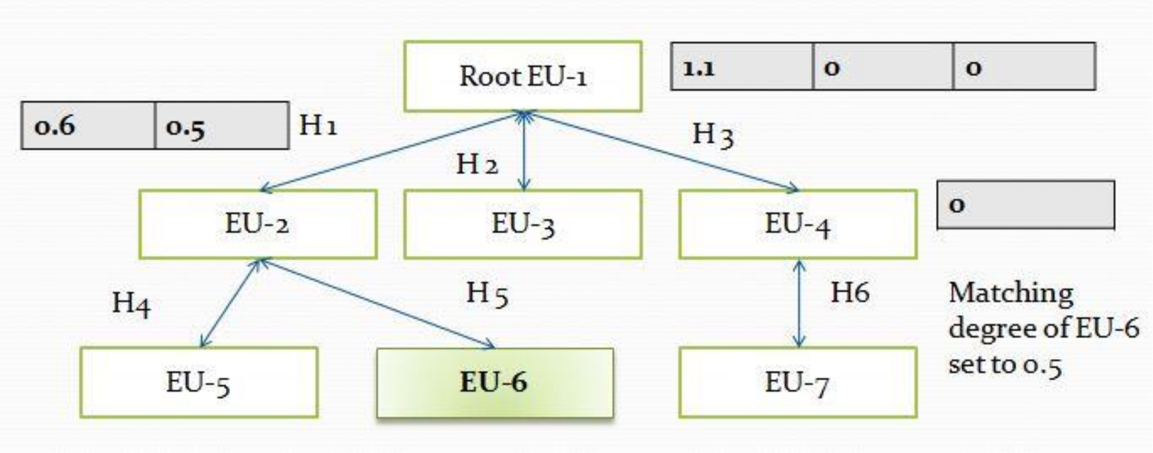


Figure 4-5. E-tree ET1 after EU6 Match.

If EU-4 matches an observation in current context with matching degree = 0.2, the view of ET1 is shown in Figure 4-6.

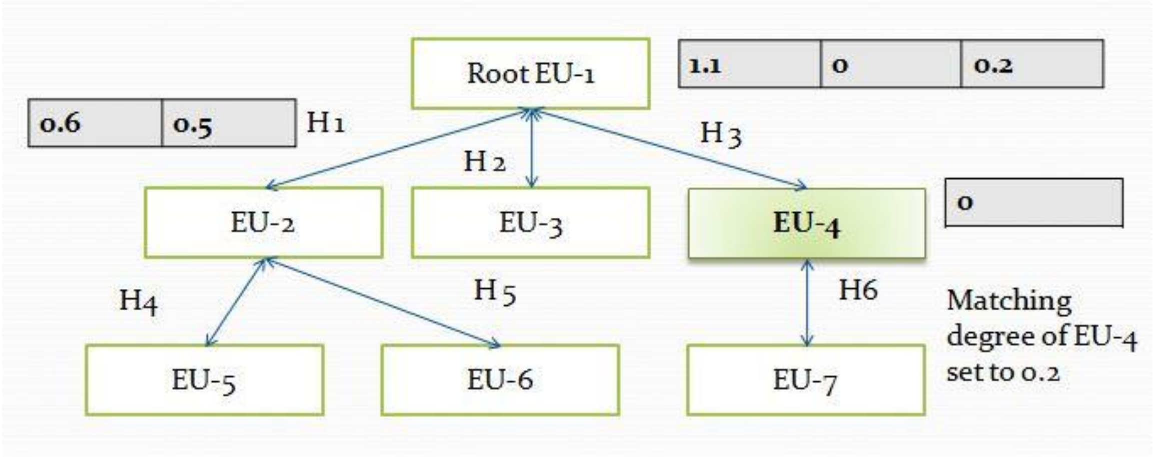


Figure 4-6. E-tree ET1 after EU4 Match.

Currently, the maximum sub-tree match of ET1 is 1.1. This process is carried out for all the matched ETs in the experience base resulting in a list of ranked E-Trees. Upon completing the match propagation for all the observations in the current context, the retrieval and update propagation is triggered by the retrieval module. The time complexity of the match propagation algorithm is O (Number of matched EUs * Average length of the matched path).

Retrieval and Update Propagation Algorithm

In order to retrieve the most relevant E-tree, select the E-Tree with the maximum sub-tree match. Upon selecting the most relevant E-tree, the path is retrieved by traversing along the sub-tree EUs with the maximum degree of match. If ET1 is the most relevant E-tree (with maximum sub-tree match), the retrieved path is shown in Figure 4-7.

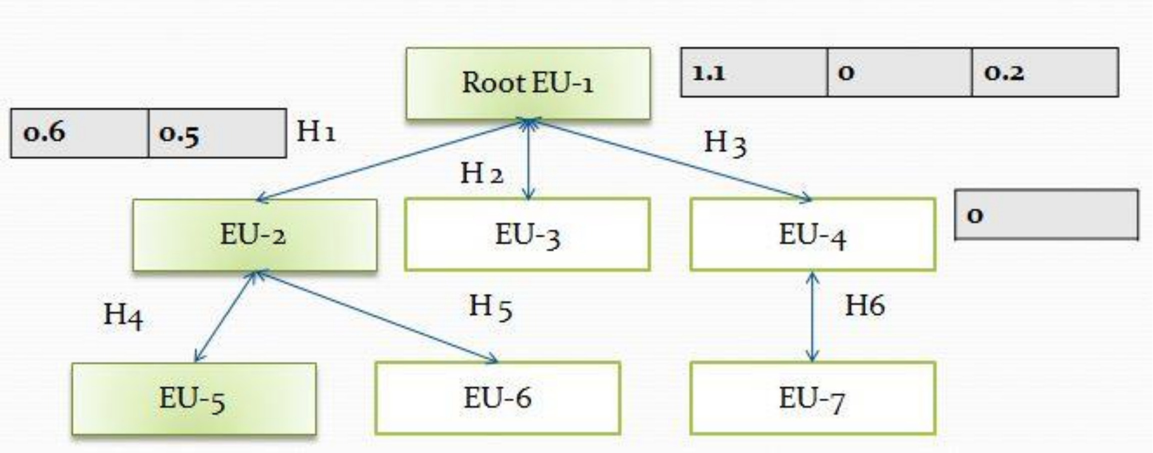


Figure 4-7. Most Relevant Path Retrieval.

Upon retrieving the most relevant path, the E-tree needs to be updated with the new sub-tree matching degrees, before the next relevant path is retrieved. This update propagates till the root of the matched path. Figure 4-8 shows the results of the update propagation upon retrieving the most relevant path EU1 -> EU2 -> EU5.

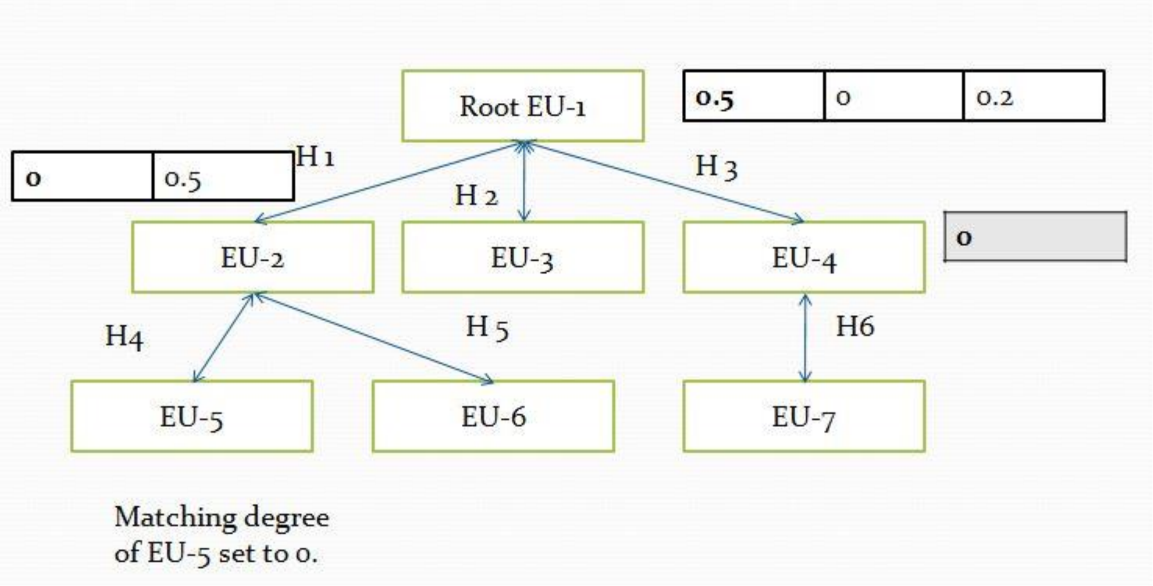


Figure 4-8. Update Propagation.

The update propagation involves setting the matching degree of the leaf node of the retrieved path to 0 and updating the matching degree of sub-tree list of all the nodes along the path to the root EU (EU1). The update propagation algorithm is summarized below,

```
Update-Matching-Degree (parent-EU, child-EU, new-value)
{
    Child-EU Matching degree = new-value
    If (parent EU! = null)
    {
            Find index corresponding to child-EU in the child-EU-list of parent EU
             Update the matching degree sub-tree corresponding index to new value
             Find sum over all matching degree sub-trees =>Max-sub-tree-match
             Call Update-Matching-Degree recursively with the parent EU of parent EU, parent EU,
            Max-sub-tree-match
    }
}
```

The time complexity of the update algorithm is O (Length of the matched path). This operation is performed upon retrieval of each relevant path from ET.

Chapter 5

Evaluation

We evaluate our experience retrieval approach based on the precision and recall of the results of search with respect to the ground truth.

Experiment Design

Consider a network topology shown in Figure 5.3. The ground truth involves two multi-step attack chains targeting the data base server and mail server respectively.

The experience base contains 34 experiences. They are categorized into 3 types as shown in the table below,

Table 5-1. Types of Experiences.

Type of Experience	**Meaning**	**Number of Experiences in Experience Base**
Relevant Experiences	Related to Ground Truth.	4
Partially Relevant Experiences	Have one or more steps in common with Ground Truth.	15
Irrelevant Experiences	Not related to Ground Truth.	15

Attack Scenarios

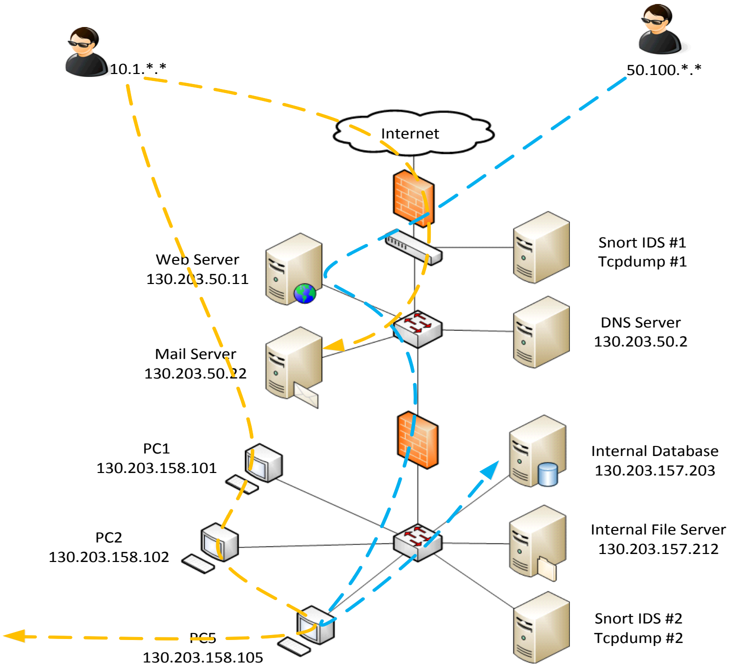


Figure 5-1. Network Topology and Ground Truth.

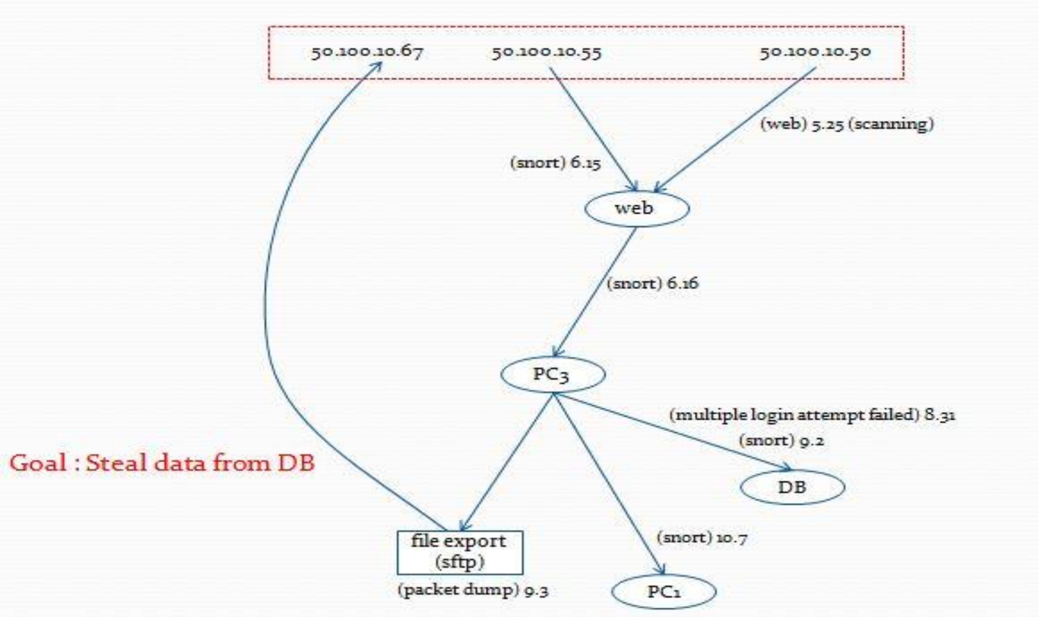


Figure 5-2. Database Records Ex-filtration.

Consider the network shown in Figure 5-1. The scenarios included in ground truth are described below,

Attacker → Web server → PC → Database server

The goal of the attacker is to ex-filtrate packets from the database server. Remote code execution attack takes place at the web server. Malicious page served to PC3 resulting in RAT installation on PC3. DB records are ex-filtrated via PC3.

Attacker → PC

→ Mail server

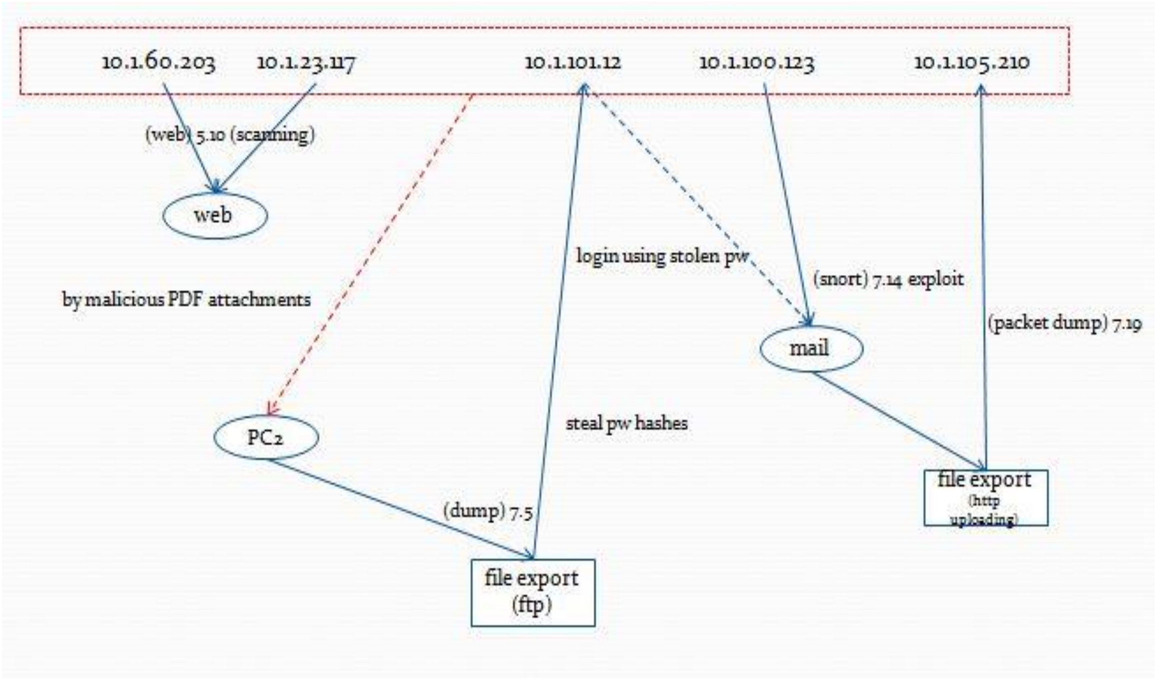


Figure 5-3. E-mail Ex-filtration.

RAT installation on PC2 takes place via spam e-mail, followed by cracking stolen password hashes from PC2 for mail server login. Mail server privilege escalation attack takes place and packets/emails are ex-filtrated.

The scenarios related to partially relevant experiences in the experience base are shown below,

Attacker → PC → File server

PC installed with RAT with spam mail and file server vulnerability exposed resulting in access to shared files and directories.

Attacker →DNS server→ Web server → PC

DNS cache poisoning attack on DNS server resulting in web server serving malicious pages to PC installing RAT.

Attacker → Web server → Database server

Remote code execution attack takes place on web server, followed by exploitation of DB SQL injection vulnerability on database server with a crafted request.

Attacker → Web Server → File Server → PC

Remote code execution attack takes place on web server, followed by privilege escalation on File server leading to ex-filtrating files from PC via shared file server.

The scenarios related to irrelevant experiences in the experience base are shown below,

Attacker → Web Server

Denial of Service attack takes place on the web server.

Attacker →Mail Server

Denial of Service attack takes place on the mail server.

Observations in the current context can be classified as relevant and irrelevant observations with respect to the ground truth. The data sources involved and associated base weights and individual field weights are shown in the following table,

Table 5-2. Weights of Individual Fields

Data Source	**Weights**
Snort	Base match (0.6), Source IP (0.2), Destination IP (0.2), Source host type (0.1), Destination host type (0.1).
Vulnerability List	Base match (0.7), Host type (0.2), IP address (0.3)
Packet Dump	Base match (0.6), Source host type (0.2), Destination host type (0.2).
Port Information	Base match (0.8), IP address (0.2)
Web Server Log	Base match (0.5), IP address (1.0)
Authentication Log	Base match (0.5), IP address (1.0)
Antivirus Log	Base match (0.8), IP address (0.2)
Accessibility	Base match (0.8) Source IP (0.1), Destination IP (0.1)

Systematic Evaluation

Evaluation of the experience retrieval approach should answer the following questions,

- How do irrelevant observations affect the efficiency of retrieved results?
- How do irrelevant experiences in experience base affect the efficiency of retrieved results?
- How do partially relevant experiences affect the efficiency of retrieved results?

From Figure 5.4, it can be seen that the precision increases with increasing number of relevant observations. For relevant observations < 20% it can be seen that, the top-5 results have better precision than top-1 and top-3 results. This is because the relevant results get ranked lower than some irrelevant paths when the ratio of relevant observation is less than 20%. In all other cases, top-1 and top-3 results have higher precisions since the relevant results get higher rank.

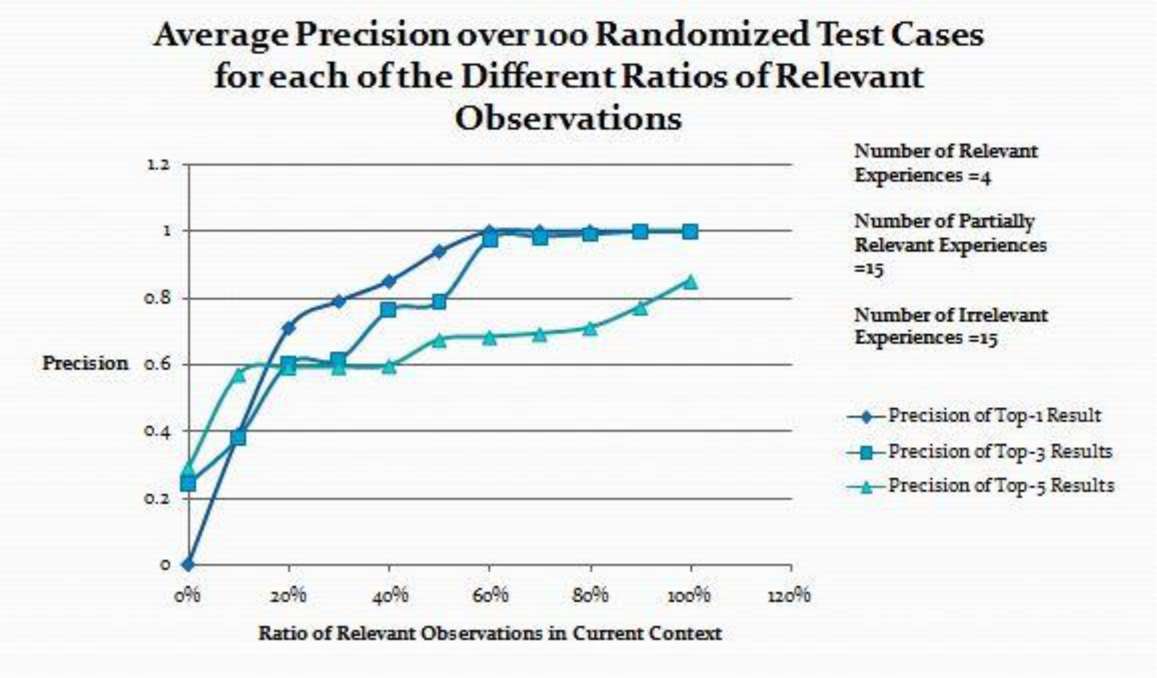


Figure 5-4. Effect of Relevant Observations on Precision

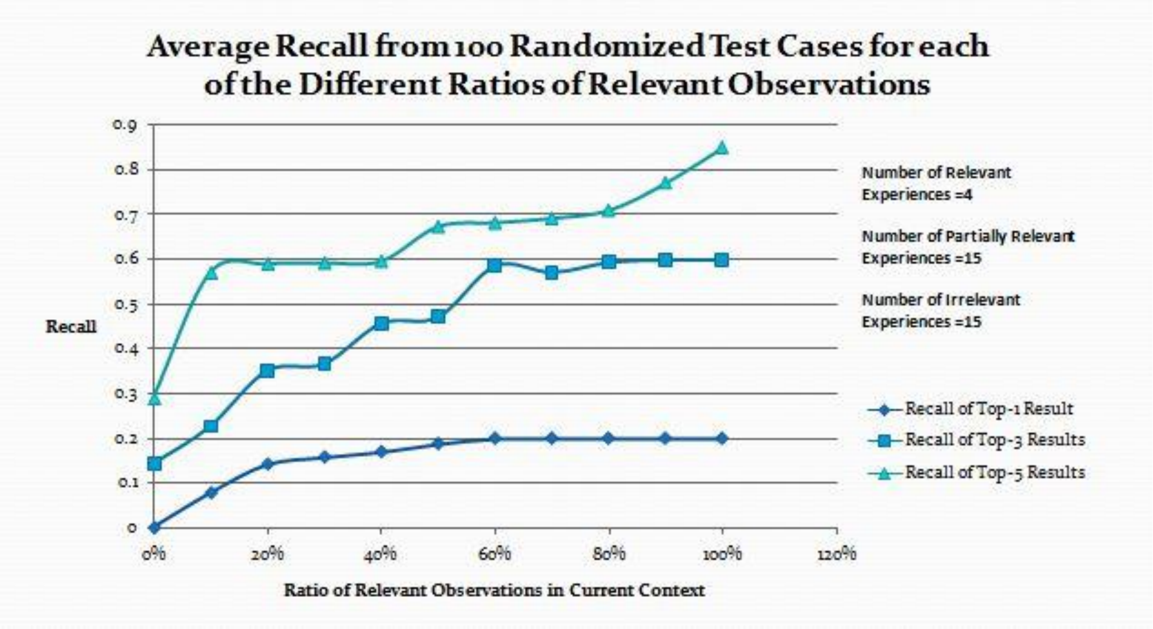


Figure 5-5. Effect of Relevant Observations on Recall

From Figure 5-5, it can be seen that recall improves on increasing ratio of relevant observations.

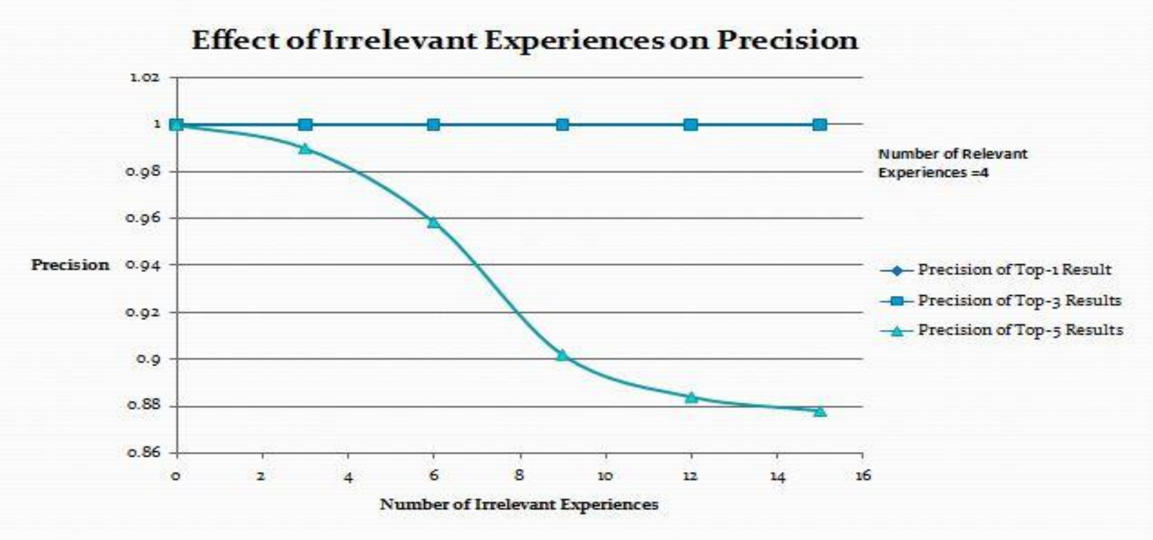


Figure 5-6. Effect of Irrelevant Experiences on Precision

From Figure 5-6 and 5-7, it can be seen that both precision and recall of the top-5 results decreases marginally on increasing number of irrelevant experiences while the top-1 and top-3 results are not affected.

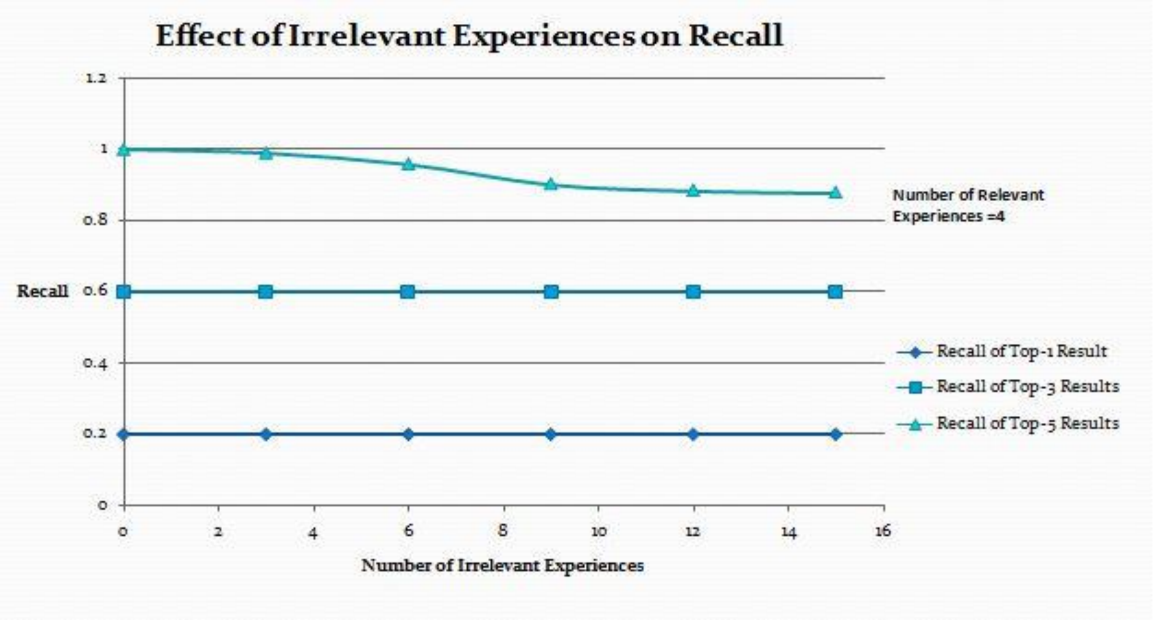


Figure 5-7. Effect of Irrelevant Experiences on Recall

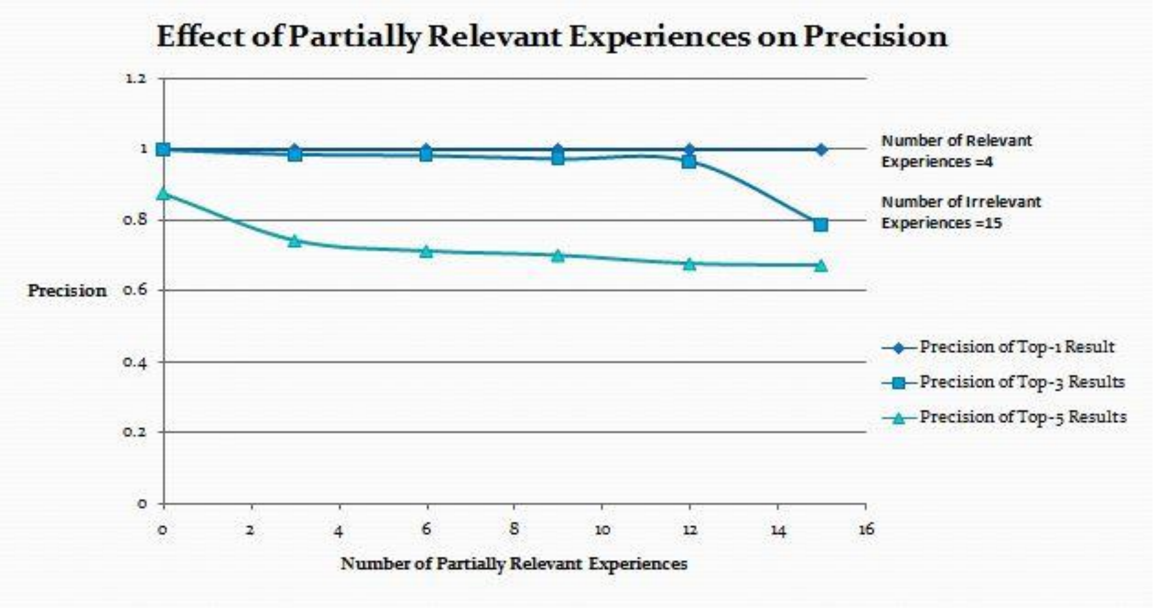


Figure 5-8. Effect of Partially Relevant Experiences on Precision

From Figure 5-8, it is evident that precision of top-3 and top-5 results are more affected by increase in the number of partially relevant experiences. The increase in partially relevant experiences affects precision more adversely compared to effect of increase in irrelevant experiences.

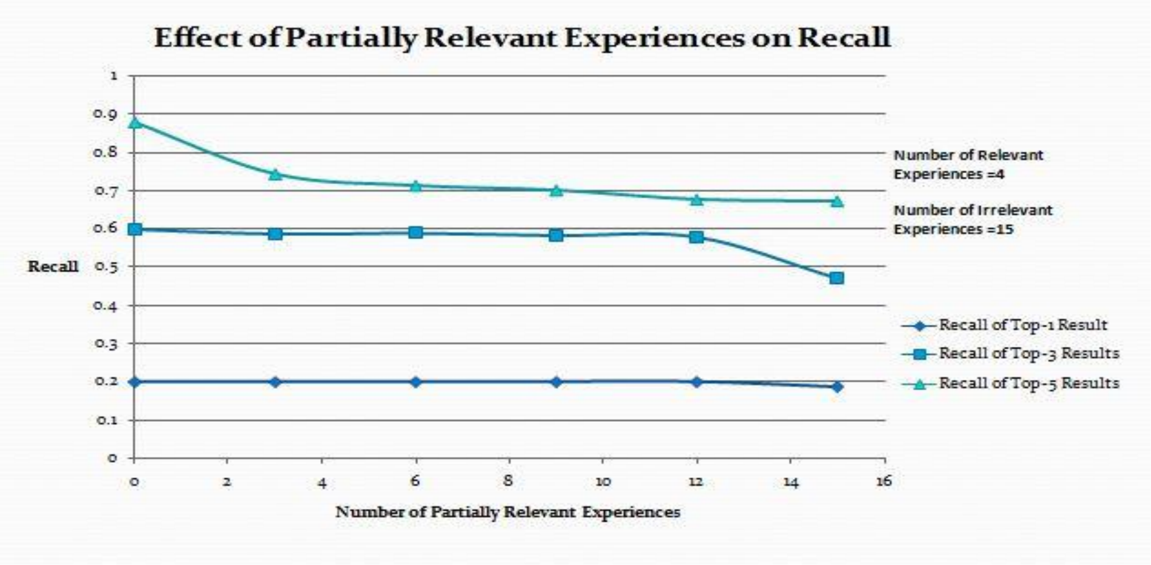


Figure 5-9. Effect of Partially Relevant Experiences on Recall

From Figure 5-9, it is evident that recall of the top-3 and top-5 results is more affected by increase in the number of partially relevant experiences. The increase in partially relevant experiences affects recall more adversely compared to effect of increase in irrelevant experiences

From the evaluation, it can be concluded that with increase in the number of irrelevant and partially relevant experiences in the experience base, the ranking of the results get affected when the current context includes a small ratio of relevant observations. The precision and recall of the system is improved when the ratio of relevant observations is over 20%.

Chapter 6

Conclusion and Future Work

Our approach can be extended to support keyword based searching from the H-nodes in the knowledge base. Also, the experience reasoning process can be captured as a trace since human reasoning may not be always structured as represented in the tree. Also, depending on the nature of hypotheses generated, the tool can be extended to support conjunctive hypotheses. This can help analyze more complex attacks.

The ranking can be improved by including additional constraints in the algorithm including temporal constraints. This can improve ranking in the presence of partially relevant experiences. The approach should be evaluated for real world network monitoring with a large experience base (~1000 experiences).

Our work raises a number of research questions related to context-based searching in AI, human computer interaction, pattern recognition in the human reasoning process in cognitive science, etc.

Appendix A

Snapshots and Implementation Details

Tool is developed in C# language with over 5K Lines of Code (LOC). The experience retrieval framework accounts for nearly 750 Lines of Code (LOC). The experience and hypotheses trees are represented using .xml format. The tool uses mouse clicks and keystroke recording in order to identify analyst's observations.

Working Schema of the System

The working schema of the tool is described in the experience reasoning support system framework [18]. It includes data view that provides data monitoring, navigation view that provides hypothesis navigation and knowledge view that facilitates experience guidance. The high level overview of the interface providing these views are shown in figure A-1 described by Zhong et al [18] on the reasoning support system.

Figure A-2 shows the main interface of the tool showing the Snort (IDS) alerts in the order in which they are generated. The tool provides two modes of operation namely: Using experience and Capturing experience. The former is used for novice analysts and latter is used to capture expert analyst experience. We use the Using experience mode for evaluating the search and retrieval algorithm.

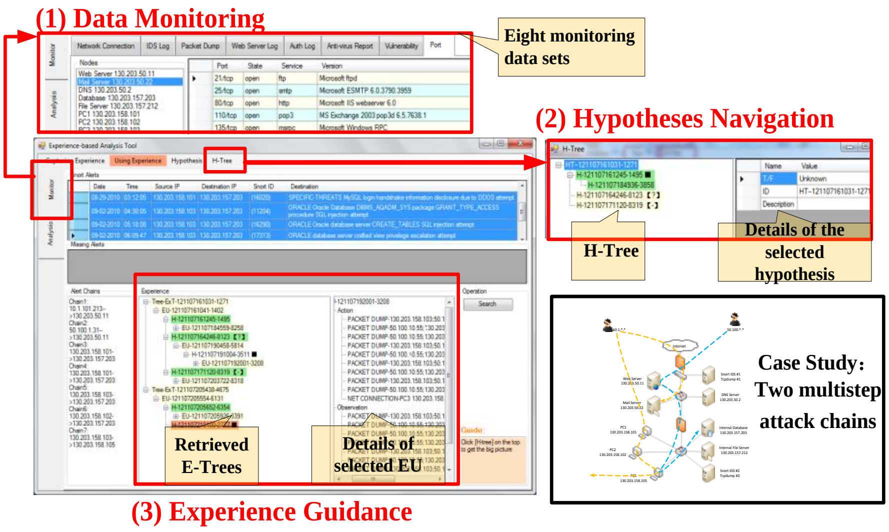
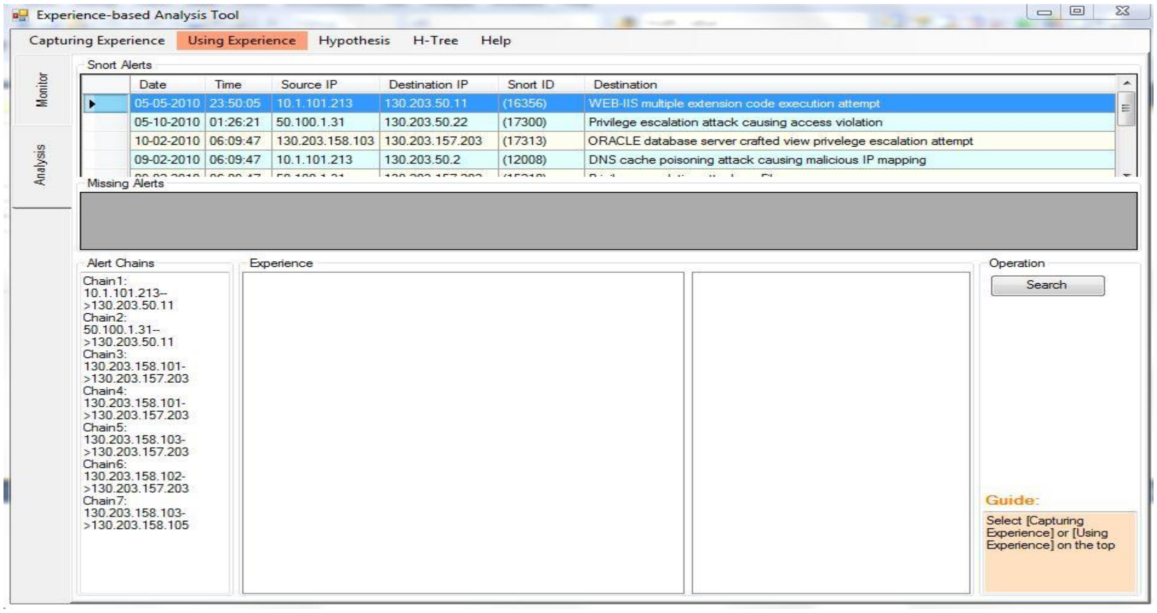

(1) Data Monitoring
Eight monitoring data sets
(2) Hypotheses Navigation
H-Tree
Details of the selected hypothesis
Case Study: Two multistep attack chains
Retrieved E-Trees
Details of selected EU
(3) Experience Guidance

Figure A-1. Tool Outlook



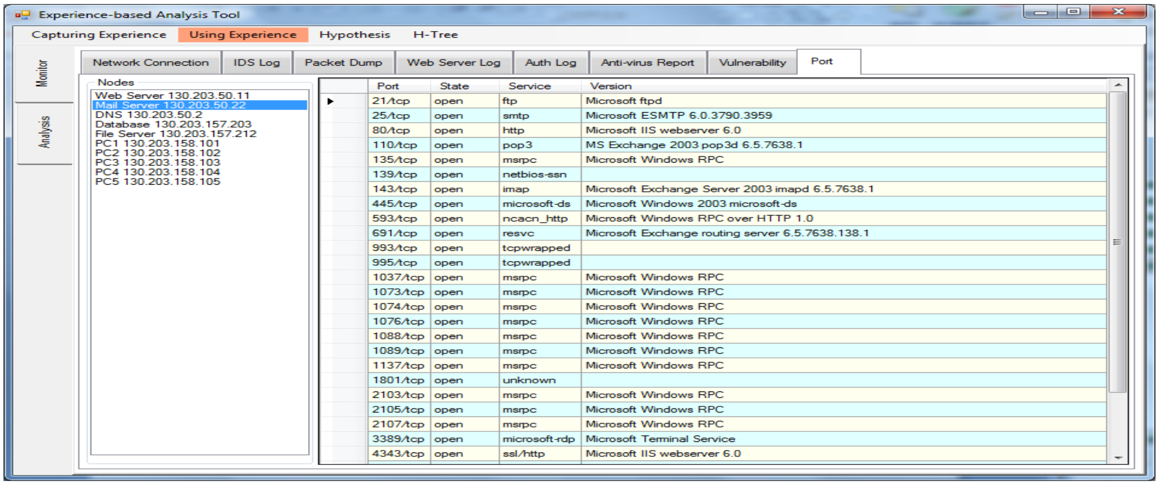
Date	Time	Source IP	Destination IP	Snort ID	Destination
05-05-2010	23:50:05	10.1.101.213	130.203.50.11	(16356)	WEB-IIS multiple extension code execution attempt
05-10-2010	01:26:21	50.100.1.31	130.203.50.22	(17300)	Privilege escalation attack causing access violation
10-02-2010	06:09:47	130.203.158.103	130.203.157.203	(17313)	ORACLE database server crafted view privelege escalation attempt
09-02-2010	06:09:47	10.1.101.213	130.203.50.2	(12008)	DNS cache poisoning attack causing malicious IP mapping

Chain1:
10.1.101.213--
>130.203.50.11
Chain2:
50.100.1.31--
>130.203.50.11
Chain3:
130.203.158.101-
>130.203.157.203
Chain4:
130.203.158.101-
>130.203.157.203
Chain5:
130.203.158.103-
>130.203.157.203
Chain6:
130.203.158.102-
>130.203.157.203
Chain7:
130.203.158.103-
>130.203.158.105

Select [Capturing Experience] or [Using Experience] on the top

Figure A-2. Snapshot of Tool Interface

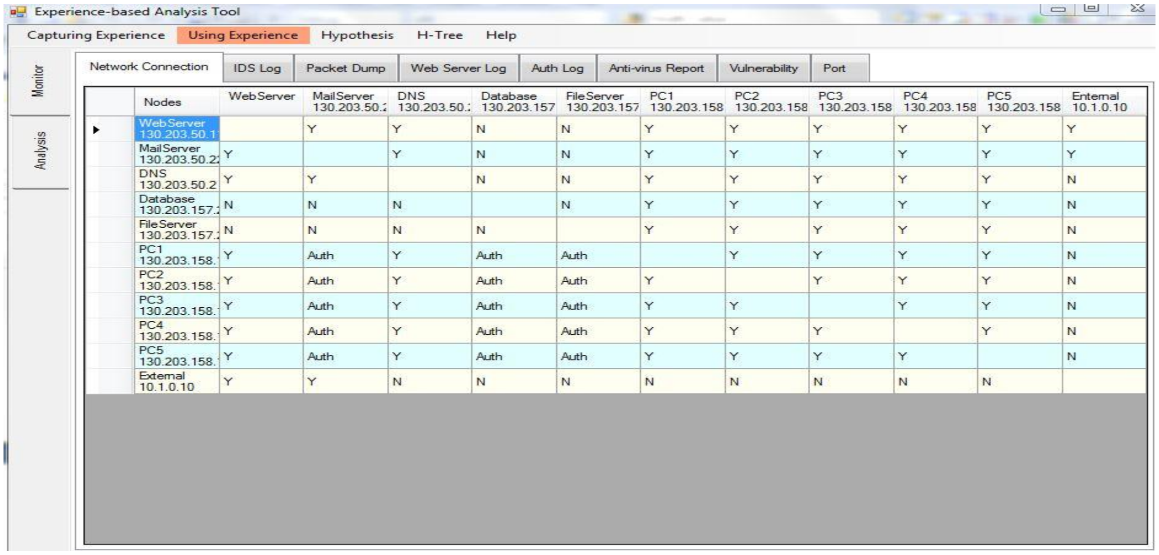
Figure A-3 and Figure A-4 show the monitor view of the tool where the analyst could browse through any of the data source that he would like to view. The observations are recorded via mouse clicks and keystroke events.



Nodes
Web Server 130.203.50.11
Mail Server 130.203.50.22
DNS 130.203.50.2
Database 130.203.157.203
File Server 130.203.157.212
PC1 130.203.158.101
PC2 130.203.158.102
PC3 130.203.158.103
PC4 130.203.158.104
PC5 130.203.158.105

Port	State	Service	Version
21/tcp	open	ftp	Microsoft ftpd
25/tcp	open	smtp	Microsoft ESMTP 6.0.3790.3959
80/tcp	open	http	Microsoft IIS webserver 6.0
110/tcp	open	pop3	MS Exchange 2003 pop3d 6.5.7638.1
135/tcp	open	msrpc	Microsoft Windows RPC
139/tcp	open	netbios-ssn	
143/tcp	open	imap	Microsoft Exchange Server 2003 imapd 6.5.7638.1
445/tcp	open	microsoft-ds	Microsoft Windows 2003 microsoft-ds
593/tcp	open	ncacn_http	Microsoft Windows RPC over HTTP 1.0
691/tcp	open	resvc	Microsoft Exchange routing server 6.5.7638.138.1
993/tcp	open	tcpwrapped	
995/tcp	open	tcpwrapped	
1037/tcp	open	msrpc	Microsoft Windows RPC
1073/tcp	open	msrpc	Microsoft Windows RPC
1074/tcp	open	msrpc	Microsoft Windows RPC
1076/tcp	open	msrpc	Microsoft Windows RPC
1088/tcp	open	msrpc	Microsoft Windows RPC
1089/tcp	open	msrpc	Microsoft Windows RPC
1137/tcp	open	msrpc	Microsoft Windows RPC
1801/tcp	open	unknown	
2103/tcp	open	msrpc	Microsoft Windows RPC
2105/tcp	open	msrpc	Microsoft Windows RPC
2107/tcp	open	msrpc	Microsoft Windows RPC
3389/tcp	open	microsoft-rdp	Microsoft Terminal Service
4343/tcp	open	ssl/http	Microsoft IIS webserver 6.0

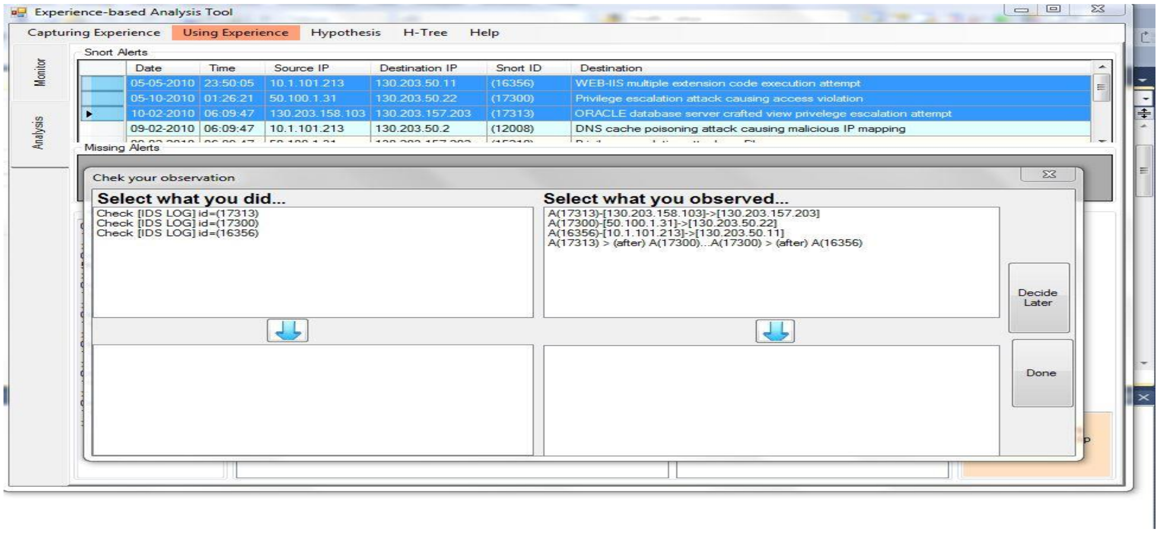
Figure A-3. Snapshot of Port Information Data Source



Nodes	WebServer	MailServer 130.203.50.2	DNS 130.203.50.	Database 130.203.157	FileServer 130.203.157	PC1 130.203.158	PC2 130.203.158	PC3 130.203.158	PC4 130.203.158	PC5 130.203.158	External 10.1.0.10
WebServer 130.203.50.1		Y	Y	N	N	Y	Y	Y	Y	Y	Y
MailServer 130.203.50.2	Y		Y	N	N	Y	Y	Y	Y	Y	Y
DNS 130.203.50.2	Y	Y		N	N	Y	Y	Y	Y	Y	N
Database 130.203.157.	N	N	N		N	Y	Y	Y	Y	Y	N
FileServer 130.203.157.	N	N	N	N		Y	Y	Y	Y	Y	N
PC1 130.203.158.	Y	Auth	Y	Auth	Auth		Y	Y	Y	Y	N
PC2 130.203.158.	Y	Auth	Y	Auth	Auth	Y		Y	Y	Y	N
PC3 130.203.158.	Y	Auth	Y	Auth	Auth	Y	Y		Y	Y	N
PC4 130.203.158.	Y	Auth	Y	Auth	Auth	Y	Y	Y		Y	N
PC5 130.203.158.	Y	Auth	Y	Auth	Auth	Y	Y	Y	Y		N
External 10.1.0.10	Y	Y	N	N	N	N	N	N	N	N	

Figure A-4. Snapshot of Network Connections Data Source

Figure A-5 shows the observation window that opens when analyst finishes his observations. This window allows the analyst to confirm the observations from the list he viewed.



	Date	Time	Source IP	Destination IP	Snort ID	Destination
	05-05-2010	23:50:05	10.1.101.213	130.203.50.11	(16356)	WEB-IIS multiple extension code execution attempt
	05-10-2010	01:26:21	50.100.1.31	130.203.50.22	(17300)	Privilege escalation attack causing access violation
▸	10-02-2010	06:09:47	130.203.158.103	130.203.157.203	(17313)	ORACLE database server crafted view privelege escalation attempt
	09-02-2010	06:09:47	10.1.101.213	130.203.50.2	(12008)	DNS cache poisoning attack causing malicious IP mapping

Select what you did...

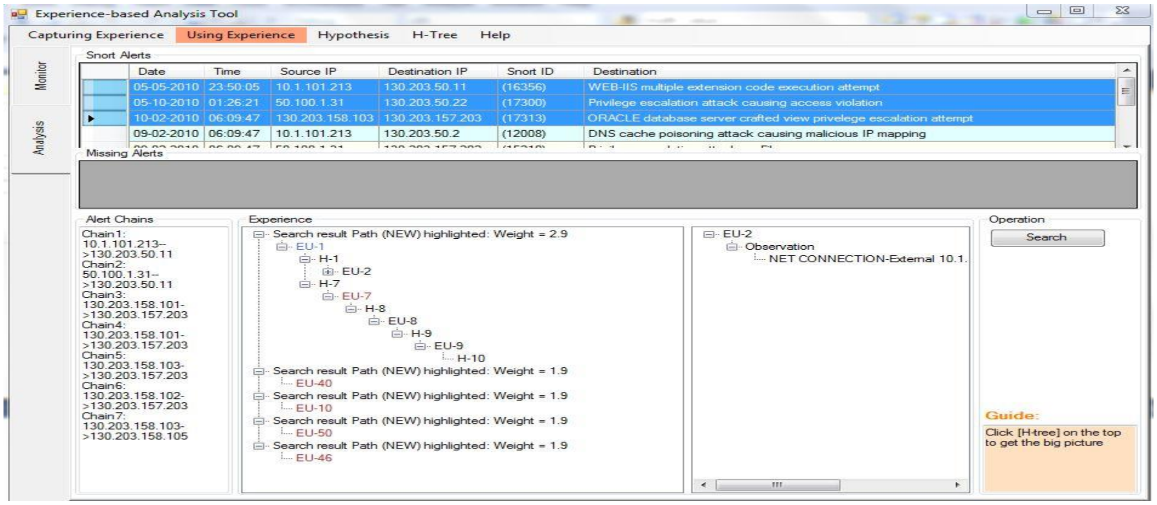
Check [IDS LOG] id=(17313)
Check [IDS LOG] id=(17300)
Check [IDS LOG] id=(16356)

Select what you observed...

A(17313)-[130.203.158.103]->[130.203.157.203]
A(17300)-[50.100.1.31]->[130.203.50.22]
A(16356)-[10.1.101.213]->[130.203.50.11]
A(17313) > (after) A(17300)...A(17300) > (after) A(16356)

Figure A-5. Snapshot of Confirmed Observations Window

Figure A-6 shows the results of the search. It shows the top 5 paths retrieved from ETs in the knowledge base based on the current context. The matched EUs are shown in red and the remaining EUs in the matched path in blue.



	Date	Time	Source IP	Destination IP	Snort ID	Destination
	05-05-2010	23:50:05	10.1.101.213	130.203.50.11	(16356)	WEB-IIS multiple extension code execution attempt
	05-10-2010	01:26:21	50.100.1.31	130.203.50.22	(17300)	Privilege escalation attack causing access violation
▸	10-02-2010	06:09:47	130.203.158.103	130.203.157.203	(17313)	ORACLE database server crafted view privelege escalation attempt
	09-02-2010	06:09:47	10.1.101.213	130.203.50.2	(12008)	DNS cache poisoning attack causing malicious IP mapping

Chain1:
10.1.101.213--
>130.203.50.11
Chain2:
50.100.1.31--
>130.203.50.11
Chain3:
130.203.158.101-
>130.203.157.203
Chain4:
130.203.158.101-
>130.203.157.203
Chain5:
130.203.158.103-
>130.203.157.203
Chain6:
130.203.158.102-
>130.203.157.203
Chain7:
130.203.158.103-
>130.203.158.105

```
Search result Path (NEW) highlighted: Weight = 2.9
  EU-1
    H-1
      EU-2
    H-7
      EU-7
        H-8
          EU-8
            H-9
              EU-9
                H-10
Search result Path (NEW) highlighted: Weight = 1.9
  EU-40
Search result Path (NEW) highlighted: Weight = 1.9
  EU-10
Search result Path (NEW) highlighted: Weight = 1.9
  EU-50
Search result Path (NEW) highlighted: Weight = 1.9
  EU-46
```

```
EU-2
  Observation
    NET CONNECTION-External 10.1.
```

Click [H-tree] on the top to get the big picture

Figure A-6. Snapshot of Results of Search

References

[1] Dain, O. and Cunningham, R. A heterogeneous alert stream into scenarios. *In Proc. of the 2001 ACM Workshop on Data Mining for Security Applications, pages 1-13*, Nov. 2001.

[2] Morin, B., Mé, L., Debar, H. and Ducassé, M. A logic-based model to support alert correlation in intrusion detection. *Information Fusion, 10(4), 285-299*, 2009.

[3] Tabia, K., Benferhat, S., Leray, P. and Mé, L. Alert correlation in intrusion detection: Combining AI-based approaches for exploiting security operators' knowledge and preferences. *In The third IJCAI-11 Workshop on Intelligent Security (Security and Artificial Intelligence SECART-11), (pp. 42-49)*, 2011.

[4] Ou, X., Boyer, W. F. and McQueen, M. A. A scalable approach to attack graph generation. *In Proc. of the 13th ACM conference on Computer and communications security (pp. 336-345)*, 2006.

[5] Chen, P. C., Liu, P., Yen, J. and Mullen, T. Experience-based cyber situation recognition using relaxable logic patterns. *In Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), IEEE International Multi-Disciplinary Conference on (pp. 243-250)*, 2012.

[6] Yen, J., McNeese, M., Mullen, T., Hall, D., Fan, X. and Liu, P. RPD-based hypothesis reasoning for cyber situation awareness. *Cyber Situational Awareness, 39-49*, 2010.

[7] Pirolli, P. and Card, S. The sense making process and leverage points for analyst technology as identified through cognitive task analysis. *In Proc. of International Conference on Intelligence Analysis*, 2005.

[8] Shrinivasan, Y. B. and van Wijk, J. J. Supporting the analytical reasoning process in information visualization. *In Proc. of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (pp. 1237-1246),* April 2008.

[9] Cuppens, F. and Ortalo, R. A language to model a database for detection of attacks. *In Proc. Of Recent Advances in Intrusion Detection (RAID 2000), pages 197-216,* September 2000.

[10] Debar and Wespii, A. Aggregation and correlation of intrusion-detection alerts. In *Recent Advances in Intrusion Detection, LNCS 2212, pages 85-103,* 2001.

[11] Cuppens, F. and Miege, A. Alert correlation in a cooperative intrusion detection framework *In Proc. Of the 2002 IEEE Symposium on Security and Privacy,* May 2002.

[12] Jha, S. Sheyner, O. and Wing, J. Two formal analyses of attack graphs *In Proc. of the 15th Computer Security Foundation Workshop,* June 2002.

[13] Ning, P., Cui, Y. and Reeves, D.S. Analyzing intensive intrusion alerts via correlation. *In Proc. of the 5th Int'l Symposium on Recent Advances in Intrusion Detection (RAID 2002),* October 2002.

[14] Ning, P., Cui, Y. and Reeves, D.S. Constructing attack scenarios through correlation of intrusion alerts. *In CCS Proceedings of the 9th ACM conference on Computer and communications security*.

[15] Ning, P. and Xu, D. Query optimization techniques for efficient intrusion alert correlation. *Technical Report TR-2002-14, North Carolina State University, Department of Computer Science,* September 2002.

[16] Ou, X., Govindavajhala, S. and Appel, A.W, MulVAL: A logic-based network security analyzer. *pp. 1-16.*

[17] Aamodt, A. and Plaza, Case Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications, Volume 7,* 1994.

[18] Zhong, C., Kirubakaran, D.S, Yen, J., Liu, P. Hutchinson, S. and Cam, H., How to use Experience in Cyber Analysis: An Analytical Reasoning Support System. *ISI Conference,* 2013.