Cyber attack simulation and information fusion process refinement optimization models for cyber security

Jason Kistner
Rochester Institute of Technology

CYBER ATTACK SIMULATION AND
INFORMATION FUSION PROCESS REFINEMENT OPTIMIZATION
MODELS FOR CYBER SECURITY

A Thesis

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Industrial Engineering

in the

Department of Industrial & Systems Engineering
Kate Gleason College of Engineering

by

Jason Kistner

B.S./M.S., Industrial Engineering, Rochester Institute of Technology, 2006

July 2006
M.S. DEGREE THESIS

The M.S. Degree Thesis of Jason Kistner
has been examined and approved by the
thesis committee as satisfactory for the
thesis requirement for the
Master of Science degree

Approved by:

Michael E. Kuhl
Dr. Michael E. Kuhl, Thesis Advisor

Moises Sudit
Dr. Moises Sudit
Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: Cyber Attack Simulation and Information Fusion Process Refinement Optimization Models for Cyber Security

Name of author: Jason Kistner
Degree: Master of Science
Program: Industrial and Systems Engineering
College: Kate Gleason College of Engineering

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:

I, Jason Kistner, hereby grant permission to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: Jason Kistner Date: ____________

Print Reproduction Permission Denied:

I, __________________________, hereby deny permission to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part.

Signature of Author: __________________________ Date: ____________

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive

I, Jason Kistner, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.

I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: Jason Kistner Date: ____________
Dedication

This thesis is dedicated to my professors who believed I could continue with higher learning and to my parents and friends who supported me every step of the way.
Acknowledgments

I thank Dr. Michael Kuhl for his continued support and guidance throughout the research, development and writing of this thesis. I also thank Dr. Moises Sudit for helping to guide this research as well. Also, thank you to both Dr. Kuhl and Dr. Sudit for convincing me to step up and do a thesis in the first place. Both of you, thank you so much.

I also thank Dr. Shanchieh Yang and Jared Holsopple for their thoughts and contributions, especially with Process Refinement and also for the excellent work they have done in fusion Level 3. Also, Adam Stotz and Rich Giomundo from the University of Buffalo, thank you for helping to educate me in networking and hacking, a tough task.

Much of this work would not be possible without the support of John Solerno, Doug Boulware, George Tadda and Mike Hinnman from the Air Force Research Lab so to those gentlemen, thanks very much. I also thank CUBRC at the University of Buffalo for their support as well.
Abstract

Cyber crime is an increasingly prominent threat to all aspects of society including businesses, government, banks, transportation, and individuals. The security of computer networks is dependent on the ability to recognize and defend against malicious cyber attacks. The goal of this thesis is to utilize operation research techniques to create tools that will significantly contribute to cyber security. A simulation framework and template is developed to efficiently represent computer networks and cyber security intrusion detection systems. The simulation is capable of generating complex cyber attacks based on the computer network configuration and the capabilities of the attacker. The simulation results in alert messages corresponding to attack actions and ordinary network behavior which are typically used by situational awareness tools or systems administrators to identify and take action against the attack. Through verification, validation, and an experimental performance evaluation, the simulation model is shown to be an effective tool to enable testing of situational awareness tools and for determining network vulnerabilities. In addition, this thesis extends the highly effective information fusion methods of situational awareness and threat assessment by introducing a method of adaptive process refinement for cyber security. The adaptive process refinement model utilizes integer programming optimization to improve the success of cyber attack detection, tracking, and identification. The process refinement model is designed to dynamically provide recommendations for optimal allocation of network detection resources subject to processing capacity, current attack activity, and network vulnerabilities. The cyber attack simulation methodology is utilized to create a set of attack scenarios on computer networks that are used conduct an experimental performance evaluation of the adaptive process refinement model to determine its capabilities and limitations. The simulation and process refinement methods provide operations research tools that will help to advance the field of cyber security.
Table of Contents

1. Introduction .......................................................................................................................... 1

2. Problem Statement ................................................................................................................ 4

3. Literature Review .................................................................................................................. 8
   3.1 Modeling and Simulation of Computer Networks ................................................................. 8
   3.2 Modeling and Simulation of Hacker Attacks .......................................................................... 10
   3.3 Detecting and Identifying Attacks on a Network ................................................................. 12
   3.4 Information Fusion and Applications in the Cyber Domain ................................................. 14
   3.5 Summary ............................................................................................................................. 16

4. Development of Simulation Methodology ............................................................................. 18
   4.1 Original Simulation Model .................................................................................................... 18
      4.1.1 Creation of Virtual Network ........................................................................................ 19
      4.1.2 Attack Flow Through the Network ............................................................................... 19
      4.1.3 Attack Specification ...................................................................................................... 21
      4.1.4 Noise Generation ........................................................................................................ 23
      4.1.5 Alert Production ........................................................................................................... 23
      4.1.6 Example Simulation Scenario ....................................................................................... 24
   4.2 Formulation of Automatic Attack Generation Methodology ............................................. 25
      4.2.1 Inputs to the Automatic Attack Generation Methodology ........................................... 26
      4.2.2 Automatic Attack Generation Logic ............................................................................. 31
      4.2.3 Capabilities and Limitations ......................................................................................... 40
   4.3 Addition of Further Machine Attributes ............................................................................ 42
      4.3.1 Attribute Definitions ..................................................................................................... 43
      4.3.2 Attribute Implementation in Simulation Template .......................................................... 44
      4.3.3 Action File Definition .................................................................................................. 46
      4.3.4 Integration .................................................................................................................... 47
   4.4 Verification and Validation .................................................................................................. 48
      4.4.1 Verification of Manual Attack Generation .................................................................... 49
      4.4.2 Validation of Automatic Attack Generation Methodology .......................................... 50
      4.4.3 Noise Production, Template, and File Creation Verification ........................................ 55
      4.4.4 Summary ....................................................................................................................... 58
5. Investigation of Process Refinement in the Cyber Domain ........................................... 59

5.1 Process Refinement Conceptual Model ............................................................................ 59
5.2 Sensor Management and Process Refinement .................................................................. 61
5.3 Process Refinement Mathematical Model ......................................................................... 62
5.4 Model Implementation .................................................................................................... 67

6. Evaluation of Process Refinement Performance .............................................................. 81

6.1 Test System Setup ........................................................................................................... 81
6.1.1 Network Topology and Information Definition .............................................................. 82
6.1.2 Testing Setup for Process Refinement ........................................................................... 84
6.1.3 Attack Scenarios and Evaluation Metrics ...................................................................... 85

6.2 Single Attack Scenarios .................................................................................................. 86
6.2.1 Random Scenario Results ............................................................................................ 86
6.2.2 Process Refinement Recommendations ........................................................................ 88
6.2.3 Comparison of Random Detection to Process Refinement Detection ......................... 91

6.3 Multiple Attack Scenarios .............................................................................................. 91
6.2.1 Random Scenario Results ............................................................................................ 91
6.2.2 Process Refinement Recommendations ........................................................................ 93
6.2.3 Comparison of Random Detection to Process Refinement Detection ......................... 96

6.4 Process Refinement Failure ............................................................................................ 97
6.4.1 Failure Reasons ............................................................................................................ 97
6.4.2 Example of Failure ....................................................................................................... 98

6.5 Testing Conclusions ........................................................................................................ 100

7. Conclusions and Recommendations for Future Research ........................................... 102

References .......................................................................................................................... 105

Appendices .......................................................................................................................... 110

Appendix A: Original Simulation Model ............................................................................. 110
Appendix B: Original Simulation Model - Attack Example .................................................. 116
Appendix C: Automatic Attack Generation Code Logic ....................................................... 118
Appendix D: Verification and Validation ............................................................................. 121
Appendix F: Process Refinement Testing Setup - Single Attack Scenarios .............. 134
Appendix G: Process Refinement Testing Setup - Multiple Attack Scenarios ........... 138
Appendix H: Single Attack - Process Refinement Recommendations ..................... 146
Appendix I: Multiple Attack – Process Refinement Recommendations .................... 150
Appendix J: Process Refinement Failure ................................................................... 157
Appendix K – Simulation and Process Refinement Files Stored on CD ..................... 161
1. Introduction

The development of computer networks and the use of the Internet have grown rapidly since their early use some forty years ago. In that time span, both the size and complexity of operations in computer networks have grown enormously. Many administrators are finding themselves hard pressed for simple cyber security solutions. In addition, more and more organizations and individuals are using networks to store and transfer sensitive data such as social security numbers, credit card information, company secrets, and even government classified information. As a result, crimes such as thievery and sabotage have shifted from the physical domain to the cyber domain. Today's bank robbers and terrorists do not always operate in the open, but often from a computer terminal, stealing money and personal information, or corrupting and exploiting confidential data. In large networks, the attacks or thefts may not be realized until the hacker is long gone. Crime has become easier to commit and harder to detect (Furnell, 2003).

Security measures are being developed to help counteract this cyber crime paradigm. Companies use tools such as firewalls, router filters, access control lists, and network management protocols to limit access and transmissions of malicious packets (Cisco Systems, 2003). These tools are designed to prevent unauthorized users from gaining access to the network and block potentially harmful files from entering the network. Unfortunately, even with these security methods, intrusions still occur and other methods of security are needed (Kemmerer & Vigna, 2002).

An intrusion detection system (IDS) is used in many cases to detect malicious attacks that penetrate other security tools in place. In the ever-changing environment of computer technology, attackers identify vulnerabilities in the large range of existing computer systems
faster than systems administrators can fix them (Bandy, Money & Worstell, 2002). As a result, the IDS is often used as a last line of defense for a network. Rather than preventing access and transmission of data, intrusion detection systems are used to identify unusual or prohibited actions that occur in the network and report the actions in some way, often in the form of an alert. One of the duties of an analyst is to parse through the IDS alerts and identify where they feel an attack may be occurring.

Systems administrators performed early intrusion detection in a manual way. Administrators would examine network traffic and look for anomalous activity. As network sizes grew, this became labor intensive and administrators started using audit logs, which are a list of all actions a computer has performed, for their analysis. Unfortunately, this could only be used as a forensic tool to determine the causes of past security incidents. Later, programs were developed that could analyze the audit logs but they were not fast enough to detect security breaches in time to prevent the consequences (Kemmerer & Vigna, 2002).

In the early 1990s, the first intrusion detection systems to analyze network traffic in real-time were developed. At this point, some attacks could be stopped before they reached a goal and some attacks could be prevented before an intrusion occurred. However, as networks continued to grow, intrusion detection systems were required for use in large networks. The detection capability requirements also grew with the enormous number of vulnerabilities in systems being found. Current intrusion detection system development focuses on deployment in large networks and increased detection capability (Kemmerer & Vigna, 2002).

Today, intrusion detection systems are used in many large companies as part of their security policy. However, due to the complexity of modern networks, intrusion detection systems have become extremely complicated and often difficult to manage. Many larger
networks could experience hundreds of thousands of alerts each day, with the rate increasing if the network contained information that would be especially valuable to hackers (Sudit, Stotz & Holender, 2005). As a result, the complexity of analyzing all of the alerts produced has increased significantly, and techniques are needed to assist in the interpretation of the alerts.

One technique used for analyzing large amounts of data from multiple sources is called information fusion. Information fusion is a closed-loop process for analyzing data from one or many sources and creating relevant and useful information, which an analyst can use to make decisions. Fusion operates at five different levels: Level 0 - Sub-Object Data Association and Estimation; Level 1 - Object Refinement; Level 2 - Situation Refinement; Level 3 - Impact Assessment; and Level 4 - Process Refinement (Llinas, 2002). Each level is discussed in detail in section three.

Many Industrial Engineering techniques can be used in the fusion process to provide solutions to information fusion problems at various levels. Specifically, operations research and simulation have applications in this area (Bistarelli, Foley & O’Sullivan, 2004; Sudit, Stotz, & Holender, 2005; Kuhl & Kistner, 2005). This thesis focuses on using simulation to create multistage cyber attacks on a simulated network and using mathematical programming to optimize the processing of information by a cyber security information fusion system.
2. Problem Statement

The goal of this thesis project will be twofold. First, to develop a simulation methodology for modeling multistage cyber attacks which can have various uses in the cyber domain, including testing of new cyber security assistance tools. Second, to create a system that can refine and improve the way in which data is analyzed, so that an analyst is presented with the most relevant information for the current situation.

The first goal stems from the need to test and evaluate cyber security tools in a realistic computer network environment. Traditionally, physical computer networks have been used to perform and record cyber attack scenarios. Although realistic, these exercises involve high costs resulting from setting up the physical network (and reconfiguring the network for each scenario), experts needed to run the attacks, potential loss due to damage resulting from attack actions (such as viruses), and long setup and execution times.

Therefore, this research investigates the development of a cyber attack simulation model to accurately and efficiently simulate cyber attacks in a computer network. The simulation methodology can provide a flexible environment for constructing a computer network and efficiently specifying and generating cyber attacks.

The information fusion process will be the base methodology for the second goal of this thesis. Fusion involves correlating data at different levels to structurally analyze the data and provide better understanding of what the data means. Fusing the data provided by IDSs provides a situational awareness for the analyst that can tell them what is currently happening in the network. The fusion process can also provide an assessment of the potential impact or threat of the current activities that are occurring in the network. Finally, the process will also give advice to the analyst on how they should refine the detection process which is called Process
Refinement. Process Refinement is the area that will be investigated to help improve the data analysis that an analyst must perform.

The problem of analyzing all of the intrusion alerts has been investigated by many different sources, such as: Sudit, Stotz, & Holender, (2005); Mathew et al., (2005); Undercoffer & Pinkston, (2002); Bass, (2000). In general, the analysis is done through correlation. Researchers have developed systems that correlate the alerts of the intrusion detection systems through a variety of correlation criteria. Information fusion is a structured method, which may provide more straightforward results than other systems.

Process refinement is relatively new to the cyber domain, though extensive work has been done in other areas (Musick & Malhotra, 1994 and Malhotra, 1995). Process refinement involves altering the real-world sensing environment in different ways so that the other levels of fusion can provide better information or altering the fusion process itself to improve the way in which data is processed. Unlike other levels of fusion, the refinement process is concerned with all other levels rather than just the level before or after, so that the system as a whole can increase performance (Llinas, 2002). Also unlike the other levels, process refinement provides a recommendation to the analyst on how to change the real world environment (Malhotra, 1995). All other levels deal only with informing the analyst of the situation or status.

In the cyber domain, process refinement does not necessarily mean the relocation of sensors. The “costs” of sensing in the cyber domain are minimal in many cases due to open source software availability and high-speed networking. Process refinement also does not necessarily mean changing the sensor itself. With today’s computing technology, collecting larger quantities of data in the form of alerts is not necessarily the problem. Rather, the problem is processing the information that will help to identify and potentially stop an attack before too
great of a loss is incurred. The fact is that although IDS’s may produce large numbers of alerts, in most cases a relatively small number of the alerts correspond to malicious attacks. Consequently, selectively processing the alerts that have the best potential of identifying the presence and progress of an attack, would allow for near real-time decision-making.

Therefore, the focus of this research is to create a system that can identify what specific IDS information to process, as well as when and how to process the information. The system could be used to provide more accurate results faster in the other levels of fusion. Rather than managing sensors themselves as suggested in Musick & Malhotra, 1994 and Malhotra, 1995, managing and refining the fusion process as a whole is necessary in the cyber domain.

The objectives of the simulation methodology and the process refinement system include:

1) Developing an initial simulation methodology to model multistage cyber attacks. The simulation methodology should include an automatic attack generation methodology. Using the automatic attack generation methodology, the simulation could run under a variety of conditions with minimal setup required.

2) Creating a method for simulating a computer network involving machines and switches. The machines and switches should contain attributes which can be used to make them unique. This will allow for more specific modeling and will provide more realistic attack scenarios.

3) Verifying and validating the attack simulation methodology. The methodology should be validated to ensure that the data being generated truly reflects the cyber attack environment.
4) Obtaining the most useful information given the processing capacity of the resources. By doing this, the system will be able to optimize the number of alerts that need to be processed to obtain the best information about the situation.

5) Having the capability to capture new information and analyze information in real-time as events occur. When a new attack or attack step occurs, the process refinement system should suggest ways in which to reallocate resources to the new problem area. The system should make optimal choices between existing problems involving where to allocate resources.

6) Measuring the effect or benefit of any suggested refinements. In particular, metrics will be created to measure the benefit of using Process Refinement over static detection scenarios.

Process refinement is an area of fusion, which has not yet been integrated in the cyber domain. By constantly refining and updating the fusion system, the system can keep pace with the evolving vulnerabilities and actions that hackers can take as attacks progress. By allowing the system to pull from more accurate and more relevant data sources, the fusion system will be able to provide better information to the analyst. Ultimately, the fusion system is designed to provide the analyst with the most relevant data based on the current situation and provide the analyst with a good picture of what is wrong with the network. A direct result of this process may also be a reduced amount of information that the analyst must view. By constantly refining this process, the analyst will receive more pertinent and useful information.
3. Literature Review

Related work to process refinement and attack simulation in the cyber domain is very prevalent in some aspects but not in others. The following topics will be discussed: Modeling and simulation of computer networks, modeling and simulation of hacker attacks, detecting and identifying attacks on a network, and information fusion and applications in the cyber domain.

3.1 Modeling and Simulation of Computer Networks

Modeling of computer networks can take a variety of forms. Models involving graphs and graph-based algorithms have been developed to analyze computer networks. Rule based programs and systems have also been developed to test or simulate different parts of computer networks. These applications are not only used to test the security of computer networks but also for more traditional model analysis such as new technology feasibility, learning environments and many more.

Graph theory is an area of study in which groups of nodes and edges connected together form a network, which could model many different entities, including a computer network or even the Internet. As such, graph theory techniques have been used to evaluate computer networks in many different ways. Specifically, graph theory problems, such as the shortest path problem, have been integrated into computer networking to analyze a network. Solving the shortest path problem for a computer network would show the analyst the shortest way to get from a starting point to every other point in the network. This information could be used to determine improvements to a network’s security policy so that the hacker either has no path to secret information or must take a very long route to get to the information, risking detection the whole way. The shortest path problem is solved by Dijkstra’s Algorithm, which is used in Fitch & Hoffman, 1993 to evaluate a computer network.
Simulation has been used to test and evaluate systems without interfering with the system in situations such as manufacturing or service. Currently, researchers are attempting to do similar types of analysis on computer networks. Since experimenting with attacks on a computer network can often be harmful to the network, a new format of testing is required. Some analysts evaluate their own network by creating test bed systems that are not connected to the network, but this requires both hardware and knowledgeable testers. Others have used simulation to test their network by building a virtual network and simulating hacker attacks (Zaliwski, 2005). With this method, no hardware is needed except for the machine running the simulation, and there is no risk of permanent damage. However, for the simulation results to be accurate, an extensively detailed model must be created (Kuhl & Kistner, 2005).

In addition, simulated computer network environments can be used to teach administrators about new attacks and how to mitigate the consequences of those attacks (Zaliwski, 2005). This method of learning about hacker attacks and their consequences can also be integrated into the redesign of security tools and the development of new ones, such as intrusion tolerant systems architecture (Research Triangle Institute, 2005). Intrusion tolerant systems architecture involves adapting to new attacks as they arrive using a fault tolerant based methodology applying to existing and dynamic faults.

In general, security based computer network modeling and simulation is used to find problems or issues with a computer network without consideration of a particular type of attack. By examining the network and finding different exploitable or vulnerable points, an analyst can try to patch those areas. The problem with this method is that it exists in a reactive mode. Hackers, rather than administrators, more often find new vulnerabilities at an ever-increasing rate (McClure, Scambray & Kurtz, 2001). As a result, an administrator using these types of reactive
tools may not be able to keep up with the increasing complexity and ingenuity of the modern hacker.

3.2 Modeling and Simulation of Hacker Attacks

Modeling and simulation of hacker attacks builds upon modeling and simulating of computer networks in that researchers wish to model and simulate how different hacker attacks can move through the network. Some researchers are concerned with the speed of an attack and how easily the attack spreads (Symantec, 2005) while other researchers are more concerned with how to adjust the security policy to prevent particular attacks (Seo & Cho, 2003). As with the previous topic, the analysts are attempting to figure out ways in which the network is vulnerable. In this area, they are simulating attacks and in some cases, coordinated sets of attacks, to determine the resulting impact on the network.

The main objective in most simulations of hacker attacks is to test a particular security policy framework against a variety of common attacks to determine ways in which the security policy fails. By creating various types of attacks, analysts can determine which attacks would be successful and which would fail by running simulations of those attacks on a simulated network. Once successful attacks are identified, the analysts can adjust firewalls, IDS and other policy components to attempt to prevent them. The simulation can then be repeated under the adjusted network to test the results of the change (Korea Information Security Agency, 2005).

Models have also been developed to identify which parts of a network are vulnerable to types of attacks. This method, which is like preemptive attack determination, can take many forms. For example, Shyner, Haines, Jha, Lippman and Wing (2002) create “attack graphs” based on some type of hacker goal such as retrieve information x from machine y. The system
builds a graph, which identifies ways in which a hacker might achieve the goal that was input. An example of an attack graph is shown in Figure 1.

![Attack Graph](image)

**Figure 1 - Attack Graph**

Attack simulations can also be used to generate many different types of data depending on what the application calls for. The data generated could be of immediate use in improving the security of a network, or could be used in testing another system. In a methodology created by Kuhl and Kistner (2005), the attack simulation is used to generate sample intrusion detection system output, which can be used in testing other systems. The simulation model development of this thesis builds off of the work of Kuhl and Kistner.

A comprehensive use of simulation for improving security of networks and training of administrators is presented in the RINSE methodology (Liljenstam, Liu, Nicol, Yuan, Yan & Grier, 2005). The RINSE methodology uses a simulation environment in a type of gaming environment in which a “game manager” can run attacks on target machines and the “players” can diagnose what attacks are occurring, use different security measures to block the attacks and
interact with the target machines to adjust their properties. This allows a user to learn about the type of attacks that hackers use today and can provide a framework for testing attack mitigation for newly developed attacks. The RINSE framework diagram is shown in Figure 2. This area of research is very useful in training administrators how hackers’ attacks progress and affect a network.

Figure 2 - RINSE Framework

Simulations can also be useful in identifying ways to eliminate the attacks before they can occur. A major draw back of these methodologies, however, is that in large networks the analysis of different attacks can rapidly grow complex. Also, it is time-consuming and tedious to analyze every different combination of attack.

3.3 Detecting and Identifying Attacks on a Network

The research that involves detecting attacks on networks can be split into two different sections based on when the detection occurs. The most common is real-time attack detection using intrusion detection systems or other types of logging systems. Also, a process that is
becoming obsolete with new technology is the forensic analysis of logs to determine when and how an attack occurred.

Most real-time detection systems integrate with intrusion detection systems. The idea behind these systems is to correlate the alerts that seem to go together based on some alert attributes. By correlating many alerts together, the system can create a string, which may represent a hacker’s progression through a network (Sudit, Stotz & Holender, 2005).

The use of logs and network traffic data as a forensic tool has started to become obsolete with the newest technology. IDSs are becoming faster and easier to update, in addition to becoming open source so everyone can afford them. As a result, forensic analysis is not often used except for identifying unknown successful attacks (Kemmerer & Vigna, 2002).

The research involving identifying attacks is an extremely important one because it can allow an administrator to block progressive or future attacks stemming from an original that has been identified. Identifying attacks on networks involves detecting the attack and piecing together information to determine what the attack is trying to accomplish. This can be achieved by a single IDS alert or through complex analysis of multiple information sources.

The types of attack paths that can occur in a computer network have also been studied by many different groups. Neumann and Parker (1989) classify attacks in terms of technique. This indicates the type of damage caused or actions that the hacker is trying to take. Lindqvist and Johnson (1997) expand on this formula, with classification by more than one category. Categories suggested are intrusion technique, intrusion result and others. Further research has been conducted to identify what makes an event, attack or incident and what features each may have (Howard & Longstaff, 1998). These features include possible attackers, attack tools, vulnerabilities, results and objectives. Other research has focused on classifying cyber attacks by
focusing on the target of the attack (Undercoffer & Pinkston, 2002). Mathew, Britt, Giomundo, Upadhyaya, Sudit and Stotz (2005) suggest a method for understanding multi-stage attacks and how to relate individual attacks to their role in reconnaissance, intrusion, escalation of privilege and goal in a large cyber attack. In their work, Mathew et al. create a framework in which nodes corresponding to scenario graphs are activated based on intrusion information provided by IDS. The scenario graph builds and nodes connect together as further attack steps are instantiated and detected by the IDS.

Yang and Holsopple (2005) discuss a framework for analyzing the threat of cyber attacks in a dynamic manner. Threat is analyzed by examining three attributes of an attack: capability, opportunity and intent. Capability refers to the hacker's ability to execute different actions against a network. Opportunity is based on what the hacker can do next. This is determined by the vulnerabilities of the network, the information that the hacker has already gained and the topology of the network. Intent is what the hacker intends to accomplish with an action or series of actions. The authors suggest that this is very difficult to achieve in the cyber domain. Examining the attack path already traveled by the hacker is suggested as a way to determine the hacker's intent.

3.4 Information Fusion and Applications in the Cyber Domain

The study of multi-source information fusion began with military applications in which accurate and relevant information was required about attributes of an opposing force, such as location, strength, movement direction, etc. At the time, using one sensor was simply not enough to detect all of these different attributes. A system was required to integrate the information from many different sensors to provide an analyst with a good view of the overall situation, or what is known as situational awareness.
As a result of this need, the first models of the information fusion process were born. There are five levels in the information fusion process as identified by the JDL (Joint Directors of Laboratories) Fusion Model, each level further refining the data presented by different sensors until an overall awareness of the situation can be presented (Linas, Bowman, Rogova, Steinberg, Waltz & White, 2004). For simplicity, Level 0 will not be included in this discussion because it is unrelated. A diagram of the JDL Fusion process is shown in Figure 3.

<table>
<thead>
<tr>
<th>Level</th>
<th>Processing</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 0</td>
<td>Sub-object Data Association &amp; Estimation</td>
<td>Pixel/signal level data association and characterization</td>
</tr>
<tr>
<td>Level 1</td>
<td>Object Refinement</td>
<td>Observation-to-track association, continuous state estimation (e.g. kinematics), and discrete state estimation (e.g. target type and ID) and prediction</td>
</tr>
<tr>
<td>Level 2</td>
<td>Situation Refinement</td>
<td>Object clustering and relational analysis, to include force structure and cross force relations, communications, physical context, etc.</td>
</tr>
<tr>
<td>Level 3</td>
<td>Impact Assessment (Threat Refinement, Threat Intent Estimation, Event Prediction)</td>
<td>Consequence prediction, susceptibility and vulnerability assessment</td>
</tr>
</tbody>
</table>

**Figure 3 - JDL Information Fusion Model**

At Level 1, object refinement, techniques such as correlation, statistical estimation and pattern recognition are used to identify what different sensors are detecting similarities and differences in. The most common method used in the cyber domain is correlation (Sudit, Stotz & Holender, 2005, Bass, 2000, and Julisch, 2003). Correlation may involve aligning all observations in the same frame of reference (time, space, etc) and assigning weighted metrics.

In Level 2, situation refinement, statistical techniques or knowledge-based models are used. Situation refinement involves aggregating sets of alerts based on a variety of
characteristics. Alerts produced by intrusion detection systems are aggregated based on a number of factors such as source and target IP address, protocol, time, etc (Bass, 2000).

Level 3, impact assessment, attempts to estimate the result or impact of potential situations that levels one and two have defined. This level is also known as threat assessment. Threat assessment and impact assessment are technically the same level of fusion but threat assessment tries to predict not only the impact of the current situation but also what an attacker might try to do next. Techniques of level three analyses include Bayesian networks and Markov models as well as many others. Threat assessment in the cyber domain incorporates impact analysis of current hacker attacks. Yang and Holsopple (2005) also present research investigating threat assessment in the cyber domain that involves using an information graph to determine what information a hacker can obtain using the information they currently have.

Level 4, process refinement, deals with the management of the sensors and the management of the fusion process itself. This area has been researched and defined in many military applications such as aviation sensing (Musick & Malhotra, 1994 and Malhotra, 1995). However, to date no research has been conducted in the area of process refinement in the cyber domain.

3.5 Summary

Modeling and simulation methodologies are used in many different applications of security analysis. However, the general framework of network simulation, although it may be designed differently for different systems, operates in very similar ways. The attributes of the network that are modeled are very similar, as is the way that entities might operate within the network. The main difference is the level of detail in which the network is simulated. Some simulation models use very detailed attributes of the network (Kotenko, 2003) while others use
only the main attributes such as topology and IP address (Kuhl & Kistner, 2005). The level of
detail required depends on the application. However, any framework, no matter how detailed,
can be used to produce a variety of different results including recommendations for
improvement, training, and data generation.

Information fusion is a process that is still relatively new to the cyber domain and as
such, has not been fully developed. Applications of levels one and two processes have been
developed, however, work in the area of level three is just beginning. Level four processes such
as sensor location have been analyzed (Graham, 2005 and Calabrese, 2002), but the refinement
of the fusion process has not been investigated. Due to the changing environment in which
systems administrators operate, any tool that assists an administrator must be as dynamic as the
environment. A fusion process without process refinement will not be able to adapt to a
changing environment and will not last long in the dynamic cyber domain.
4. Development of Simulation Methodology

The cyber attack simulation methodology can assist in training administrators and testing cyber security assistance tools such as cyber security information fusion systems. The methodology should be capable of modeling a variety of cyber attacks on a customizable simulated network. The methodology should incorporate a method for generating attacks quickly and with minimal user bias. The simulated network should contain attribute detail of the machines and switches contained in the network. Finally, the model should be validated to ensure that outputs of the methodology reflect real output in the cyber domain.

An initial simulation model was created for the Air Force Research Lab in Rome, NY (Kuhl & Kistner, 2005). The Air Force required a simulation methodology to model multistage, coordinated cyber attacks performed on a virtual network and the data that would be generated by intrusion detection systems placed at different locations in the virtual network. This model provides the basis for the development of the overall cyber attack simulation methodology.

Improving the cyber attack simulation methodology will allow for better applications of the simulation methodology in general and allow the simulation methodology to assist in the testing of the process refinement system. Development of the simulation methodology involves four main steps: creation of an initial multistage cyber attack simulation methodology, formulation of a methodology for automatic attack generation, addition of further machine attributes to current network modeling template, and verification and validation of the updated methodology.

4.1 Original Simulation Model

This section describes the original simulation model built to model multistage cyber attacks. The simulation was created in the Arena simulation package produced by Rockwell
Software. The model consisted of five main pieces: creation of virtual network, attack flow through the network, attack specification, noise generation and alert production.

4.1.1 Creation of Virtual Network

To create a virtual network, inclusion of machines, which process data on a network, and switches that route data through a network, was necessary. In order for the simulation model to be robust, it was necessary to allow dynamic creation of virtual networks. This meant that network topology and information had to be created by the user rather than incorporated into the simulation logic.

The Arena software uses templates to provide standard modeling techniques to model developers. A new template had to be created for the purpose of building a computer network. The template created provided the user with a means of building the network topology and specifying the network information through a graphical means. By dragging and dropping the icons into the model development window, the user could quickly build the network. Each icon created needed to have some information detailed about it such as the IP address for machines and the addition of an IDS sensor for the switches. Another critical piece of information that the user must specify is whether a machine is internal or external. This determines whether the machine is accessible from the Internet. If the machine is accessible from the Internet, then it can be a starting point for an external threat attack. The template that was created and the information required of each instance of a template icon can be found in Appendix E.

4.1.2 Attack Flow Through the Network

Modeling the attack flow through the network is where most of the simulation logic was required. It was necessary to model the movement of each attack through the computer network. Attacks could possibly occur at the same time, sequentially or in any other way based on the user
input. The attacks would progress through the simulation logic, looping back to the beginning if there are multiple steps to the attack. The attacks may also loop if they are unsuccessful. A diagram of the entity flow can be found in Appendix E.

Entities which represent attacks are first created by a create block and then assigned a unique attack number at the corresponding assign block. The VBA block samples some attack attributes based on the way each attack was setup by the user. The next assign block gives the attack entity the attributes for the first step of its attack. One of the attributes assigned is the target IP address that is used in the FindJ block. The FindJ block finds the station associated with the target IP address and then the station number is assigned as an attribute. The entity is then delayed for a period of time corresponding to user input and then routed to the station that it was assigned.

The entity then travels into another sub-model of simulation logic. This is the main attack routing area in which the steps of the attack are executed and any repetition of a step due to step failure is made. The entities start at the station block and are transported to a branch block. The branch block will send copies of the entity to another area for processing of alerts. The area that the entity travels to next is another branch block. This branch block determines if the attack step was successful. Success or failure is determined by comparing the success probability specified by the user to a random sample from a uniform(0,1) distribution. If it was successful it will travel to another piece of logic to be assigned a new attack step, step attributes and routed to another station much like it did when it was created. If the step fails, the entity will not be assigned a new attack step but will follow the same path as successful entities. However, since it has not been assigned a new attack step, the attributes assigned will be the same as the
previous iteration. The entity will then be routed back to the same station and repeat the process which it just went through. The entity will continue doing this until successful.

The VBA block in the last line of the logic specifies the next step information. This VBA block samples from the user input to determine the next target and the type of attack to perform. The FindJ, Assign, Delay and Route blocks work the same as they did when the entity was created. When an entity reaches the last step of its attack and is successful, it is routed to a final assign block to indicate the attack completion and then disposed.

4.1.3 Attack Specification

Attacks can be specified using a graphical user interface or through an input file. The input file could be created by any means but for each attack scenario created with the GUI, an input file is generated. This allows the user to run an attack scenario multiple times without needed to input the information multiple times. The GUI was designed in Visual Basic which is the support language incorporated into the Arena software.

Using the interface, the user specifies a unique name for the attack scenario, as well as some other general scenario information. This includes the distribution for sampling of delay times, the method in which to specify delays (specific or sample over an interval), the time to run the simulation after the last attack has been completed as well as some information about the noise (this will be covered in the next section). The user can then specify up to ten different attacks with up to 30 steps in each attack.

When specifying an attack, the user must first give the attack a unique name and input the number of steps that the attack will contain. When the user sets the number of steps by pressing the Reset Form button, the multi-page will adjust its number of tabs to allow the user to specify information for each step. In each step, the user must specify the IP address from which the
attack will originate and the IP address for the target machine. After specifying the source IP address of the attack, the user can click the Update Target IP button to update the list of IP addresses, which can be reached by the source IP address based on the network topology. Next the user can specify the success probability of the step and whether they want the information in the step to be encoded. Many hackers will encode their information to attempt to hide it from the intrusion detection system. This problem is handled by a preprocessor system that evaluates information packets not by the signature but by the format of the information (Snort, 2005). The user then specifies the specific action that they want to perform. These actions are split into five main categories, each category having subcategories as well. The categories and subcategories are listed in the Literature Review. The user can specify a category, a subcategory or a specific action. If a specific action is not chosen, an action will be sampled from the subcategory if one was specified or from the general category. If no information is input for the action, an action will be sampled at random from the entire population. The buttons located under the action input boxes will update the information in consecutive boxes to give the user a refined list to choose from. Lastly, the user can specify the specific delay time for each attack (if this was the method chosen on the main screen) or the time interval over which the entire attack should take place. The user also can specify the time to delay the start of the attack.

Clicking done will take the user back to the main screen and the attack that was just created will appear in the list box. The user can then add another attack, edit any that are currently in the list box and delete any attacks that are in the list box. Clicking the Run Simulation button will start the simulation with the information input by the user. See Appendix E for screen shots from the user interface.
4.1.4 Noise Generation

The generation of noise was modeled independently of the generation of attacks. This was done to reflect how attacks and noise information packets would interact in a real network. The information about the noise is specified during attack scenario setup in the GUI. The user can specify what types of noise they want to include and at what percentage of the total that the different types arrive. The types of noise are based on the categories of attack actions as mentioned in the Literature Review. The user also needs to specify the noise frequency that will be used in the noise creation logic.

The logic for noise creation can be seen in Appendix E. At the start of the simulation, one entity is created. If no noise frequency has been specified, the entity will be disposed and no noise will be generated. If a frequency has been specified, the entity will enter a loop in which it will be delayed for a time sampled from a Poisson distribution with a mean based on the noise frequency. After the delay, the entity will enter a branch block to determine if it is time to stop noise creation. This metric is based on all attacks being complete and the time to run after attack completion specified by the user has been exceeded. If the two conditions for stopping noise generation are met, the entity will be disposed. If noise generation should continue, the entity will be routed to VBA blocks that are used to create the noise alert and write it to a file.

4.1.5 Alert Production

When attack entities are traveling through the main attack flow logic, they are at one point, copied and routed to another set of simulation logic that is used to produce alerts in different files. One set of logic represents alerts that correspond to actual attack events (no noise). In the literature, this is called Ground Truth. Another set of logic represents the creation of alerts by the IDS. The file created by this set of logic is meant to be representative of the files
created by actual intrusion detection systems. As such, they contain alerts corresponding to actual attacks mixed with alerts that are considered noise.

Each set of logic has a VBA block, which executes from code to create the alert. There are currently two blocks in each set, one for the Snort IDS and one for the Dragon IDS. The different systems require their own code because they have different formats. The branch module creates entity copies so that each VBA block is reached. The dispose module removes the entities after their processing is complete.

Each IDS that is setup in the virtual network will have a file associated with it. If an attack is detected by the IDS based on its location then the alert will be produced in its file. Noise alerts are distributed based on the location of the machine. If the machine is external, there is a higher chance that it will experience some noise alerts than an internal machine. Therefore, some files may contain many alerts while others, only a few. A file is created to represent ground truth in a format familiar to the user and others are created to represent the ground truth in alert format for each type of IDS. The files are displayed in the interface at the end of the simulation run so they can be viewed, have their name changed or deleted.

4.1.6 Example Simulation Scenario

An example attack scenario is listed below and the network, files and data corresponding to it are contained in Appendix F. The goal of this attack is to create a denial of service on a machine on the internal network. Information is gathered about the external network, and then the VPN server is penetrated. The server is then used as a stepping-stone to reach the target machine.

The following are the steps of the attack:
1) Enumeration on VPN server from outside of the network attempting to get user passwords. Succeeded on first attempt. Step was encoded.

2) Intrusion at the user level on the VPN server from outside of network. Succeeded on first attempt and attacker gained access to server. Step was encoded.

3) Backdoor left on the VPN server for future access if necessary. Succeeded on first attempt. Step was encoded.

4) Reconnaissance on machine in subnet with snort sensor, specifically ICMP Ping from VPN server. Succeeded on first attempt.

5) Intrusion at the user level on machine 100.10.20.1 from VPN server. Succeeded on first attempt.

Denial of service enacted on 100.10.20.1 from VPN server. Succeeded on first attempt. Step was encoded.

4.2 Formulation of Automatic Attack Generation Methodology

Developing an automatic attack generation methodology was necessary to create multiple attacks quickly and efficiently while ensuring that user bias was minimized. The automatic attack generation methodology is different from the manual attack generation methodology of the original simulation model in that the user need only specify an ultimate goal on an ultimate target machine, and a feasible attack path will be automatically generated. This improves the capabilities of the cyber attack simulation methodology by making the generation of attacks easier on the user. Attacks will be created quickly and with minimal input from the user. In addition, because the attacks are generated randomly, there is no bias from the user involved in the generation of the attacks.
4.2.1 Inputs to the Automatic Attack Generation Methodology

In order to automatically formulate feasible attack tracks to achieve an ultimate goal, determining the actions feasible at different stages was necessary. The methodology uses a graph-based template to determine which groups of actions are feasible at different stages of the attack. This graph-based template for the structure of attacks can then be converted into an adjacency matrix that can be used by the simulation model to evaluate what actions can occur next based on what actions have already occurred. Within each node of the graph based template are a number of attack actions that can be performed.

The simulation model accepts the inputs for the graph-based template adjacency matrix and the actions that are possible in the different stages of the template in text file form. The logic for reading these text files and creating the data matrix in the simulation model can be viewed in Figure 1.

```
; Read in guidance template adjacency matrix from guidance template file and save in array -- START
LineCount = 0
Do while not EOF(6)
   'Loop defines the guidance template
   Line input #6, CurrentLine
   CharCount = 0
   Do while Not Mid(CurrentLine, CharCount + 1, 1) = ""
      Adjacency(LineCount, CharCount) = CInt(Mid(CurrentLine, CharCount + 1, 1))
      CharCount = CharCount + 1
   Loop
   LineCount = LineCount + 1
Loop
Close #6
LineCount = 0
CharCount = 1
Do while not EOF(7)
   'Loop defines the actions within each stage
   If a = "End" Then
      LineCount = LineCount + 1
      CharCount = 1
   Else
      StageActions(LineCount, CharCount, 1) = a
      Input #7, a
      StageActions(LineCount, CharCount, 2) = a
      CharCount = CharCount + 1
      End If
      Input #7, a
Loop
Close #7
; Read in guidance template adjacency matrix from guidance template file and save in array -- END
```

Figure 4 - Automatic Attack Generation Input Setup
A diagram of the graph-based template that the simulation model currently operates under is shown in Figure 2. The graph is a directed graph, which means that an edge only indicates a feasible stage transition in the direction that the edge is pointing. Nodes within the same group form a complete graph $K_n$ where $n$ is the number of nodes in the group. A complete graph is a graph where every node in the graph is connected to every other node in the graph by an edge. This concept is shown in Figure 3.

![Figure 5 - Graph-based Template](image)

![Figure 6 - Group Forming a Complete Graph](image)

An adjacency matrix is an $m \times m$ matrix where $m$ represents the number of nodes in the graph. The entries of an adjacency matrix are 1's and 0's where entry $(i,j)$ is 1 if there is a directed edge from node $i$ to node $j$. Note that $i$ is the row and $j$ is the column. For example, in
the adjacency matrix shown in Table 1, the entry (0,4) is 1 because there is an edge from S0 to S4 in the graph in Figure 1. The entry (4,0) is 0 because there is no directed edge from S4 to S0.

Table 1 - Graph-based Template Adjacency Matrix

<table>
<thead>
<tr>
<th>From Node i</th>
<th>To Node j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The nodes in the graphs in Figures 1 and 2 represent the stages of an attack. Within each stage there are a number of actions that can be performed. The following list describes the types of actions that can occur during the different stages of the simulated attack using the current graph-based template (Sudit, Stotz, & Holender, 2005):

Stage 0: Intrusion-Other, Reconnaissance-Enumeration, Reconnaissance-Footprinting,

Miscellaneous-Other

Stage 1: Intrusion-User, Intrusion-Other, Miscellaneous-Other

Stage 2: Escalation-Service, Escalation-Other

Stage 3: Intrusion-Root

Stage 4: Goal-Denial of Service, Goal-Backdoor, Goal-Pilfering

Stage 5: Intrusion-Other, Reconnaissance-Enumeration, Reconnaissance-Footprinting,

Miscellaneous-Other, Reconnaissance-Scanning

Stage 6: Miscellaneous-Other, Intrusion-Other, Intrusion-User
Stage 7: Escalation-Service, Escalation-Other

Stage 8: Intrusion-Root

Stage 9: Goal-Denial of Service, Goal-Backdoor, Goal-Pilfering

A graphical user interface (GUI) is used to specify many simulation model parameters. The GUI for automatic attack generation can be seen in Figure 4. As new parameters were created for the automatic attack generation methodology, a new GUI needed to be created. The GUI still requires specification of a scenario name, noise parameters and run time after the last attack completes. The difference is in the way that attacks are specified. A user will enter the number of attacks that they wish to generate and for each attack, they will specify the target IP address upon which they want a specified goal to be achieved. A measure of the efficiency of the attacker and the stealth level of the attacker must also be specified as well as the overall delay time for the attack and the average delay between steps. For a complete instruction on the use of the GUI in specifying model parameters for automatic attack generation, reference Appendix G.

To develop feasible attack paths, a new methodology for defining attacks parameters of the simulation model logic had to be created. The idea behind the automatic attack generation methodology is to work backwards, defining the steps of the attack in reverse, until the logic terminates with the initial step of the attack. Two matrices are vital to the automatic attack generation methodology: the machine connection matrix and the graph-based template adjacency matrix. The machine connection matrix is used to identify new targets and attackers in the automatic attack generation logic and the graph-based template adjacency matrix is used to determine the progression of stages of the attack during each distinct exchange between an attacker and a target. The machine connection matrix is generated during the initialization of the
simulation model based on the network template modules that have been placed in the model and the connections between these modules.

**Figure 7 - Automatic Attack Generation GUI**

![Automatic Attack Generation GUI](image-url)
4.2.2 Automatic Attack Generation Logic

The automatic attack generation logic is series of loops that execute until conditions relative to the attacker's movement through the network are met. The logic processes for each attack that the user creates in the GUI when the simulation model run begins. The parameters specified by the user will influence how the attack is generated. If the user specifies a high efficiency score for the attack, the attack will have fewer steps and fewer detour attacks (attacks off of the shortest path from the outside of the network to the target). If the user specifies a low stealth score, more goal type actions will occur such as placement of backdoors, stealing of files or denial of service. The following flowchart in Figure 5 will help explain the automatic attack generation logic.

![Flowchart](image)

**Figure 8 - Automatic Attack Generation Logic**

The first task that must be done is to obtain all of the necessary information of the current attack being generated such as the target, goal, efficiency, stealth, and delay information. If the target specified in the GUI is an internal machine, then the logic above will execute, otherwise it will be skipped. As long as the target remains an internal machine, the logic above will continue to execute. The first action that executes when entering this logic is the choice of an attacker. The pseudo code in Figure 6 will help explain the logic for choosing an attacker.
Go to connection matrix column for Target IP
Any entries with a 1 in that column go into a potential attackers array
Go to connection matrix row for Target IP
Any entries in the potential attackers array with a 1 in the row are lateral moves, denote a 0 in the second row, entry column, of the potential attackers array
Any entries in the potential attackers array with a 0 in the row are upward moves, denote a 1 in the second row, entry column, of the potential attackers array
Sample from Uniform(0,1) Distribution
If Sampled Value <= Efficiency Factor
    Move Up = 1
Else
    Move Up = 0
End If
Sample from set of potential attackers with second row = Move Up
Current Attacker = Sampled Value

Figure 9 - Attacker Choice Logic

The goal of this logic is to determine all of the potential machines that could be the attacker based solely on network topology, and then determine whether the potential attackers are on the same level or a higher level in the network tree structure than the target. This is achieved by evaluating the column and row of the connection matrix corresponding to the current target. If there is a one in the connection matrix column corresponding to the current target then the machine corresponding to the intersecting row is a potential attacker. A one in the connection matrix row corresponding to the current target and the connection matrix column corresponding to the potential attacker indicates that the potential attacker is on the same level of the network tree structure as the target. A zero in the same position indicates that the potential attacker is on a higher level of the network tree structure, indicating an upward (more efficient) move. Noting the level in the network tree structure of the potential attackers is important when choosing an attacker because based on the efficiency specified for the attack, the logic may choose an attacker on the same level or on a higher level. The determination of the attacker from the list of potential attackers is done in two steps. First, sampling a random number from zero to one and comparing the resulting value to the efficiency score will determine if the logic will use an attacker on the same level or a higher level. If the efficiency is greater than the random
number, then the attacker will be from a higher level. Otherwise, the attacker will be from the same level. Secondly, the logic samples a random attacker from the group of attackers chosen (higher or same level). The attacker sampled becomes the attacker for the current series of attack steps.

The next loop that the logic enters after the choice of the attacker is used to continuously attack new targets until the logic determines that the progression of the attack will move to the next higher level in the network tree structure. Within the loop for attack step generation, the attacker will remain constant as the steps of the attack on the current target are generated. Then, a new target will be chosen and the loop condition will be evaluated to determine if a new attacker needs to be generated. The pseudo code in Figure 7 illustrates the process of creating the individual steps of the attack.

```plaintext
If Used Final Goal = False
   Put GUI specified Goal in attack array
   Used Final Goal = True
   Current Stage = 9
Else
   Sample from Uniform(0,1)
   If Sampled Value > Stealth Factor
      Sample a SubGroup from Stage 9
      Put sampled SubGroup in attack array
      Current Stage = 9
   End If
Do while StageComplete = False
   If Current Stage = None
      Sample from Stages 5-8
      Current Stage = Sampled Value
      Sample a subgroup from sampled stage
      Put sampled SubGroup in attack array
   Else
      Go to Current Stage Column of Guidance Template
      Sample from Entries 5 or greater with a 1 to obtain stage
      Current Stage = Sampled Value
      Sample from Uniform(0,1)
      If Sampled Value <= Efficiency Factor
         Stage Complete = True
         Current Stage = None
      End If
End If
Loop
```

Figure 10 - Creating Individual Steps of Attacks
The first logic condition determines if an attack step for the GUI specified goal has been set up. If the GUI specified target is an internal machine, then the first execution of the logic for attack step generation will cause an attack step for the GUI specified goal to be created. Each execution of the logic after the first will skip the attack step setup for the GUI specified goal.

In successive executions of the logic, the first action that is taken is to determine if there will be a goal attack step during the current attacker/target exchange. This is determined by generating a random number between zero and one and comparing the sampled value to the stealth score. If the random number is greater than the stealth score, a goal attack step will be generated. If the random number is less than the stealth score, no goal attack step will be generated and the logic will continue to the next task.

The logic then moves into a loop used to create the remainder of the attack steps for the current attacker/target exchange. Creating the remainder of the attack steps for the current attacker/target exchange is accomplished by evaluating what types of stages from the graph-based template have occurred and determining the possible connected stages from the graph-based template. If there are any one's in the column of the graph-based template corresponding to the stage that the most recent attack step came from, then actions from the stage corresponding to the intersecting row can be the next actions of the attack. After achieving any stage after a goal step on an internal machine, the logic has the ability to proceed to another machine. Therefore, after each attack step is set up, a random number is generated between zero and one and compared to the efficiency score to determine if the attack will move to another target. If the sampled value is greater than the efficiency score, another stage will be sampled and another attack step created from the sampled stage. If the sampled value is less than the efficiency score,
The loop condition will change to false and the logic will move on to choose a new target. The pseudo code in Figure 8 illustrates the logic for new target determination.

```
Sample from Uniform(0,1) Distribution
If Sampled Value <= Efficiency Factor
    Current Target = Current Attacker
Else
    Go to Connection matrix row for Current Attacker
    Any entries with a 1 in that row go into the potential targets array
    Sample from potential targets
    Current Target = Sampled Value
    Clear potential targets array
End If
```

Figure 11 - New Target Determination Logic

There are two possible groups of targets that can be chosen from. The new target can become the previous attacker or the new target can become another machine on the same level in the network tree structure as the current target. Figure 9 below illustrates the choices that can be made for the new target.

![Figure 12 - New Target Choice](image)

In Figure 9, the solid square is the current attacker and the solid circle is the current target. If the target will stay on the same level of the network tree structure, then the lowest dashed circle is a potential target. If the attacker will become the new target, then the dashed circle in the middle will be the new target. If the attacker becomes the new target then the
dashed square is on the level of the network tree structure that the next attacker would come from.

Choosing the previous attacker as the new target will model the way in which attackers penetrate networks by exploiting machines at subsequent levels of the network tree structure in a "leap-frogging" manner. Attackers may also try to exploit many machines in the same subnet, which can be modeled by choosing the next target as a machine from the same level of the network tree structure.

To determine what machine the new target will be, a random number is generated and the sampled value compared to the efficiency factor. If the sampled value is less than the efficiency factor, the new target will become the current attacker. If the sampled value is greater than the efficiency factor, a target on the same level of the network tree structure will be chosen. Choosing a target on the same level of the network tree structure is accomplished by looking at the connection matrix row for the current attacker. Any columns containing a one in the connection matrix row corresponding to the current attacker indicates that the machine corresponding to the column is a possible target. One target is sampled at random from all of the potential targets.

If the new target is another machine on the same level of the network tree structure as the previous target, then the loop condition of Do While Attacker != Target will still be true and the logic for generating attack steps will repeat again. The logic for creating attack steps and choosing new targets will continue to repeat until the current attacker is chosen as the new target during new target determination. When the new target becomes the current attacker, the loop condition Do While Attacker != Target will evaluate to false. When the loop condition Do While Attacker != Target evaluates to false, the target has moved up a level in the network.
tree structure. Therefore when a new attacker is chosen, the new attacker will be a level higher in the network tree structure than the previous attacker. Continuing this looping process will eventually cause the target to move to the highest level of the network tree structure and become an external machine. When the target becomes an external machine, the loop condition \textbf{Do While Target = Internal Machine} will become false and the internal machine logic will finish.

When the internal machine logic completes, the logic for external machines will begin. Any targets specified in the GUI that are external machines will skip automatically to the external machines logic. The second row of the flowchart in Figure 5 illustrates the processing logic for external machines.

The external machine logic executes until the logic creates an attack complete condition, which will cause the loop to exit. The first task that the logic performs is to sample some random numbers to create a random IP address. A random IP address is necessary because when hackers attack a network from the Internet, they use a "spoofed", or disguised IP address, which is random. After generation of a random IP address, the logic for creating the individual attack steps begins. The pseudo code in Figure 10 explains the logic for creating attack steps on external machines.

The logic progression for creating attack steps on external machines is very similar to the logic used to create attack steps on internal machines. The first condition is executed if the GUI specified target is an external machine. If the GUI specified target is an internal machine or the current iteration is not the first time the code has executed, the first condition will not be executed. The difference between the logic for creating attack steps on external machines rather than internal machines is in the termination of the logic.
If Used Final Goal = False
   Put GUI specified Group in attack array
   Current Stage = 4
   Used Final Goal = True
Else
   Sample from Uniform(0,1) Distribution
   If Sampled Value > Stealth Factor
      Sample a SubGroup from Stage 4
      Put sampled SubGroup in attack array
      Current Stage = 4
   End If
Do While StageComplete = False
   If Current Stage = None
      Sample from Stages 1-3
      Sample a subgroup from sampled Stage
      Put sampled SubGroup in attack array
      Current Stage = Sampled Stage
      Intrusion Required = False
   Else
      Go to Column of Current Stage in Guidance Template
      Sample from Entries with a 1
      Current Stage = Sampled Value
      If Current Stage = 1-3
         Intrusion Required = False
      End If
   End If
   If Intrusion Required = False
      Sample from Uniform(0.1)
      If Sampled Value <= Efficiency Factor
         If Current Stage = 1-3
            Sample a SubGroup from Stage 0
            Put sampled SubGroup in attack array
         End If
      End If
      Stage Complete = True
      Current Stage = None
   End If
End If
Loop

Figure 13 - Creating Attack Steps on External Machines

There are two conditions that must be satisfied for the termination of the logic for creating attack steps. First, an action from stages 1-3 must be completed before the logic can terminate. Once an action from stage 1-3 has been completed, the sampling of a random value and comparison to the efficiency factor can occur. If the sampled value is greater than the efficiency factor, then the attack step creation loop will repeat and additional attack steps will be created. If the sampled value is less than the efficiency factor, then the loop for creating attack steps will end and the second condition for attack stage termination will be evaluated. The second condition is that the last action to take place must be an action from stage zero.
Therefore, when the logic determines that the attack stage will finish, if the current stage is not stage zero, then an attack step with an action from stage zero will be generated. Once an action from stage zero has occurred, then the attack step creation logic can end.

To determine if the attack will continue, a random value is sampled between zero and one and compared to the efficiency factor. If the sampled value is less than the efficiency factor, the loop condition will be changed to false and the loop for the external machines will end. If the sampled value is greater than the efficiency factor, a new target will be chosen.

Now that the attacker is attacking from the Internet, the only targets that the attacker can reach are external machines. Therefore, a random external machine is chosen to be the next target. Logic is included to make sure the previous target is not chosen as the new target. After a new target is chosen, the loop repeats with the sampling of a random attacker IP address and the creation of the attack steps for the new target.

Once the loop for external machines has ended, the entire process ends for the current attack. The process repeats for every attack that the user specified in the GUI. When all attacks are complete, the resulting array of important simulation model parameters is complete. However, the array will be in reverse order because the attacks are generated in reverse. Therefore, the last processing step is to reverse the order of the array so that the last item in the array becomes the first, the second to last item becomes the second and so on, until the first item in the original array has become the last item in the new array. At this point, automatic attack generation has completed with all of the important simulation model parameters ordered properly in the parameter array.
4.2.3 Capabilities and Limitations

The automatic attack generation methodology significantly improves the capabilities of the cyber attack simulation methodology developed by Kuhl and Kistner in 2005. The simulation methodology now has the ability to quickly generate attacks in a random but feasible pattern without a great deal of user input. This will help to improve the objectiveness of the simulation methodology. The original methodology was heavily biased on the user who input the data required by the simulation model because the user specified the attack path in the original model. The new methodology is much more robust to user bias because of the random generation of the attack path. In addition, the process of setting up the attacks that the user wishes to simulate is much less input intensive with automatic attack generation. The user does not need to bother with the intermediate steps of the attack, which may be unimportant to many users. Also, for the same input factors the automatic attack generation could develop many different feasible attack paths to reach the same goal on the same target. With the addition of automatic attack generation, a user can automatically create a great deal of the feasible paths that an attacker could take to reach a given target and goal with minimal user input.

However, there are some subsets of the feasible attack paths which are not possible with the current version of automatic attack generation. In order for automatic attack generation to work, the method for calculating the connection matrix had to be adjusted slightly from the original methodology. The changes were necessary in order to simulate the “leap-frogging” maneuvers of attackers as they penetrate a network.

As the calculation of the connection matrix is independent of the attack specification engine, the change to the connection matrix calculation affects both automatic attack generation and the manual attack generation methodology. The change in the calculation method prevents a
user of the manual attack generation methodology from creating attacks in which the target is on a higher level of the network tree structure than the attacker (attacking upwards). This type of activity is uncommon when attackers are initiating their attack from the Internet but it might be necessary for a target to be on a higher level of the network tree structure than the attacker when modeling an inside attack. An inside attacker is an attacker who already has access to the network, like an employee or contractor, who can begin their attack from within the private network. An inside attacker is also known as "insider threat."

The automatic attack generation also does not cover every possible attack path that an attacker can take through a network. Attack paths with detours through more than one level of the network tree structure are not possible with the current version of automatic attack generation. Figure 11 and 12 explain detours and the limitations that detours have in the current version of automatic attack generation.

The progression in Figure 11 is not possible because the attack takes an initial detour (steps one and two) that is deeper than one level of the network tree structure. The progression in Figure 12 is possible because the initial detour (step one) is only one level of the network tree structure deep.

The automatic attack generation methodology is also limited by some parameters of the base simulation model. Specifically, the maximum number of steps that can be created for an attack in the current simulation model is 30. However, specifying a very low efficiency could result in far more than 30 steps. The current automatic attack generation methodology is therefore limited to attacks with a maximum of 30 steps. Stopping the automatic attack generation methodology when 30 steps are reached can result in attacks that have not been generated completely.
4.3 Addition of Further Machine Attributes

The attributes of a machine can limit the forms of actions that can occur on a particular machine. In the simulation methodology, the network topology determines the connectivity attributes of the machines, which allows the user to set up attack actions against a machine.
depending on the source machine of their attack. All actions are currently feasible methods of attacking a given machine but depending on the attributes of the machine, this may not be the case. Attributes such as the type of machine (PC or server) or the operating system of the machine could have an impact on the types of actions that can feasibly be performed. By allowing a user to specify these attributes for machines in the simulation model and by defining the actions in terms of the machines attributes, further accuracy and feasibility can be obtained from generated attacks.

4.3.1 Attribute Definitions

There are hundreds of attributes that could be added to the machines in the simulation model template. However, in order to keep the model building process simple, a few key attributes of a machine have been added to the machine module of the simulation template.

The first attribute is the machine type which indicates the machine’s purpose on the network. Most machines in a network are either a workstation (usually a desktop computer) or a server. Thus, two different machine types created for the simulation model are workstation and server. The hardware used in a workstation machine and a server is very similar but the software is vastly different. This difference is the reason why machine type was included as a machine attribute. The actions that attackers perform against machines on a network are usually highly dependant on the software of the machine rather than the hardware. Therefore, even though the two different machine types can look physically identical at times, they are far different from a hacker’s prospective.

The second attribute defined is the operating system. The operating system is one of the most distinguishing features of a machine. Whether the machine is a workstation or server, each machine has an operating system which manages the hardware and software resources. In
addition, the operating system manages network communications at the machine level. Because the operating system handles so much of the processing and communication of the machine, the operating system is one of the first things that a hacker tries to identify when attempting to attack a machine.

The properties of the operating system also vary from one operating system to another. In addition, hackers find different flaws that can be exploited in different operating systems. For example, certain actions which can affect a machine with a Windows operating system may not be able to affect a machine with a Linux operating system. Therefore, the machine’s operating system limits the types of actions that can be performed against the machine.

4.3.2 Attribute Implementation in Simulation Template

To add the above mentioned machine attributes to the machine module of the simulation template, the template needed to be adjusted. Currently, when a machine module is added to a model, the user may double-click on the module to open the module dialog box and specify information about the machine. Choices for the machine type and the operating system needed to be added to this dialog box.

To add the machine type and operating system choices, two additional operands were added to the existing machine module operands in the operand window. The operands were setup as radio button groups. There are two choices for the machine type: PC and Server. PC represents the workstation and Server represents a server. There are four choices for the operating system: Windows XP, Windows 2000, Linux and Unix. The changes made to the dialog box are shown in Figure 13.
Figure 16 - Machine Module Dialog Box Changes

The logic for the machine module also needed to be updated so that the values for the machine type and operating system operands could be stored in the existing machine variable. The values of the radio button groups are numbers corresponding to the option chosen. The PC option returns a value of 1 and the Server option returns a value of 0. The number for the machine type is assigned to the machine variable array in position seven. For the operating system, the Windows XP option returns a 0, the Windows 2000 option returns a 1, the Linux option returns a 2 and the Unix option returns a 3. The value for the operating system is stored in the machine variable array in position eight. In addition, the appearance of the module when placed in the model window was adjusted so that the information that the model builder specified for the machine would be displayed. The new user display for the machine module is shown in Figure 14.
4.3.3 Action File Definition

In addition to updating the simulation template, the action definitions needed to be adjusted as well. To limit the actions that can occur on a particular machine based on the machine's attributes, the actions needed to be defined in terms of the machine attributes. Thus, the actions input file for the simulation model needed to be adjusted so that actions were specified for certain machine types and operating systems.

In the current actions file, actions are identified by their action group and action subgroup. In order for the actions to be limited by machine type and operating system, the specific machine type and operating system that the action will work for must be specified for each action. If an action can work for both machine types or for multiple operating systems, then the action must be listed for each unique machine type and operating system configuration that is feasible. Thus, some actions may be listed multiple times. The lines in Figure 15 show the definition of an action, "EXPLOIT nlps x86 Solaris overflow," that is feasible for a PC or Server running Windows XP.

```
"Escalation","OS","PC","windows XP","EXPLOIT nlps x86 Solaris overflow"
"Escalation","OS","Server","windows XP","EXPLOIT nlps x86 Solaris overflow"
```

Figure 18 - New Action Definition Example
When the action file definitions are read at the beginning of the simulation run, the machine attributes feasible for the action are stored in additional indexes of the existing actions array. The codes for each action were kept the same to reduce the complexity of the simulation logic upgrade.

4.3.4 Integration

With the improvement to the simulation template, the simulation model can distinguish unique attributes of each machine created. In addition, with the updates to the action input file, the different actions are now dependant on the machine attributes. Using some additional logic, the simulation model can now create attacks in which only actions which are specific to the attributes of the machine being attacked are available.

If the user decides to set up an attack using the manual method, the list of possible actions that they can choose is dependant on the target that they have chosen. Thus, the actions available for the user to choose in the action combo box in the GUI are limited to the actions feasible to the chosen target’s machine attributes. Additional functionality was built in to eliminate the possibility of the user selecting a subgroup of actions which contain no actions feasible to the chosen target. Previously, the user could leave combo boxes blank and the simulation model would choose an action, sub group, or group at random based on the number of combo boxes that were left blank. Logic was added to ensure that the randomly chosen elements of the action were feasible to the chosen target’s attributes as well.

When using the automatic attack generation logic, simple loops were implemented to ensure that the actions chosen were feasible to the target chosen. In the automatic attack generation logic, action groups and subgroups are chosen based on a graph-based template that defines the logical progression of attack actions. Functionality was incorporated to ensure that
the group and subgroup chosen contained feasible actions. If the group and subgroup did not contain feasible actions, another group and subgroup were selected until a feasible set was found. Once the group and subgroup were chosen, actions were chosen at random and evaluated to determine if they met the machine attribute criteria of the chosen target. If the attributes did not match, another action was chosen until a match was found.

4.4 Verification and Validation

Verification and validation is a necessary activity for any simulation model. Ensuring that the model performs as designed and the results of the simulation are true to the system being modeled is essential. A verified and validated model enables the modeler to confidently draw conclusions from the results and take action to improve a system based on those conclusions. However, if the model is not validated, conclusions drawn may not necessarily be correct.

To verify and validate the cyber attack simulation model, several steps needed to be taken. First, the generation of attacks using the manual method needed to be validated. Second, the automatic generation of attacks needed to be investigated to ensure that valid attacks were being generated. Third, production of noise, creation of output files, and network template module input needed to be verified.

A sample network was created to test the cyber attack simulation model. Although this network was small in order to be manageable, the network was setup with common network entities that one would see in a real network. These entities included external servers of varying operating system and subnets of internal machines. Internal servers were setup for one subnet as well. The sample network is shown in Figure 16.
4.4.1 Verification of Manual Attack Generation

Four scenarios were used to test the manual attack generation methodology. The first scenario involved minimal input to test the smallest amount of information required to run the simulation. The second scenario involved a single attack with six steps which represents a common cyber attack. The third scenario had multiple attacks indicative of common cyber attacks and the attacks were setup to overlap one another. The fourth scenario contains 10 attacks, each with 30 steps and was created to verify the input upper limits for the manual attack generation methodology.

Each scenario was setup with the same noise generation parameters. These parameters were the number of alerts to produce per hour, the percentage of each group type that alerts should take and the additional run time after the last attack was complete. The number of alerts per hour was set to 5,000 per hour for each scenario and the percentages for each group were 60% for reconnaissance, 15% for intrusion, 15% for escalation, 5% for goal, and 5% for
miscellaneous. Each scenario was also set to run for an additional five minutes after the last attack was completed.

In scenario one, the total run time was five minutes simulation time. This shows that the additional run time is operating correctly because the attack in scenario one was setup to complete at time zero. In the GUI, the attacker IP was selected as “external” and the resulting target IP choices were all machines with external access, proving that the external functionality works properly.

In the second scenario, the attack setup was consistent with the ground truth of the attack scenario. In addition, the options for targets, both when the source was external and when IP address 100.10.20.5 was used, were valid. Finally, the total simulation run time for scenario two was 90 minutes. This is consistent with the total attack time of 60 minutes and the attack delay of 30 minutes.

The third scenario involved three attacks occurring simultaneously. The output for each attack was accurate to the input specified. In addition, the second of the three attacks completed first because the attack delay and attack duration were shorter than the other two attacks.

The fourth scenario had 10 attacks, each with 30 steps. This scenario was setup to verify the maximum number of attacks that could be created and the maximum number of steps in each attack. The attack ran successfully, proving that the upper limits of attack specification were working properly.

4.4.2 Validation of Automatic Attack Generation Methodology

To validate the automatic attack generation methodology, several scenarios were generated. In each scenario, the attack parameters “efficiency” and “stealth” were varied. The efficiency and stealth parameters were set to be either high (0.9), medium (0.5), or low (0.2). All
possible efficiency and stealth level combinations were examined, resulting in nine different scenarios as shown in Table 2.

**Table 2 - Attack Scenarios for Automatic Attack Generation**

<table>
<thead>
<tr>
<th>Attack Scenarios</th>
<th>Stealth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High</td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>Scenario 1</td>
</tr>
<tr>
<td>Medium</td>
<td>Scenario 4</td>
</tr>
<tr>
<td>Low</td>
<td>Scenario 7</td>
</tr>
</tbody>
</table>

Five attacks were run for each scenario, each with a target on the network level shown in Figure 17. Each attack was delayed by 10 minutes and the average delay time between steps was set to be 10 minutes. Noise was set to occur at a frequency of 5,000 alerts per hour with 60% reconnaissance alerts, 15% intrusion alerts, 15% escalation alerts, 5% goal alerts, and 5% miscellaneous alerts. The simulation was set to run for five minutes after the last attack was complete.
To verify whether the attack progression created was valid, each attack was evaluated on a number of criteria. First, the attack’s progression through the network was validated to ensure that it aligned with the network topology. Second, the progression of the attack at each machine was validated to ensure that the actions taken were feasible. Third, the number of failed attack actions was measured to test the “efficiency” parameter. Finally, the number of additional goals besides the final goals for each attack was measured to test the “stealth” parameter. The inputs set for each scenario compared to the resulting outputs for each scenario are shown in Table 3.

**Table 3 - Inputs Compared to Outputs**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Efficiency</th>
<th>Success Rate</th>
<th>Stealth</th>
<th>Additional Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>0.9</td>
<td>0.91</td>
<td>0.9</td>
<td>2</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>0.9</td>
<td>0.94</td>
<td>0.5</td>
<td>6</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>0.9</td>
<td>0.91</td>
<td>0.2</td>
<td>12</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>0.5</td>
<td>0.44</td>
<td>0.9</td>
<td>21</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>0.5</td>
<td>0.51</td>
<td>0.5</td>
<td>67</td>
</tr>
<tr>
<td>Scenario 6</td>
<td>0.5</td>
<td>0.46</td>
<td>0.2</td>
<td>83</td>
</tr>
<tr>
<td>Scenario 7</td>
<td>0.2</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario 8</td>
<td>0.2</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario 9</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In scenario one, both the efficiency and stealth parameters were set high. This resulted in attacks that quickly arrived at the final target in 5-6 steps. Figure 18 shows the first attack’s progression through the network. The progression of each attack through the network was valid based on the network topology. In addition, each attack had valid attack action progression at each machine. Three failed actions were recorded out of a total of 37, resulting in a success percentage of 91%. This corresponds to the 0.9 value of the efficiency parameter. In addition, there were only two additional goals across the five attacks. This small number of additional goals is because the stealth value was set high.
Scenario two consisted of attacks with high efficiency and medium stealth. These attacks generated similar attack progression but contained six additional goals rather than two. The topology progressions as well as the attack action progression within each machine were both valid. Two failures occurred out of total of 36 actions, resulting in a 94% success rate which corresponds to the high efficiency parameter.

Scenario three had attacks with high efficiency and low stealth. This scenario also exhibited attacks with few detours but there were 12 additional goals, reflective of the low stealth score. The topology progression was valid as was the attack action progression at each machine. A total of four failures were identified out of 44 attack actions resulting in a 91% success rate.

Scenario four exhibited much more erratic attack progression because of the medium efficiency score. Many machines were attacked that were out of the shortest path to the final target. Although the movement through the network was erratic, the progression was still valid based on the network topology. The attack action progression at each machine was much longer with the lower efficiency score than in previous scenarios but the attack action progressions were
still valid as well. A total of 99 failures were recorded out of a total of 179 attack actions. These failures result in a success rate of 44%, corresponding to the efficiency parameter value of 0.5. In this case there were 21 additional goals to the final goals of each attack. The reason that this number is high even though the stealth parameter was set high is because there were many more opportunities for additional goals to occur due to the low efficiency score.

Scenario five also exhibited a number of unnecessary attack steps to reach the final goal because of the lower efficiency score. Figure 19 shows the attack progression of attack three. The topology progression and the attack action progression were both valid for all attacks in this scenario. There were 114 total failures in this scenario out of a total of 236 actions. This results in a success rate of 51% which corresponds to the 0.5 efficiency value. A total of 67 additional goals were identified which reflects the decreasing stealth value from the previous scenario.

![Attack Progression](image)

**Figure 22 – Scenario 5, Attack 3 Progression**

Scenario six also exhibited erratic attacks because of the lower efficiency score. The topology progression of the attacks was valid as was the attack action progression within each machine. This scenario had 132 failures out of 248 actions resulting in a success rate of 46%.
There were a total of 83 additional goals in this scenario which corresponds to the low value for the stealth parameter.

The last three scenarios had low efficiency scores which caused extremely erratic attacks to be generated. Although these types of attacks are technically feasible, the current simulation methodology can only support attacks with 30 steps. When a very low efficiency is specified, attacks generated quickly reach the maximum number of steps in the attack before they are completed. Thus, these attacks are not very useful for analysis. The maximum number of steps in an attack needs to be increased for low efficiency attacks to be generated.

4.4.3 Noise Production, Template, and File Creation Verification

Several things needed to be verified in addition to the attack generation methodologies. Specifically, within the attack generation methodologies, the production of noise needed to be verified for accuracy. In addition, the variables and stations created from the network simulation template modules needed to be verified. Finally, the creation of some output files is dependant on choices for sensors on network simulation template modules and the creation of these output files needed to be verified as well.

To verify the production of noise, scenarios were run in which minimal input is required. In these situations, practically all of the output is due to noise and these alerts can be counted, classified, and evaluated. There are three things that needed to be verified for the production of noise. First, the amount of noise that occurred in each scenario needed to be verified against the number of noise alerts per hour that was setup for each scenario. Second, the percentages of each type of alert that can be used for noise needed to be verified. Finally, the percentage of noise that occurs on internal machines and external machines needed to be verified as well.
Through several scenarios generated by the automatic attack generation methodology, the amount of noise present was evaluated. Each scenario consisted of a single attack and the attack target and goal was the same for each scenario. In addition, the initial delay time and the average delay time between steps was set to 10 minutes. Table 4 shows the inputs compared to the outputs for the noise creation validation.

Table 4 - Inputs Compared to Outputs for Noise Generation

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Input Alerts/Hour</th>
<th>Simulation Time</th>
<th>Total Alerts</th>
<th>Output Alerts/Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>58 mins</td>
<td>64</td>
<td>62</td>
</tr>
<tr>
<td>2</td>
<td>800</td>
<td>71 mins</td>
<td>1,127</td>
<td>955</td>
</tr>
<tr>
<td>3</td>
<td>5,000</td>
<td>64 mins</td>
<td>4,674</td>
<td>4,409</td>
</tr>
<tr>
<td>4</td>
<td>30,000</td>
<td>87 mins</td>
<td>37,491</td>
<td>26,589</td>
</tr>
</tbody>
</table>

In the first scenario, the number of alerts per hour was set to 60. Running this simulation took 58 total minutes simulation time. The total number of alerts generated in this scenario was 64. With 58 minutes and 64 alerts, the number of alerts per hour generated is 62.

In the second scenario, the number of alerts per hour was set to 800. The total simulation run time was 71 minutes and the number of alerts generated during this time was 1,127. This results in 955 alerts per hour which corresponds to the 800 alerts per hour set up.

The third scenario had the number of alerts per hour set to 5,000. The simulation run time was 64 minutes and the total number of alerts generated was 4,674. The resulting number of alerts per hour for the simulation run was 4,409.

The number of alerts per hour in the third scenario was set to 30,000 to reflect the frequency of alerts that a large company might see. The time for the run was 87 minutes in simulation time and the total number of alerts generated was 37,491. The resulting alerts per hour for the simulation run were 26,589 which correspond to the 30,000 alerts per hour that was setup for the scenario.
A scenario was created to evaluate the types of alerts generated based on the percentages specified. The percentages used for testing were: 60% reconnaissance, 15% intrusion, 15% privilege escalation, 5% goal, and 5% miscellaneous. The scenario was run for a total of five minutes at 5,000 alerts per hour, generating a total of 107 alerts. Of the 107 alerts, 44 were reconnaissance alerts, 31 were intrusion alerts, 13 were privilege escalation alerts, 16 were goal alerts and 4 were miscellaneous. Although not exactly correct because the alert distributions are random rather than exact, the trends show that the percentages seem to be working properly. Using more alerts would only further verify that the percentages are functioning correctly.

The example network used for validating the manual and automatic attack generation methodologies was used to test the creation of variables and stations from the network simulation template modules as well. The example resulted in 20 machine variable arrays created. Each of the machine variables reflects the information input in the modules. The modules also create 20 stations, one for each machine. To see the complete data that shows the variables and stations, refer to Appendix H.

Lastly, the creation of output files needed to be verified. When the user creates a sensor on either a connector module or a machine module, a corresponding alert file should be created. Several different network detection configurations were used to test the creation of these files. Two scenarios were simple scenarios that one might see in a regular network. Another scenario has no connector sensors and one scenario has no machine sensors. The last scenario has a sensor at every possible place in the network. All files which were intended to be created existed at the end of the simulation runs for each scenario. To see each of the scenarios and the resulting files created, refer to Appendix H.
4.4.4 Summary

Verification and validation is very important because it allows a modeler to draw conclusions from a simulation with confidence that the simulation reflects the real system to the degree necessary for analysis. To verify the manual attack generation methodology, several examples were generated to make sure that the input for attacks matched the output and that valid targets were presented for different choices of attackers. In addition, the minimal amount of input was used as well as the maximum amount of input to test the methodology at the input bounds.

Validation of the automatic attack generation methodology was also necessary. Several more examples were created to do the validation, each with varying degrees of attack parameter input. Each of these examples was validated for correct network topology progression as well as attack action progression within each attacked machine. In addition, the success rate was validated as well as the creation of additional goals besides each attack’s final goal.

Noise creation was validated as well to ensure that the proper amount of noise alerts and the correct percentage of alert types were being generated. Several scenarios were tested to examine this with all examples showing that noise creation was working properly. The creation of the machine variables, stations, and output files from template modules were examined with a number of examples as well and found to be valid.

With these validation techniques performed, the model has been properly validated. As a result, an analyst may comfortably draw conclusions from the results of a simulation run. In addition, the data from simulations can be used to test other systems.
5. Investigation of Process Refinement in the Cyber Domain

Cyber security is a growing problem and as a result, systems are being developed to assist systems administrators detect and track cyber attacks. Specifically, information fusion systems have been developed to correlate IDS alerts into meaningful attack situations. From these attack situations, other parts of the fusion system can measure the impact of the attack situations and estimate the threat that the attack scenario might pose in the future.

Process Refinement acts after the threat projection to refine and improve the fusion system to improve the capabilities of the system in detecting, tracking, and interpreting the attack scenarios. This section introduces the topic of Process Refinement and explains how Process Refinement is unique compared to the other levels of information fusion. Process Refinement’s application to cyber security and Process Refinements role in a cyber security fusion system will also be discussed. Finally, an initial integer programming optimization model for Process Refinement is presented and examined.

5.1 Process Refinement Conceptual Model

The concept of Process Refinement is to optimize the data acquisition and interpretation functions of an information fusion system. This is achieved by examining the output of the various levels of fusion and identifying areas where information should be obtained, where information is lacking, where the integrity of the information is less than desirable and many other areas. However, there may be many different types of improvements that could be implemented, many of which are mutually exclusive in the analysis required. For example, suggestions pertaining to what sensor information to process are independent from recommendations pertaining to correcting sensor errors. In addition to using information from other levels of information fusion, the improvement methods may use input from or provide
output to other improvement methods. From the example earlier, the suggestion pertaining to sensor error correction may feed into the suggestion pertaining to which sensor information to process so that information from the malfunctioning sensor is not obtained until the sensor errors are corrected. Thus, a system of Modular Process Refinement is suggested in which each module of Process Refinement handles a different type of improvement. The modules will interact with other levels of information fusion and with each other. Figure 20 illustrates the concept of Modular Process Refinement and how Modular Process Refinement fits into the fusion realm. Some improvement methods might include changes to guidance templates, adjustments to sensors, using alternative correlation methods or a number of other improvements. The focus of this thesis will be the dynamic management of sensing resource information provided to the fusion system and from this point further, the term Process Refinement will pertain to the dynamic management of sensing resource information.

Figure 23 - Modular Process Refinement
5.2 Sensor Management and Process Refinement

In the realm of information fusion, there are two terms that are often used interchangeably to describe the fourth level of information fusion: Sensor Management and Process Refinement. It is important to point out the distinction between the two in the cyber domain because they describe very different tasks. The task of Sensor Management involves determining the placement of sensors in a network, often using an optimization technique associated with the cost of placing a sensor and the benefit that the sensor will bring. Process Refinement is a more dynamic type of analysis in that as new information is obtained from other levels of fusion, Process Refinement adapts to the new information, making iterative recommendations for improvement.

Many network security advisors recommend placing both host-based and network-based sensors. The placement of the two types of sensors depends on the topology of the network and the criticality of the machines in the network. Network-based sensors could be placed before or after external firewalls, on major network junctions or on critical subnets. Placing the sensors in these locations allows a network analyst to get a very broad view of the network situation, enabling better detection of attacks. Optimization can be used to determine where the sensors will be located based on a number of constraints, costs and values of the different location points in the network.

Sensor management has many benefits in other settings but falls slightly short in usefulness in the cyber domain. The problem with sensor management in the cyber domain is that the costs of placing sensors in the cyber domain are often very small. Many network and host-based sensors are open-source programs, meaning that they can be downloaded for minimal to no cost. In the case of host-based sensors, if the software does not cost anything, then the
entire sensor system does not cost anything because the hardware requirements are met by the machine on which the sensor will be placed. Network sensors do require hardware to run the sensor software but with decreasing costs of computers, the cost requirement of hardware becomes very low.

As a result, many sensor management optimization models may result in an optimal solution of sensor placement with a sensor in nearly every location on the network. The problem with sensor placement at every network location is that the majority of systems which process sensor data would become overloaded in trying to process data from so many sources. Therefore, it is ideal to have a sensor at every location in the network from a detection standpoint because every possible attack could be detected but it is not feasible to obtain the data from all sensors at once. However, the majority of the time, data from all sensors is not needed because attackers are usually not attacking every network location at once. Thus, at different points in time, there are different attack situations occurring that require different resources for detection. As a result, using an optimization model to determine which sensors to obtain data from based on a given situation is necessary.

### 5.3 Process Refinement Mathematical Model

By defining Process Refinement in terms of a mathematical optimization model, optimal solutions for the management of network detection resources can be determined. Specifically, Process Refinement is concerned with maximizing the value of information that is obtained. In Process Refinement, value refers to the system administrator’s perceived value of the information on a machine. When Process Refinement maximizes the value of the information that is obtained, the system administrator can be assured that given a situation in which all data
can not be obtained, data from the machines which the administrator considers most valuable will be obtained.

There are three parts to every mathematical optimization model: decision variables, objective function, and constraints. The decision variables are the factors that the mathematical model is trying to optimize. The objective function is a function of the decision variables and scalars such as cost, distance, value, etc., such that changing the value of a decision variable changes the value of the objective function. The assignment of the values of the decision variables is motivated towards the objective definition of minimize or maximize. Constraints are functions of the decision variables and scalars that limit the values of the decision variables that can be chosen, thereby limiting the value that the objective function can obtain.

The decision variables in Process Refinement are based on three dimensions: the type of information, the information source, and the machine on which the source is located. Types of information include alert messages, sensor preprocessor alerts, logs, and many other types. Sources of information are any type of sensing resource such as a network-based IDS sensor, host-based IDS sensor, System Log output and many others. As a result of the decision to place a sensor everywhere in the network, each source can be obtained from each different machine. It is understood that some types of information are not feasible to some information sources. This problem is handled by specifying sets of information that are feasible to each sensor.

In Process Refinement, the objective is to maximize the total value of the information obtained based on the current situation. Using the constraints, the optimization model effectively eliminates the selection of machines which can not possibly be targeted next based on the current situation. The resulting reduction in the information to gather from a situation in which all data is collected is described in an example in Figure 21. In the diagram, there are three network-
based IDS, one for each switch and nine machines each of which could have \( x \) information sources. Thus the number of sources which would need to be processed in a situation where every source is processed is \( 9x + 3 \). The machine at the top of the tree structure is the only compromised machine based on the current situations so the two machines at the next level down in the tree structure are threatened but the remaining machines are not. This would result in a total of \( 3x + 1 \) total sources to be processed if Process Refinement were used because the machines at the third level of the tree structure are not threatened and therefore, can provide no useful information about the next attack steps based on the current situation.

![Diagram of information reduction](image)

**Figure 24 - Information Reduction**

The following descriptions define the decision variables and scalars of the Process Refinement mathematical model and the model is shown in Figure 22. In the following statements, the term "information" pertains to the different types of information that a sensor can report such as alerts, preprocessor alerts, etc. The source of information pertains to the sensor or
system log from which the information is coming. The machine on which a sensor source is located is the machine in the network that contains the sensor.

Decision Variables:

\[ x_{ijk} = \begin{cases} 1 & \text{if information } i \text{ is obtained from source } j \text{ on machine } k \\ 0 & \text{otherwise} \end{cases} \]

\[ i = \text{Information (Alerts, Preprocessor Information, etc.)} \]
\[ j = \text{Source of Information (Sensor, Log, etc.)} \]
\[ k = \text{Machine on Which Source is Located} \]

Scalars:

\[ Z = \text{the objective function value} \]
\[ v_k = \text{value of information contained on machine } k \]
\[ c_{ijk} = \text{bandwidth requirement for extracting information } i \text{ from source } j \text{ on machine } k \]
\[ B = \text{bandwidth limit} \]
\[ p_{ijk} = \text{processing requirements for information } i \text{ from source } j \text{ on machine } k \]
\[ L = \text{maximum processing power available to the fusion engine} \]
\[ t_k = \text{threat level (0 to 1) from fusion Level 3: Threat Assessment for machine } k \]
\[ D_{ij} = \text{ability of information } i \text{ from source } j \text{ to detect action } l \text{ (0 or 1)} \]
\[ S_{4k} = \text{set of all actions that can be performed next on machine } k \text{ (from set } S_{4} \text{ of fusion Level 3: Threat Assessment)} \]
\[ F_j = \text{set of all information } (i) \text{ feasible to source } j \]
\[ \varepsilon = \text{very small number} \]
The objective of the optimization model is to maximize the total value obtained by selecting types of information from different information sources on different machines in the network. Maximizing the sum of the products of the values and the decision variables for all dimensions of the decision variables which are feasible, as shown in equation (1), will maximize the total value of the information selected.

The first constraint, equation (2), limits the information that can be obtained based on the bandwidth of the network connection between the sensor data collection sites and the information fusion processing engine. The second constraint, equation (3), limits the information that can be obtained based on the processing power of the information fusion processing engine. If information files are large, then less information can be obtained. The third constraint, equation (4), limits the selection of information from machines with no threat. The fourth constraint, equation (5), limits the information selected so that only information types and sources that can provide information about the next possible actions on each machine are selected.
5.4 Model Implementation

To show that the mathematical model works as intended, the model was built using ILOG OPL Studio in the Optimization Programming Language (OPL). The benefit of using OPL Studio is that the model can be built generically and data files can be used to adjust the inputs to the model from situation to situation. Figures 23 and 24 show the model that was built in OPL Studio.

```plaintext
enum Information ...;
enum Sources ...;
enum Machines ...;
enum Actions ...;

// F(j) or the information feasible to source j
{Information} Feasible[Sources] = ...;

// S4(k) or the set of all actions that can be performed next on machine k
{Actions} S4[Machines] = ...;

// Value of each machine based on implied situation independent by the user
float+ Value[Machines] = ...;

// Bandwidth required to obtain information i from source j on machine k
float+ BandwidthCost[Information, Sources, Machines] = ...;

// The maximum bandwidth available for transferring
float+ BandwidthMax = ...;

// The amount of processing power required to extract information i from
// source j on machine k
float+ ProcessingCost[Information, Sources, Machines] = ...;

// The overall maximum amount of processing power available to the fusion
// system
float+ ProcessingMax = ...;

// The current threat value of machine k
float+ Threat[Machines] = ...;

// A very, very, very small number
float+ Epsilon = ...;

// The ability of information i from source j to detect action 1
float+ Detect[Information, Sources, Actions] = ...;

// The variable x(i,j,k) which is a binary decision variable
// The variable = 1 if information i will be obtained from source
// j on machine k
var int+ x[i in Information, j in Sources, k in Machines] in 0..1;
```

Figure 26 - OPL Model Declarations
Objective of process refinement is to maximize the value of the information obtained:

\[
\text{maximize } \sum_{k \in \text{Machines}, j \in \text{Sources}, i \in \text{Feasible}[j]} \text{value}[k] \times x[i,j,k]
\]

subject to:

- Can't exceed the maximum bandwidth:
  \[
  \sum_{k \in \text{Sharing}[h], j \in \text{Sources}} \text{BandwidthCost}[i,j,k] \times x[i,j,k] \leq \text{BandwidthMax};
  \]

- Can't exceed the overall processing power available to Fusion:
  \[
  \sum_{i \in \text{Information}, j \in \text{Sources}, k \in \text{Machines}} \text{ProcessingCost}[i,j,k] \times x[i,j,k] \leq \text{ProcessingMax};
  \]

- Do not take information from machines with no threat:
  \[
  \forall (i \in \text{Information}, j \in \text{Sources}, k \in \text{Machines}) \quad x[i,j,k] \leq \text{Threat}[k] + 1 - \text{Epsilon};
  \]

- Don't take info that will not help detect next possible actions:
  \[
  \forall (i \in \text{Information}, j \in \text{Sources}, k \in \text{Machines}) \quad x[i,j,k] \leq \sum_{l \in S4[k]} \text{Detect}[i,j,l];
  \]

\[\]

Figure 27 - OPL Mathematical Model

This model accepts data for constant model parameters such as bandwidth and processing costs as well as variable inputs like the value parameters and the $S4_k$ subsets. An example was created consisting of a small sample network and a series of situations which could be modeled as input to the Process Refinement model. The sample network is shown in Figure 25.

In this network, there are three external machines: Web Server, FTP Server and Email Server. On the internal network, there are file and database servers as well as administrative and client machines. This provides a wide variety of types of machines that an attacker could target. In addition, the spectrum of machine importance is also very broad. Table 5 defines the situation independent value of each machine (scale of 1-10).
Figure 28 - Sample Network

Table 5 – Machine Values ($v_i$)

<table>
<thead>
<tr>
<th></th>
<th>Web Server</th>
<th>FTP Server</th>
<th>Email Server</th>
<th>Web Admin</th>
<th>Subnet File Server</th>
<th>Internal Database</th>
<th>Dbase Admin 1</th>
<th>Dbase Admin 2</th>
<th>Client 1</th>
<th>Client 2</th>
<th>Client 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value ($v$)</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.6</td>
<td>0.9</td>
<td>0.6</td>
<td>0.6</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>
As each new situation arises, the values can change based on the calculated threat values. These threat values would be obtained from Level 3: Threat Assessment of the information fusion process but for purposes of this example, the threat value is an assumed value and not directly calculated. The resulting value at each situation can be a combination of the threat level of the machine and the situation independent value.

Although all of the model data can be variable, some model data is constant from situation to situation for a given example. This data includes the bandwidth cost matrix, the bandwidth maximum, the processing cost matrix, the processing cost maximum, epsilon and the detection matrix. The situation constant data as input into the model for the example is shown in Table 6 and Figure 26.

```
// Enumerations
Information = { Recon, Int, Esc, Goal };
Sources = { Snort, SysLog };
Machines = { WebServer, FTPServer, EmailServer, WebAdmin,
            SubnetFileServer, InternalDatabase, DbaseAdmin1,
            DbaseAdmin2, Client1, Client2, Client3 };
Actions = { ReconFPrint, ReconEnum, ReconSniff, ReconScan, IntrusionRoot,
            IntrusionUser, PESCOS, PESCService, GoalPilf, GoalBD, GoalDos };
Feasible = [{ Recon, Int, Goal }, { Recon, Int, Esc, Goal }];
```

**Figure 29 - Situation Constant Data**

To test the Process Refinement system, a typical cyber attack was developed and the parameters of Process Refinement at each stage were estimated. The attack began with reconnaissance, an intrusion, and ultimately a goal of denial of service on the Web Server. After attacking the Web Server, the attack moves to the Internal Database with a reconnaissance action, followed by a privilege escalation.
### Table 6 - Situation Constant Data

<table>
<thead>
<tr>
<th></th>
<th>Recon</th>
<th>Intrusion</th>
<th>Escalation</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Snort</td>
<td>SysLog</td>
<td>Snort</td>
<td>SysLog</td>
</tr>
<tr>
<td>WebServer</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FTPServer</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>EmailServer</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DBaseAdmin1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DBaseAdmin2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Client1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Client2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Client3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>FTPServer</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>EmailServer</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DBaseAdmin1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DBaseAdmin2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Client1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Client2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Client3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Detectable Actions

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ReconFPrint</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ReconEnum</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ReconSniff</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ReconScan</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IntrusionRoot</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IntrusionUser</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PEscOS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PEscService</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GoalPllf</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GoalBD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GoalDoS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BandwidthMax = 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processing Max = 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Epsilon = 0.000000000001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The initial run of the test system was based on a situation in which no problems were occurring. In this situation, the value of all machines was zero because the threat to all machines was zero. The $S_d$ subsets were all empty because there were no actions against the network.
taking place. The case of zero values was a trivial case but necessary to ensure that the test system was working correctly. As suspected, the decision variables indicated that no information should be obtained from any source because there is no situation. The zero values case proved that the model performed properly under trivial conditions. Aside from testing, the system should be set up such that if no situation is occurring, machines on the external network have a threat value greater than zero. By making machines on the external network threatened, the system will be able to detect attacks when they begin.

The first situation run performed with meaningful data was designed to resemble an initial reconnaissance attack that an attacker might do to gain further information about the network. The data in Table 7 was used for input for the situation variable inputs of the test model.

It can be seen from this data that the values and threat for all internal machines remains zero at this point because an intrusion has not yet occurred on the external network. The \( S4_k \) subset defines the next actions that could potentially occur. The resulting resource allocation obtained from the test model using the data above is shown in Table 8.

The model obtained all of the resource allocation that it could, based on the constraints present. From the information in the \( S4_k \) subset, the model determined that the only actions that can occur next on the FTP Server and Email Server are reconnaissance attacks. As such, the model obtained resources on these machines pertaining to reconnaissance only. The model obtained many information sources from the Web Server. Information from Snort pertaining to Escalation types of information was not obtained because Escalation information is not in the \( F_{Snort} \) subset defined in the constant values section of the data input. Reconnaissance information
from SysLog was not obtained because SysLog’s reconnaissance type information is not able to
detect any of the actions from the $S\text{4}_{\text{WebServer}}$ subset.

Table 7 – Model Input after First Reconnaissance

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Threat</th>
<th>$S\text{4}_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalP1f, GoalDoS</td>
</tr>
<tr>
<td>FTPServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>EmailServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>0.4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>0.6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>0.9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>0.6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>0.6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 1</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 8 – Model Recommendation after First Reconnaissance

<table>
<thead>
<tr>
<th></th>
<th>Snort</th>
<th>SysLog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recon</td>
<td>Int</td>
</tr>
<tr>
<td>WebServer</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FTPServer</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EmailServer</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>WebAdmin</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InternalDatabase</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The next attack step tested was an initial intrusion on an external machine. The Web Server was attacked once again, this time with an intrusion exploit. The data in Table 9 was used for input to the test model.

### Table 9 – Model Input after Intrusion

<table>
<thead>
<tr>
<th>WebServer</th>
<th>Value</th>
<th>Threat</th>
<th>S4k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalPflf, GoalDoS</td>
</tr>
<tr>
<td>FTPServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>EmailServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>0.4</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>0.9</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>Client 1</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

When intrusions occur on the external network, internal network resources now become vulnerable to attack. As a result, there are a number of machines which are now threatened and have value and threat scores. However, at this stage, only reconnaissance attacks are expected on the internal machines as is reflected in the $S4_k$ subsets. The results from the test model run are displayed in Table 10.
As suggested above, all internal machines one level below the external machines are now threatened by reconnaissance. As such, the model has chosen to monitor reconnaissance information on all of the machines that are threatened. As the value of the Web Server remained the highest value of all machines, all of the information obtained for the Web Server during the previous situation was obtained again.

Following a typical attack progression, the next situation to occur was a goal of denial of service on the Web Server. The data in Table 11 was used for input for the denial of service situation. The main point of interest with this set of data is that the actions in the S4_k subsets have not changed for this situation. The actions in the subsets have not changed because achieving a goal does not necessarily provide the attacker any additional knowledge to use in penetrating the network further. Usually, goal actions occur at the end of an attack progression when the attacker is finished penetrating the network. Other possible goals actions besides denial of service include file pilfering and installing backdoors. The results from the denial of service scenario are shown in Table 12.
Table 11 – Model Input after Denial of Service

<table>
<thead>
<tr>
<th>Service</th>
<th>Value</th>
<th>Threat</th>
<th>S4k</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalPilf, GoalDoS</td>
</tr>
<tr>
<td>FTPServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>EmailServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>0.4</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>0.9</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>Client 1</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 12 – Model Recommendation after Denial of Service

<table>
<thead>
<tr>
<th>Service</th>
<th>Snort</th>
<th>SysLog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recon</td>
<td>Int</td>
</tr>
<tr>
<td>WebServer</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FTPServer</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>EmailServer</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>WebAdmin</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Client 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since the attacker gained no additional knowledge for network penetration in performing a denial of service, the S4_k subset values did not change. As a result, the system did not obtain information from any additional sources than the previous situation. Had the S4_k subsets changed, then additional data may have been obtained. If the S4_k subsets do not change, different information can only be obtained by significant changes in value of the machines. As
the web server has been continually attacked and no new machines threatened, the recommended information to obtain does not change.

Now that the attacker has compromised a machine completely on the external network, they may decide to continue attacking machines on the external network or try to penetrate the internal network. As most external networks do not contain the types of information that hackers often attack networks to obtain, a penetration of the internal network is more likely. As such, hackers may attempt to perform reconnaissance on an internal machine or subnet to try to learn more about the information contained on the internal network and the security which protects the network resources containing the information. The data shown in Table 13 was used for input for a reconnaissance attack on the internal database.

Table 13 – Model Input after Second Reconnaissance

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Threat</th>
<th>S4k</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEScService, IntrusionRoot, GoalBD, GoalPlif, GoalDoS</td>
</tr>
<tr>
<td>FTPServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>EmailServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>0.4</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>0.9</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEScService, IntrusionRoot, GoalBD, GoalPlif, GoalDoS</td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEScService, IntrusionRoot, GoalBD, GoalPlif, GoalDoS</td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEScService, IntrusionRoot, GoalBD, GoalPlif, GoalDoS</td>
</tr>
<tr>
<td>Client 1</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td>0.3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Once a reconnaissance attack has occurred on a subnet in the internal network, machines connected to the machine that has been attacked are threatened. Thus, when the reconnaissance
attack occurred on the internal database, Dbase Admin 1 and Dbase Admin 2 are also threatened. This causes the list of actions in the $S_{4_{DbaseAdmin1}}$ and $S_{4_{DbaseAdmin2}}$ subsets to expand into all possible actions because they are on the same level as the internal database. The resulting recommendation for this situation is shown in Table 14.

Table 14 – Model Recommendation after Second Reconnaissance

<table>
<thead>
<tr>
<th></th>
<th>Snort</th>
<th></th>
<th></th>
<th>SysLog</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recon</td>
<td>Int</td>
<td>Esc</td>
<td>Goal</td>
<td>Recon</td>
<td>Int</td>
<td>Esc</td>
</tr>
<tr>
<td>WebServer</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTPServer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmailServer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebAdmin</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At this point, the recommendations are vastly different between the situation when the internal network was not penetrated and the situation when the internal network has been penetrated. At this point, the bandwidth and processing constraints begin limiting the number of network resources that information can be obtained from. The values for many of the internal machines are intrinsically higher than the external machines so now that the internal machines are compromised, the overall values of the internal machines becomes higher than most of the external machines. As a result, the network resources which information will be obtained from will lean more towards those resources that can obtain information about the internal machines than external.

Now that the attacker has gained some information about the internal database, the attacker would more than likely try to exploit this machine with the information that they have
obtained. The data used to create a privilege escalation attack on the internal database is shown in Table 15.

Table 15 – Model Input after Privilege Escalation

<table>
<thead>
<tr>
<th>Model Input</th>
<th>Value</th>
<th>Threat</th>
<th>GoalBD, GoalPilf, GoalDoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalPilf, GoalDoS</td>
</tr>
<tr>
<td>FTPServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>EmailServer</td>
<td>0.5</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>WebAdmin</td>
<td>0.4</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>0.9</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalPilf, GoalDoS</td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalPilf, GoalDoS</td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>0.6</td>
<td>1</td>
<td>ReconFPrint, ReconEnum, IntrusionUser, PEscService, IntrusionRoot, GoalBD, GoalPilf, GoalDoS</td>
</tr>
<tr>
<td>Client 1</td>
<td>0.3</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>Client 2</td>
<td>0.3</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
<tr>
<td>Client 3</td>
<td>0.3</td>
<td>1</td>
<td>ReconFPrint, ReconEnum</td>
</tr>
</tbody>
</table>

The main difference in the data input for the reconnaissance attack and the data input for the privilege escalation is that the values of the machines have changed. The web server has not been attacked recently and the attacker appears to be attacking the internal network so there is less need for network resources on the external network. The result is that the internal database and the two machines connected to it have the highest value of any machines in the network. The data in Table 16 shows the results from the privilege escalation situation.

The maximum amount of detection resources have been obtained to detect attacks on the internal database, Dbase Admin 1 and Dbase Admin 2 because these resources have the highest value. The remaining capacity for processing and bandwidth is allocated to Web Admin, Subnet
File Server and the Web Server because they all have the same value. At this point, the shift in detection focus from external to internal has been reached.

Table 16 – Model Recommendation after Privilege Escalation

<table>
<thead>
<tr>
<th></th>
<th>Snort</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recon</td>
<td>Int</td>
<td>Esc</td>
<td>Goal</td>
<td>Recon</td>
<td>Int</td>
<td>Esc</td>
<td>Goal</td>
<td>Recon</td>
<td>Int</td>
<td>Esc</td>
<td>Goal</td>
<td>Recon</td>
<td>Int</td>
</tr>
<tr>
<td>WebServer</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTPServer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmailServer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebAdmin</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SubnetFileServer</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InternalDatabase</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DbaseAdmin1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>DbaseAdmin2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As the attacker penetrates deeper into the network, the system will continue to project their activities forward, allowing the system administrator to identify their progress until the attacker’s access to the network can be eliminated. Projecting the attacker’s actions forward allows the system administrator to easily track intruders by using the resources that provide maximum value based on the current situation. As a result, the system administrator can spend more time trying to remove the attacker from the network than spending valuable time trying to keep up with the attacker as the attacker progresses through the network.
6. Evaluation of Process Refinement Performance

By using an example network and attack scenario, Process Refinement has demonstrated the ability to track attacks as they propagate through a network by taking advantage of the most valuable of detection resources to the current situation. However, the question remains of how Process Refinement performs against other methods of detection. The power of Process Refinement is the ability to alter detection points based on the current situation rather than obtaining data in a static manner. Thus, tests were conducted to evaluate Process Refinement against a series of static scenarios in order to measure and evaluate the benefits of using the Process Refinement system.

In order for the analysis to be unbiased, the simulation model presented in section four was used to create attack scenarios with accompanying noise. The automatic attack generation methodology was used to generate the attacks to remove any bias caused by user input. A series of random static detection scenarios were generated to test against the Process Refinement system.

6.1 Test System Setup

The purpose of testing Process Refinement against static scenarios was to show the benefit of Process Refinement’s ability to adjust network detection to changing situations in a network. Therefore, for the purposes of this test, the types of information gathered were not as important as where and when they were gathered from. In order to keep the test simple yet still able to show the benefits of Process Refinement, one type of information was gathered from two different information sources. The type of information was alerts and the two sources shall be named sensor one and sensor two.
6.1.1 Network Topology and Information Definition

To further show the benefit of Process Refinement, the detection abilities of sensor one and sensor two were defined differently. Sensor one was defined as a reconnaissance detecting sensor and sensor two was defined as an intrusion and goal detecting sensor. This was done to illustrate Process Refinement's ability to not only base its recommendation on which machines could be threatened next but also on which actions could happen next as well. Thus, in this particular example, if only reconnaissance style actions can happen next based on the current situation, process refinement will only select sensor one.

The network that was created to use as a test system was designed to have many subnets of varying size which attackers to penetrate. The network was also designed to have multiple avenues into these subnets. The test network contains 33 total machines, some servers and some workstations, and a clearly defined internal and external network. A diagram of the network created is shown in Figure 27.

To determine the total amount of information that could be collected in the example network, estimates were obtained from subject matter experts as to how much information could be obtained on a network if network sensors were located at each node of the network. An estimate of 2,000 alerts arriving per second from all network detection resources was used for this test. Secondly, subject matter experts also supplied estimates for the number of alerts that could be processed by a fusion engine. The estimate used for this parameter was 1,000 alerts per second. Thus, although the network would have 2,000 alerts arriving per second if all information was gathered, the fusion engine can only process half of them.
Although the network used in this test system is far smaller than the networks obtaining 2,000 alerts per second, a similar proportion for the amount of information that can be gathered was used. The total amount of information that can be gathered in the example network was 66 pieces of information, two sensors on each of 33 machines. Following the proportion given by the subject matter experts, half of this amount, or 33 pieces of information can be obtained for the purposes of this example.

To create the detection schemes to test against Process Refinement, a random generation method was used. The machine to collect information from was chosen at random and the sensor that would be used was also selected randomly. A total of 33 unique machine/sensor combinations were generated for each detection scheme. Five total detection schemes were
created to test against Process Refinement. Diagrams for the detection schemes can be seen in Appendix I.

6.1.2 Testing Setup for Process Refinement

To evaluate a situation with Process Refinement, several parameters of the system needed to be defined. To ensure that Process Refinement could only obtain up to 33 pieces of information, the bandwidth and processing constraints of the system were set so that each piece of information gathered would cost one unit for each constraint and the maximum level for each constraint was 33. The actions that could be detected by the different sensors also needed to be defined and as mentioned previously, sensor one was set to recognize reconnaissance style actions and sensor two could recognize all other actions. The value of each machine also had to be defined. Table 17 shows the assignment of values. Highest value was given to internal network servers, then internal network workstations and finally to external network servers.

Table 17 - Value Assignment

<table>
<thead>
<tr>
<th>Machine</th>
<th>Value</th>
<th>Machine</th>
<th>Value</th>
<th>Machine</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2</td>
<td>13</td>
<td>0.8</td>
<td>25</td>
<td>0.6</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>14</td>
<td>0.8</td>
<td>26</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>15</td>
<td>0.9</td>
<td>27</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>16</td>
<td>0.5</td>
<td>28</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>17</td>
<td>0.5</td>
<td>29</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>18</td>
<td>0.5</td>
<td>30</td>
<td>0.4</td>
</tr>
<tr>
<td>7</td>
<td>0.3</td>
<td>19</td>
<td>0.5</td>
<td>31</td>
<td>0.4</td>
</tr>
<tr>
<td>8</td>
<td>0.3</td>
<td>20</td>
<td>0.5</td>
<td>32</td>
<td>0.7</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>21</td>
<td>0.5</td>
<td>33</td>
<td>0.7</td>
</tr>
<tr>
<td>10</td>
<td>0.9</td>
<td>22</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.8</td>
<td>23</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.8</td>
<td>24</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In addition to static input, process refinement contains two parameters whose input varies based on the situation. These parameters are threat and next possible actions. Each machine receives a threat score and set of next possible actions. The threat score is a one or zero parameter in which threat is equal to one if a machine is compromised or one machine away
from a compromised machine. Threat is equal to zero in all other cases. The set of next possible actions on a machine is determined by evaluating the actions that have occurred and consulting a graph-based template similar to the one discussed in section 4.2. For the purpose of this example, the graph-based template used is from Mathew et al. (2005).

6.1.3 Attack Scenarios and Evaluation Metrics

To fully test the Process Refinement system, attack situations were created in a variety of ways. The first stage of the test was used to evaluate Process Refinement’s performance as only a single attack is run against the network. The second stage becomes much more complicated as several coinciding attacks are run against the network at once. To remove bias from the analysis, all attack scenarios were created using the simulation model presented in section 4 with the automatic attack generation methodology highlighted in section 4.2.

In the first stage where only one attack was occurring at a time, attacks were designed to penetrate to deep levels of the network. Penetrating to the deepest levels of the network was done to show Process Refinement’s ability to track an attack as it progresses. These attacks were assigned varying parameters for efficiency and stealth so some attacks are short and to the point while others take a little more time. Five different attacks were created and each can be seen in Appendix J.

The second stage of the test was done to show how Process Refinement makes decisions on what information to gather when not all information can be gathered. With multiple attacks occurring at one time at different parts of the network, Process Refinement was forced to manage the detection resources more strictly to ensure that the highest value machines were being monitored. Five attack scenarios were generated for this stage as well and can be seen in Appendix K.
The primary measurement used to evaluate the detection schemes at stage one of the test was the percent of attack steps recognized. The reason that percent of steps recognized was chosen as the primary metrics was because stage one evaluated Process Refinement’s ability to track an entire attack through the network. The primary measurement used at stage two of the test was the sum of the value of the machines on which attack steps were detected. These metrics were used to evaluate Process Refinement’s ability to obtain information from the most valuable of threatened sources.

6.2 Single Attack Scenarios

Stage one of the test consisted of single attacks run against the sample network. Designed to show the power of Process Refinement in tracking attacks through a network, the primary performance metric for this stage was the percent of attack steps recognized.

6.2.1 Random Scenario Results

As discussed in section 6.1.1, several random detection scenarios were generated to evaluate against Process Refinement. Each attack generated was compared to each random detection scheme created and the detection scheme’s ability to detect each step of each attack noted. In addition, the machines that were compromised in each attack were examined to determine if any detection information was collected. The percent of machines compromised with information sources collected versus the total number of machines compromised in the attack was a secondary metric. The results from stage one of the test for the randomly generated detection schemes is shown in Table 18.
<table>
<thead>
<tr>
<th>Attack Step</th>
<th>Attack Scenario 1</th>
<th>Attack Scenario 2</th>
<th>Attack Scenario 3</th>
<th>Attack Scenario 4</th>
<th>Attack Scenario 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
<td>R5</td>
</tr>
<tr>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>9</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>12</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>13</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

| Total Found | 2 | 1 | 3 | 1 | 3 | 1 | 2 | 2 | 1 | 1 | 3 | 1 | 8 | 1 | 4 | 3 | 2 | 2 | 5 | 2 | 1 | 1 | 2 | 6 | 6 | 11 | 2 | 6 |
| Percentage  | 0.3 | 0.1 | 0.4 | 0.1 | 0.4 | 0.2 | 0.3 | 0.3 | 0.2 | 0.2 | 0.3 | 0.1 | 0.8 | 0.1 | 0.4 | 0.4 | 0.3 | 0.3 | 0.6 | 0.3 | 0.5 | 0.5 | 0.8 | 0.2 | 0.5 |
| Average     | 0.286 | 0.233 | 0.340 | 0.350 | 0.477 |

| Total Avg   | 0.337 |
| P/C Percent | 0.2 | 0.2 | 0.4 | 0.2 | 0.4 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.5 | 0.3 | 0.8 | 0.3 | 0.5 | 0.5 | 0.5 | 0.4 | 0.4 | 0.8 | 0.4 | 0.6 |
| Average     | 0.280 | 0.333 | 0.450 | 0.400 | 0.520 |
| Total Avg   | 0.397 |

The table shows each random detection scheme's ability to detect each step of each attack based on the randomly assigned information gathering sources. A "Y" in a box indicates that the attack step was detected and an "N" in a box indicates the attack step was not detected. The total number of attack steps detected by the detection scheme is listed in the row labeled "Total Found." The percentage of the attack steps that were found is shown in the next row down labeled "Percentage." The average percent of attack steps detected across all detection schemes for a given attack scenario is shown in the next row labeled "Average." The row labeled "Total Average" shows the average detection percentage across all detection schemes and attack scenarios. The secondary metric of compromised machines with information sources gathered versus all compromised machines is shown in the row labeled "P/C Percent" where "P/C" stands for Protected/Compromised. The average of the "P/C Percent" metrics for a given detection scheme is shown in the next row labeled "Average." Lastly, the next row labeled "Total
Average shows the average detection percentage across all detection schemes and all attack scenarios.

The data in the table shows that even the highest detection percent average, the average for attack five, has only 47% of attack steps detected on average with other attack scenarios in the range of 30%. In addition, the average value of "P/C Percent" for each attack scenario is highest for attack scenario five but only with a value of 52%. The results from the first stage of the test with the random detection schemes indicate that using a random detection scheme will at best detect only half of the attack actions occurring against the network with the average number of steps detected around 30%.

6.2.2 Process Refinement Recommendations

To determine the recommendation from Process Refinement for a given situation, two parameters needed to be defined for each machine in the network: threat and next possible actions. As the situation changes in the network due to proliferating attacks, the threat to machines and the next possible actions that can occur on those machines changes. The first thing that had to be done to run Process Refinement was to find out at one points in time the parameters of Process Refinement would change, triggering a recommendation to be generated. Running the system under equal parameters would not change the recommendation. Thus, the points in time at which the parameters of Process Refinement would change needed to be determined for each attack scenario. Table 19 describes the points in time when the parameters of Process Refinement would change for each attack.

The table shows the points in time at which the parameters of Process Refinement change and thus, the recommendation made by Process Refinement has the potential to change. A "C" in a box indicates that a change in Process Refinement parameters occurred after the attack step
corresponding to the “C” was executed. Thus, Process Refinement would give a recommendation after the attack step executed and would be valid for all attack steps subsequent until another recommendation was made.

Table 19 - Process Refinement Changes Stage 1

<table>
<thead>
<tr>
<th>Attack Step</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At attack step zero, or the point in time before the first step of the attack occurs, there are no attacks occurring against the network. At any such time, threats to machines on the internal network are assumed to be zero and threats to the machines on the external network are assumed to be one. The set of next possible actions for machines on the internal network is assumed to empty and the set of next possible actions for external network machines is assumed to be actions from stage zero of the graph-based template.

As attack steps occur, the parameters of Process Refinement change and a new recommendation can be made. For example, when attack step one of the first attack, a reconnaissance action, occurs, the threat values stay the same because the attack can not technically penetrate the internal network until an intrusion occurs. However, the set of next possible actions is updated because an intrusion or goal action could occur next on the external machines. Thus, Process Refinement can provide a recommendation because the next possible actions parameter has changed. Similarly, when attack step two of attack one occurs, this time
an intrusion, both the threat and next actions parameters are updated and Process Refinement can make a new recommendation. The recommendations made at each point indicated in Table 19 can be seen in Appendix L.

A similar results table to the one shown in section 6.2.1 was generated for the recommendations given by Process Refinement as well. The results for Process Refinement can be seen in Table 20. This table shows that Process Refinement was able to track each attack as it traveled through the network, detecting every step of each attack for a 100% detection average. As a result, each machine compromised was protected by some information gathering resource for a “P/C Percent” average of 100% as well.

Table 20 - Process Refinement Results Stage 1

<table>
<thead>
<tr>
<th>Attack Step</th>
<th>Process Refinement</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A1</td>
<td>A2</td>
</tr>
<tr>
<td>1</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>12</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>13</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Found Percentage Average</th>
<th>Total Avg</th>
<th>P/C Percent Average</th>
<th>Total Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

90
6.2.3 Comparison of Random Detection to Process Refinement Detection

Table 21 shows the results of all random detection schemes and the results from Process Refinement together. The data in the table shows that Process Refinement system was much better equipped to detect the attack actions of each attack as the attacks propagated through the network. Process Refinement was also much better at protecting machines that ended up being compromised.

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>Process Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Average Percent</td>
<td>0.337</td>
<td>1.000</td>
</tr>
<tr>
<td>Total Average P/C Percent</td>
<td>0.397</td>
<td>1.000</td>
</tr>
</tbody>
</table>

The first stage of the test was designed to show that when Process Refinement is not forced to choose between detection resources, it has the ability to track single attacks throughout their entirety. The next stage of the test will investigate Process Refinement’s performance when multiple attacks are occurring in the network at once.

6.3 Multiple Attack Scenarios

Stage two of the test consisted of multiple attacks occurring against the network at once. Stage two was designed to show that Process Refinement will obtain information for the highest valued network resources in a resource constrained situation. Again, the random detection schemes described in 6.1.1 were compared to detection schemes recommended by Process Refinement for each multiple attack scenario.

6.2.1 Random Scenario Results

In stage two of the test, the random detection schemes were once again evaluated to determine their performance. At this stage, the primary metric was the sum of the Process Refinement value of compromised machines on which detection information was gathered. The metrics from stage one, percent of attack steps recognized and percent of compromised machines
protected, were gathered at stage two as well. The results for the random detection schemes can be seen in Table 22.

**Table 22 - Results for Random Detection Stage 2 Primary Metrics**

<table>
<thead>
<tr>
<th>Time Step</th>
<th>Attack Scenario 1</th>
<th>Attack Scenario 2</th>
<th>Attack Scenario 3</th>
<th>Attack Scenario 4</th>
<th>Attack Scenario 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
<td>R5</td>
</tr>
<tr>
<td>1</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>12</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>13</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>14</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>16</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>17</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>18</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>19</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>20</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>21</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>22</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>23</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>24</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

| Value Sum | 3.4 | 0.8 | 2.0 | 0.8 | 3.2 | 0.7 | 0.9 | 1.2 | 0.9 | 1.2 | 2.6 | 5.2 | 1.6 | 3.9 | 3.0 | 1.6 | 3.6 | 1.0 | 4.0 | 2.6 | 1.4 | 2.1 | 1.7 | 3.3 |
| Value %   | 0.6 | 0.1 | 3.0 | 0.1 | 6.0 | 0.2 | 0.2 | 3.0 | 0.2 | 3.0 | 0.3 | 0.6 | 0.2 | 0.5 | 0.4 | 0.2 | 0.5 | 0.1 | 0.5 | 0.4 | 0.2 | 0.3 | 0.3 | 0.5 |
| Average   | 0.352 | 0.265 | 0.376 | 0.347 | 0.342 |
| Total Avg | 0.336 | 0 |

Table 22 shows the results from comparing each multiple attack scenario to the random detection schemes to determine which attack steps would be recognized. The individual attacks and steps are not distinguished in this table for simplicity but can be found in Appendix K. At each time step, a new attack step occurs from one of the three attacks. A "Y" in the box indicates that the attack step was detected and an "N" indicates that the attack step was not detected. The row labeled "Value Sum" shows the sum of the Process Refinement values of each machine that was compromised on which information was gathered. The row labeled "Value %" shows the "Value Sum" divided by the sum of the values of all compromised
machines in the particular multiple attack scenario. The next row labeled “Average” shows the average of the “Value %” statistic across all random detection schemes. The row labeled “Total Average” shows the average of the “Value %” across all detection schemes and all attack scenarios. The metrics from the first stage of the test were also collected as secondary metrics and can be seen in Table 23.

Table 23 - Results for Random Detection Stage 2 Secondary Metrics

<table>
<thead>
<tr>
<th>Attack Scenario 1</th>
<th>Attack Scenario 2</th>
<th>Attack Scenario 3</th>
<th>Attack Scenario 4</th>
<th>Attack Scenario 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Found</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R1 R2 R3 R4 R5</td>
<td>R1 R2 R3 R4 R5</td>
<td>R1 R2 R3 R4 R5</td>
<td>R1 R2 R3 R4 R5</td>
</tr>
<tr>
<td>B</td>
<td>3 4 9 13</td>
<td>3 4 9 6</td>
<td>8 9 12 4 15</td>
<td>6 7 11 2 14</td>
</tr>
<tr>
<td>Percentage</td>
<td>0.3 0.1 0.2 0.2 0.5</td>
<td>0.2 0.3 0.6 0.4 0.5</td>
<td>0.3 0.4 0.5 0.2 0.7</td>
<td>0.3 0.3 0.5 0.1 0.6</td>
</tr>
<tr>
<td>Average</td>
<td>0.306</td>
<td>0.375</td>
<td>0.417</td>
<td>0.333</td>
</tr>
<tr>
<td>Total Avg</td>
<td></td>
<td></td>
<td></td>
<td>0.365</td>
</tr>
<tr>
<td>P/C Ratio</td>
<td>0.6 0.2 0.3 0.3 0.6</td>
<td>0.1 0.4 0.4 0.4 0.4</td>
<td>0.3 0.3 0.6 0.3 0.6</td>
<td>0.4 0.3 0.5 0.2 0.5</td>
</tr>
<tr>
<td>Average</td>
<td>0.400</td>
<td>0.371</td>
<td>0.417</td>
<td>0.362</td>
</tr>
<tr>
<td>Total Avg</td>
<td></td>
<td></td>
<td></td>
<td>0.408</td>
</tr>
</tbody>
</table>

Tables 22 and 23 show that in multiple attack scenarios, the random detection schemes still only obtain about 33% of the available important information on average. The percentage of the value of compromised machines obtained is highest when looking at attack scenario three but the detection schemes can still only obtain 37% of the total value compromised. Additionally, the random detection schemes were able to detect at best 47% of the attack steps with an average of 36% of attack steps found. Finally, the random detection schemes were able to obtain information from at best 47% of compromised machines with an average of 41%.

6.2.2 Process Refinement Recommendations

In order for Process Refinement to run, the two main parameters of Process Refinement needed to be defined for each distinct situational change in time. The two parameters were threat and next possible actions and each parameter needed to be defined for each machine. The points during each attack where the situation changes in terms of Process Refinement input are shown in Table 24.
Table 24 - Process Refinement Changes Stage 2

<table>
<thead>
<tr>
<th>Time Step</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The points at which a Process Refinement recommendation would occur are indicated in the table by a "C" in the box. It is important to note that a "C" indicates that a recommendation will be created based on the data from the step corresponding to the particular "C." Thus, the step directly after a "C" and any other steps through the next point at which a recommendation will occur in the attack scenario will be monitored by the recommendation given at the point of the "C". For example, when step three of the attack scenario M1 occurs, it initiates a Process Refinement recommendation as indicated by the "C" in the box for M1 and step three. Step four, five, and six will all be monitored by the recommendation initiated at three. Once six is complete however, a new recommendation is necessary as indicated by the "C" in the box corresponding to M1 and step six. This new recommendation will be responsible for monitoring
step seven, and so on. The recommendations created for each of the points shown in table 24 can be found in Appendix M.

Process Refinement's performance with multiple attacks occurring in the network at once is shown in Table 25. The table shows that Process Refinement was able to obtain on average 97% of the total value of compromised machines and in attack scenario two, Process Refinement was able to obtain 100% of the value of compromised machines.

**Table 25 - Process Refinement Results Stage 2 Primary Metrics**

<table>
<thead>
<tr>
<th>Time Step</th>
<th>Process Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M1</td>
</tr>
<tr>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>Y</td>
</tr>
<tr>
<td>6</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>Y</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
</tr>
<tr>
<td>11</td>
<td>Y</td>
</tr>
<tr>
<td>12</td>
<td>Y</td>
</tr>
<tr>
<td>13</td>
<td>Y</td>
</tr>
<tr>
<td>14</td>
<td>Y</td>
</tr>
<tr>
<td>15</td>
<td>Y</td>
</tr>
<tr>
<td>16</td>
<td>Y</td>
</tr>
<tr>
<td>17</td>
<td>Y</td>
</tr>
<tr>
<td>18</td>
<td>Y</td>
</tr>
<tr>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>20</td>
<td>Y</td>
</tr>
<tr>
<td>21</td>
<td>Y</td>
</tr>
<tr>
<td>22</td>
<td>Y</td>
</tr>
<tr>
<td>23</td>
<td>Y</td>
</tr>
<tr>
<td>24</td>
<td>Y</td>
</tr>
</tbody>
</table>

Value Sum | 5.6 | 3.7 | 8.1 | 7.5 | 6  |
Value %    | 0.97| 1.00| 0.96| 0.99| 0.92|
Average    |     |     |     |     |    |
Total Avg  | 0.970|

The metrics from stage one of the test were evaluated for Process Refinement in stage two as well and are shown in Table 26. Table 26 shows that Process Refinement was able to
detect on average 93% of the attack steps in all attack scenarios. Process Refinement was also able to obtain information from 92% of compromised machines.

Table 26 - Process Refinement Results Stage 2 Secondary Metrics

<table>
<thead>
<tr>
<th>Process Refinement</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Found</td>
<td>23</td>
<td>16</td>
<td>22</td>
<td>22</td>
<td>19</td>
</tr>
<tr>
<td>Percentage</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>0.92</td>
<td>0.86</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Avg</td>
<td>0.939</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P/C Ratio</td>
<td>0.90</td>
<td>1.00</td>
<td>0.92</td>
<td>0.91</td>
<td>0.86</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Avg</td>
<td>0.917</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.2.3 Comparison of Random Detection to Process Refinement Detection

Table 27 shows the averages of the performance metrics for the random detection schemes and Process Refinement across all attack scenarios. The table shows that Process Refinement is superior to random detection across all performance metrics. Process Refinement obtains more of the percent value of compromised machines, detects attack steps at a greater percentage rate, and obtains information from compromised machines at a greater percentage rate as well.

Table 27 - Results for Random Detection and Process Refinement Stage 2

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>Process Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Value %</td>
<td>0.336</td>
<td>0.970</td>
</tr>
<tr>
<td>Average % Found</td>
<td>0.365</td>
<td>0.939</td>
</tr>
<tr>
<td>Average P/C Percent</td>
<td>0.408</td>
<td>0.917</td>
</tr>
</tbody>
</table>

However, a major difference between the results from stage one and the results from stage two is that Process Refinement did not perform perfect in stage two as it did in stage one. Process Refinement is not a perfect system and is still confronted by the fact that not all network resources can be obtained. Missed attack steps are bound to occur especially as the breadth of attacks and number of attacks increases. The next section of the test describes some of the limitations of Process Refinement.
6.4 Process Refinement Failure

Although Process Refinement is excellent at tracking a single attack through a network and obtaining highest value information when confronted with constraints, situations exist in which Process Refinement will fail. This section highlights the situations in which Process Refinement can fail and describes why the failures occur.

6.4.1 Failure Reasons

The reason that Process Refinement can fail is because the network has far more information than Process Refinement can choose to obtain. Tradeoffs are forced to be made as more attacks occur and as the size of the network grows, the number of these tradeoffs that Process Refinement must make grows as well. Process Refinement decides between equally threatened machines by examining the value specified for the machines. Higher value machines have a higher obtainment priority than lower value machines. Thus, at some point, a newly threatened high value machine may take the place of a still threatened but lower value machine.

As a result, the specification of the value of the machines is extremely important to Process Refinement’s effectiveness. Some analysts may choose to specify their external machines as highest value to ensure that all attacks are recognized when they start. The problem with this method is that once the attack proceeds into the network, detection resources may not be available to track the attack because they are assigned to the higher value external machines. Other analysts might choose to place the highest value on machines on the internal network. Unfortunately, placing highest value on internal machines may prevent these analysts from recognizing attacks when they begin in the network, making the attacks harder to track.

Whatever the analysts decision, at some point in time, an attack step can be missed. The best allocation of value is hard to determine empirically and should more likely be based on
specific networks with unique attributes and qualities. The following example illustrates the problem with value assignment.

6.4.2 Example of Failure

To show the potential for failure in Process Refinement, an example attack scenario has been created. This attack scenario consists of three different attacks, each targeting a different machine in the network. The example network described for the first two stages of testing is used for this example as well. The three target machines are shown in Figure 28.

Figure 31 - Targeted Machines

The machine on the left and the machine in the middle were chosen because attacking those machines would result in a large number of machines threatened. The machine on the right
was chosen because it was on the opposite side of the network from the other two machines. The attack scenario was setup so that the attacks targeting the left and middle machine shown above would execute at the same time. The attack targeting the machine on the right was delayed so that the attack would not start until the first two attacks were almost complete. As a result, when the third attack begins, there are already a large amount of machines threatened in the network. For more detailed information about the attack scenario, refer to Appendix N.

This attack scenario was designed so that detection resources could be completely allocated to detecting and tracking actions occurring in the first two attacks when the third attack starts. To ensure that detection resources will be allocated in this manner, the values needed to be defined in a specific way. The value assignments are shown in Table 28.

<table>
<thead>
<tr>
<th>Machine</th>
<th>Value</th>
<th>Machine</th>
<th>Value</th>
<th>Machine</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5</td>
<td>13</td>
<td>0.6</td>
<td>25</td>
<td>0.7</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>14</td>
<td>0.6</td>
<td>26</td>
<td>0.7</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>15</td>
<td>0.9</td>
<td>27</td>
<td>0.7</td>
</tr>
<tr>
<td>4</td>
<td>0.2</td>
<td>16</td>
<td>0.8</td>
<td>28</td>
<td>0.7</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
<td>17</td>
<td>0.8</td>
<td>29</td>
<td>0.3</td>
</tr>
<tr>
<td>6</td>
<td>0.2</td>
<td>18</td>
<td>0.8</td>
<td>30</td>
<td>0.1</td>
</tr>
<tr>
<td>7</td>
<td>0.4</td>
<td>19</td>
<td>0.8</td>
<td>31</td>
<td>0.1</td>
</tr>
<tr>
<td>8</td>
<td>0.4</td>
<td>20</td>
<td>0.8</td>
<td>32</td>
<td>0.1</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>21</td>
<td>0.8</td>
<td>33</td>
<td>0.1</td>
</tr>
<tr>
<td>10</td>
<td>0.9</td>
<td>22</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.6</td>
<td>23</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.6</td>
<td>24</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table shows that the values are assigned such that machines on the left branch of the network are much higher in value than machines on the right side of the network. This results in machines on the left side of the network having priority over machines on the right side of the network. As the first two attacks target machines on the left side of the network, these machines will become threatened first and detection resources will allocated accordingly by Process Refinement. When the third attack begins to penetrate the right side of the network, the
machines on the right side of the network will become threatened. However, since the machines on the right side of the network are lower in value, no information will be obtained from them, resulting in the entire third attack going undetected. Table 29 shows the points in time at which Process Refinement made a recommendation, the attack and step that occur at that point, and whether the step was detected or not.

Table 29 - Failure Scenario Results

<table>
<thead>
<tr>
<th>PR Update</th>
<th>Attack, Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Step</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19</td>
</tr>
<tr>
<td>Found?</td>
<td>Y Y Y Y N Y Y Y Y Y N N N N</td>
</tr>
</tbody>
</table>

This table shows that Process Refinement made a recommendation at points one, three, four, and five in order to track the first two attacks as the attacks progressed through the network. These recommendations can be seen in Appendix N. However, when the third attack began, Process Refinement did not even make a recommendation because no detection resources were allocated to right side of the network. Therefore, no change to the Process Refinement parameters occurred because the entire system did not detect any of the actions at the start of attack three. In this particular case, even if the beginning of attack three had been detected, Process Refinement’s recommendation would not change because Process Refinement would already be obtaining information from the highest value machines.

The miss of an entire attack in an attack scenario can be considered a critical failure of Process Refinement. This example illustrates how sensitive Process Refinement is to the assignment of value and the resulting importance that the assignment of value should have.

6.5 Testing Conclusions

This test has shown that Process Refinement has a better ability to detect and track attack steps as single attacks progress through a network than random detection schemes. The test has
also shown that in the same single attack case, Process Refinement will obtain information from far more compromised machines than random detection schemes. As more attacks occur in the network at the same time, Process Refinement still has the ability to detect far more of the attack steps than a random detection scheme and additionally, Process Refinement obtains information from machines with high value with a much higher percentage than random detection schemes.

However, as was shown in section 6.4, Process Refinement is not a perfect system and attack steps will be missed at times. In extreme circumstances, Process Refinement may even miss all steps of an entire attack but this is rare and highly dependant on user specified Process Refinement parameters. Although Process Refinement may miss entire attacks, the highest value machines in the network as defined by the user will still be protected.
7. Conclusions and Recommendations for Future Research

Hackers are a growing problem both due to the growing number that exist and the increasing danger from their actions. However, even as hackers become more numerous and more dangerous, technological advances continue to develop to thwart hackers (McClure, Scambray, & Kurtz, 2001). Simulating the actions of hackers allows analysts to learn more about how hackers act and behave, providing a better understanding of how to stop hackers. Additionally, information fusion techniques provide analysts with better information that they can act on when hackers attack their networks.

The simulation model presented using the automatic attack generation methodology is designed to show analysts a number of ways that hackers can attack networks. In addition, these automatically generated attacks can be used to test either network analysts or systems developed to assist analysts in detecting, identifying, and tracking hacker attacks. Additional machine attributes were added to the simulation model as well to assist in creating more realistic attacks. Validation of the simulation model presented provides insurance that the conclusions drawn from the simulation model results will have similar bearing in the real world.

Even though information fusion systems developed to assist network analysts improve the analyst’s ability to track hacker attacks, the amount of potentially valuable information in a network is enormous and often far too much for a fusion system to handle. The Process Refinement model presented allows the fusion system to focus on the most important data and is adaptive to changing situations in the network. This adaptive nature allows the fusion system as a whole to adapt and change as hacker behavior adapts and changes.

Additionally, the validated simulation model was used to test the presented Process Refinement model to ensure that the Process Refinement model would be able to help the fusion
The Process Refinement model exhibited the ability to detect and track single attacks through a network and also to obtain information from the highest valued machines in a network when faced with constrained detection resources. Through testing, the Process Refinement model was shown to perform better than a number of random detection schemes generated across multiple metrics of performance.

Testing also showed some sensitivity of the Process Refinement model to the specification of machine value. A rare case was shown in which Process Refinement missed an entire attack in a scenario where multiple attacks were occurring at once. However, Process Refinement still obtained information from the highest valued machines as specified in the test scenario.

To further develop the simulation model presented, the human-computer interface (HCI) should be redesigned to streamline the attack specification process. Specifically, when creating a simulated network, the number of modules that must be used in a model should be reduced by using modules that represent entire groups of machines. Additionally, the user's control of the attack action and alert definitions should be improved so that updates and changes can be made quickly and easily. In the automatic attack generation methodology, additional measures of hacker behavior beyond stealth and efficiency should be created in order to model a wider variety of hacker attack scenarios. Finally, the automatic attack generation methodology should be improved to allow for more depth in branches of an attack as described in section 4.2.3 and to allow for more steps to be created in each attack.

Development of Process Refinement can take two separate branches. First, the existing model can be improved to account for additional network constraints beyond bandwidth and processing power. The existing model could also be improved by investigating further the
sensitivity of value assignment to potentially determine a best case scenario for value assignment. Testing of the existing model could also be continued to see how Process Refinement performs as even more attacks are executed at the same time or how Process Refinement performs when resources are constrained more heavily. The initial model could also be tested against subject matter expert defined detection schemes rather than just random to further identify the benefits of Process Refinement.

The second branch of development that Process Refinement could take is to examine other areas of the fusion system that could be improved on a situational basis. The initial model presented obtains information from fusion level three and acts to improve fusion level one: Object Refinement where initial information detection is performed. As described in section 5.2, there could be a number of other modules of Process Refinement that obtain information from the fusion system and act to improve the fusion system’s capabilities.

Improving both the simulation methodology presented and the Process Refinement system can further help with the growing problem of hackers. Understanding hacker’s behavior and intent is a critical part of protecting a computer network. Additionally, obtaining information in a manner that maximizes the value of the information and reduces the amount of useless information that must be process can make the difference between catching the hacker and allowing them to slip away.
References


Appendices

Appendix A: Original Simulation Model

Shown below are screen shots from the original simulation model created for the AFRL (Kuhl & Kistner, 2005). The screen shots included show the network building tools, backbone simulation logic, and attack creation graphical user interface.

Network Building Tools

Figure E1 – Network Template

Figure E2 – Switch Information Box
Figure E3 – Machine Information Box

Attack Entity Logic

Create Attacks and Assign Properties

Figure E4 – Attack Entity Creation Logic
Computer Network

Figure E5 – Attack Entity Flow Logic

Noise Creation

Figure E6 – Noise Creation Logic
Graphical User Interface

Figure E7 – Attack Scenario Load
IDS Alert Files

Select a file

Example, GroundTruthActions

Open File

Edit File Name

Delete File

Done

Figure E8 – Scenario File Display
Welcome to the IDS Sensor Alerts Simulator

Scenario Name:
Use letters and numbers only. Some special characters are not recognized.

Delay Distribution
- Select random values from an exponential distribution
- Use constant values

Delay Method
- Input total time for attack and split time up between steps
- Input specific delays between steps of attacks

Noise
Type of noise: Percentage:
- Reconnaisance
- Escalation
- Intrusion
- Goal
- Miscellaneous

Alerts per hour:
0

Time to Run After Last Attack is Complete: 0 Minutes

Add Attack
Edit Attack
Delete Attack
Run Simulation

Figure E9 – Attack Generation Main Screen

Figure E10 – Attack Generation Add/Edit Attack
Appendix B: Original Simulation Model - Attack Example

This section describes an attack example created using the original simulation methodology generated for the AFRL (Kuhl & Kistner, 2005). Detailed below is the simulated computer network that was created and the resulting data files created.

The computer network created using the Cyber Attack Simulator is shown in Figure F1. A summary of the network is as follows:

- 1 main web-server;
- 4 main subnet domains;
- 3 subnet domains have further subnets attached;
- Only one external machine; and
- Red dots indicate snort sensor presence

![Figure F1 – Network Configuration](image-url)
Figure F2 – Ground Truth Files

Figure F3 – IDS Alert Files
Appendix C: Automatic Attack Generation Code Logic

New Graphical User Interface Documentation for Automatic Attack Generation

The interface for automatic attack generation is slightly different than the interface for manual attack generation. This is a documentation of the interface and an explanation of the differences between the two interfaces.

First choose “Use attack scenario creation interface”

Next choose “Quick and Dirty Method”
The screen that comes up looks like this:
<table>
<thead>
<tr>
<th><strong>Target IP:</strong></th>
<th>IP address of the machine you wish the final goal to be achieved on.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal on Target:</strong></td>
<td>The final goal of the hack being simulated</td>
</tr>
<tr>
<td><strong>Efficiency of Hacker:</strong></td>
<td>Measures how quickly the hacker will penetrate the network and achieve the goal. Also a measure of how successful the hacker’s actions are. Think of this as a measure of the hackers skill level.</td>
</tr>
<tr>
<td><strong>Stealth of Hacker:</strong></td>
<td>Measures the amount of time a hacker will execute a goal when a stepping stone machine is compromised</td>
</tr>
<tr>
<td><strong>Delay of Attack:</strong></td>
<td>The time before the attack will start</td>
</tr>
<tr>
<td><strong>Step Delay:</strong></td>
<td>The average delay time between steps of the attack</td>
</tr>
</tbody>
</table>

In this interface, the user will specify a scenario name, which will correspond to the text files generated and the folder they are stored in. Then the user will input the number of attacks they wish to create in the first text box (limit is 10 attacks). Pressing the update command button will increase or decrease the number of pages in the multipage based on the number of attacks entered. The user then specifies the information listed above for each attack on subsequent pages. The user then specifies some noise information (this is the same as the manual interface) and the time to run after the last attack completes.
Appendix D: Verification and Validation

This section shows the variables and stations created in SIMAN, the simulation programming language, when the user creates a simulated network in the model. Also shown are a series of sensor configurations on a simulated network and the corresponding alert files that are created based on the sensor configuration

Variables and Stations from Template

VARIABLES: StepsPerAttack(10), CLEAR(System), CATEGORY("None-None"):
MaxArgs, CLEAR(System), CATEGORY("None-None"), 10:
AutoAttack, CLEAR(System), CATEGORY("None-None"):
AttacksComplete, CLEAR(System), CATEGORY("None-None"):
NumInSubGroup(5,7), CLEAR(System), CATEGORY("None-None"):
AvgStepDelay(10), CLEAR(System), CATEGORY("None-None"):
AttackDelay(10), CLEAR(System), CATEGORY("None-None"):
AddRunTime, CLEAR(System), CATEGORY("None-None"):
StepDelay(10,10), CLEAR(System), CATEGORY("None-None"):
AttackCompleteTime, CLEAR(System), CATEGORY("None-None"):
SnortOn, CLEAR(System), CATEGORY("None-None"), 1:
NoiseMax, CLEAR(System), CATEGORY("None-None"):
Expon, CLEAR(System), CATEGORY("None-None"):
Attack(300,10), CLEAR(System), CATEGORY("None-None"):

0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,100,1,1,1,100,1,1,1,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0:
Machine 1(8), CLEAR(System), CATEGORY("None-None"):
Machine 2(8), CLEAR(System), CATEGORY("None-None"):
Machine 3(8), CLEAR(System), CATEGORY("None-None"):
Machine 4(8), CLEAR(System), CATEGORY("None-None"):
Machine 5(8), CLEAR(System), CATEGORY("None-None"):
Machine 6(8), CLEAR(System), CATEGORY("None-None"):
Machine 7(8), CLEAR(System), CATEGORY("None-None"):
Machine 8(8), CLEAR(System), CATEGORY("None-None"):
Machine 9(8), CLEAR(System), CATEGORY("None-None"):
Machine 10(8), CLEAR(System), CATEGORY("None-None"):

121
STATIONS:

1. Machine 1 Station,, AUTOSTATS (Yes,);
2. Machine 2 Station,, AUTOSTATS (Yes,);
3. Machine 3 Station,, AUTOSTATS (Yes,);
4. Machine 4 Station,, AUTOSTATS (Yes,);
5. Machine 5 Station,, AUTOSTATS (Yes,);
6. Machine 6 Station,, AUTOSTATS (Yes,);
7. Machine 7 Station,, AUTOSTATS (Yes,);
8. Machine 8 Station,, AUTOSTATS (Yes,);
9. Machine 9 Station,, AUTOSTATS (Yes,);
10. Machine 10 Station,, AUTOSTATS (Yes,);
11. Machine 11 Station,, AUTOSTATS (Yes,);
12. Machine 12 Station,, AUTOSTATS (Yes,);
13. Machine 13 Station,, AUTOSTATS (Yes,);
14. Machine 14 Station,, AUTOSTATS (Yes,);
15. Machine 15 Station,, AUTOSTATS (Yes,);
16. Machine 16 Station,, AUTOSTATS (Yes,);
17. Machine 17 Station,, AUTOSTATS (Yes,);
18. Machine 18 Station,, AUTOSTATS (Yes,);
19. Machine 19 Station,, AUTOSTATS (Yes,);
20. Machine 20 Station,, AUTOSTATS (Yes,);
101. Ground Truth,, AUTOSTATS (Yes,);
102. Snort Alerts,, AUTOSTATS (Yes,);
Example Networks and Alert Files Created

Configuration 1
### Configuration 1 Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Type</th>
<th>Date Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration 1</td>
<td>15 KB</td>
<td>GIF Image</td>
<td>6/20/2006 1:57 PM</td>
</tr>
<tr>
<td>Configuration 1_DragonGroundTruth</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_DragonHost_3</td>
<td>4 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_GroundTruthActions</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_HttpInspectGroundTruth</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_SnortandHttpInspectGroundTruth</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_SnortGroundTruth</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_SnortHost_3</td>
<td>5 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_SnortHost_5</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_SnortNetwork_1</td>
<td>12 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
<tr>
<td>Configuration 1_SnortNetwork_3</td>
<td>1 KB</td>
<td>Text Document</td>
<td>6/21/2006 12:19 PM</td>
</tr>
</tbody>
</table>
Configuration 2
Configuration 4
Appendix E: Process Refinement Testing Setup - Random Detection Scenarios

This section shows the random detection schemes that were created to test against Process Refinement.

Random Detection Schemes

Detection Scheme 1
Detection Scheme 2
Detection Scheme 3
Detection Scheme 4
Appendix F: Process Refinement Testing Setup - Single Attack Scenarios

This section shows the inputs to the simulation methodology used to create the single attack scenarios for testing Process Refinement. The resulting attack output from the simulation is also shown. The screen shot below shows the input for Single Attack Scenario 1 and the table summarizes the inputs for Scenarios 2-5.

Single Attack 1

![Automatic Attack Generation Screen](image)

- **Scenario Name:** Single Attack 1
- **Target IP:** 100.50.4.1
- **Efficiency of Hacker (0-1):** 0.9
- **Stealth Level of Hacker (0-1):** 0.9
- **Goal on Target:** Dos
- **Delay This Attack By (mins):** 10
- **Average Delay Time Between Steps:** 10
- **Percent of time to encode step (0-1):** 1
- **Alerts Per Hour:** 30000
- **Type of Noise:** 100
- **Time to Run After Last Attack is Complete:** 5
Simulation Input for Scenarios 2-5

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Target</th>
<th>Goal</th>
<th>Efficiency</th>
<th>Stealth</th>
<th>Attack Delay</th>
<th>Step Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>100.50.3.1</td>
<td>Phishing</td>
<td>0.9</td>
<td>0.7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>100.40.3.6</td>
<td>Backdoor</td>
<td>0.8</td>
<td>0.7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>100.60.3.2</td>
<td>Backdoor</td>
<td>0.7</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>100.60.3.4</td>
<td>DoS</td>
<td>0.7</td>
<td>0.7</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Scenario 1 Results

Attack 1, Step 1, Recon, Footprinting, RPC AMD TCP version request, Encoded, 137.28.255.173, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, OS, NETBIOS DCERPC Messenger Service buffer overflow attempt, Encoded, 122.49.175.191, ->, 100.10.1.2, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 2, Escalation, OS, NETBIOS DCERPC Messenger Service buffer overflow attempt, Encoded, 162.182.4.110, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 3, Escalation, OS, EXPLOIT SCO calserver overflow, Encoded, 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, Root, WEB-COLDFUSION startstop DOS access, Encoded, 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Intrusion, User, WEB-CGI rsh access, Encoded, 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt, Encoded, 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 7, Goal, Dos, WEB-FRONTPAGE contents.htm access, Encoded, 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS

Scenario 2 Results

Attack 1, Step 1, Recon, Enumeration, WEB-PHP myPHPNuke chatheader.php access, Encoded, 211.151.252.233, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, Service, NETBIOS SMB Data Service Session Setup AndX request username overflow attempt, Encoded, 104.217.211.172, ->, 100.10.1.3, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 2, Escalation, Service, FTP STOU overflow attempt, Encoded, 4.110.103.71, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 3, Intrusion, Root, NETBIOS SMB CD..., Encoded, 100.10.1.3, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, User, WEB-CGI upload.pl access, Encoded, 100.10.1.3, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Misc, Other, DNS EXPLOIT x86 FreeBSD overflow attempt, Encoded, 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Goal, Filering, BACKDOOR MISC Solaris 2.5 attempt, Encoded, 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS

Scenario 3 Results

Attack 1, Step 1, Intrusion, Other, WEB-PHP viewtopic.php access, Encoded, 5.147.26.27, ->, 100.20.1.1, Prob of Success =, 0.8, ->, FAIL
Attack 1, Step 1, Intrusion, Other, WEB-PHP PayPal Storefront arbitrary command execution attempt, Encoded, 138.110.130.59, ->, 100.20.1.1, Prob of Success =, 0.8, ->, FAIL
Attack 1, Step 1, Intrusion, Other, WEB-PHP viewtopic.php access, Encoded, 110.103.71.252, ->, 100.20.1.1, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 2, Intrusion, Root, WEB-PHP Advanced Poll admin_password.php access, Encoded, 204.178.103.5, ->, 100.20.1.1, Prob of Success =, 0.8, ->, FAIL
Attack 1, Step 2, Intrusion, Root, WEB-PHP IdeaBox cord.php file include, Encoded, 253.121.2.109, ->, 100.20.1.1, Prob of Success =, 0.8, ->, FAIL
Attack 1, Step 2, Intrusion, Root, WEB-CGI quickstore.cgi access, Encoded, 156.94.135.235, ->, 100.20.1.1, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 3, Goal, Pilfering, MISC rsyncd module list access, Encoded, 207.196.236.13, ->, 100.20.1.1, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 4, Recon, Enumeration, WEB-MISC MsMMask.exe access, Encoded, 58.130.159.155, ->, 100.10.1.2, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 5, Intrusion, Other, WEB-PHP b2 cafelog gm-2-b2.php remote command execution attempt, Encoded, 63.51.249.46, ->, 100.10.1.2, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 6, Intrusion, Other, WEB-CLIENT readme.eml download attempt, Encoded, 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 7, Goal, Pilfering, WEB-COLDFUSION set odbc ini attempt, Encoded, 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 8, Goal, Pilfering, WEB-ATTACKS tftp command attempt, Encoded, 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 9, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt, Encoded, 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.8, ->, FAIL
Attack 1, Step 9, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt (ADMv2), Encoded, 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.8, ->, FAIL
Attack 1, Step 9, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt (ADMv2), Encoded, 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.8, ->, SUCCESS
Attack 1, Step 10, Goal, Backdoor, BACKDOOR Infector.1.x, Encoded, 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.8, ->, SUCCESS

Scenario 4 Results

Attack 1, Step 1, Misc, Other, POLICY FTP CWD / possible warez site, Encoded, 26.27.204.73, ->, 100.20.1.3, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 1, Misc, Other, POLICY FTP STOR 1MB possible warez site, Encoded, 39.136.57.150, ->, 100.20.1.3, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 1, Misc, Other, POLICY FTP CWD / possible warez site, Encoded, 102.231.191.23, ->, 100.20.1.3, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 2, Goal, Dos, BAD-TRAFFIC IP Proto 77 (Sun ND), Encoded, 132.247.143.232, ->, 100.20.1.3, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 2, Goal, Dos, POP3 UIDL negative argument attempt, Encoded, 147.63.220.20, ->, 100.20.1.3, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 3, Escalation, OS, NETBIOS DCE/RPC ISystemActivator path overflow attempt little endian, Encoded, 100.20.1.3, ->, 100.70.2.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 4, Misc, Other, MISC BGP invalid type (0), Encoded, 100.70.2.1, ->, 100.70.3.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 5, Recon, Scanning, BAD-TRAFFIC loopback traffic, Encoded, 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 5, Recon, Scanning, BAD-TRAFFIC udp port 0 traffic, Encoded, 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 6, Intrusion, Root, TELNET APC SmartSlot default admin account attempt, Encoded, 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 7, Escalation, Service, NETBIOS SMB Session Setup AndX request unicode username overflow attempt, Encoded, 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 8, Intrusion, Root, NETBIOS SMB DCE/RPC Workstation Service bind attempt microsoft-ds, Encoded, 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 9, Goal, Backdoor, BACKDOOR Doly 1.5 server response, Encoded, 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.7, ->, SUCCESS

Scenario 5 Results

Attack 1, Step 1, Recon, Footprinting, SCAN nmap TCP, Encoded, 53.48.149.21, ->, 100.10.1.1, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 1, Recon, Footprinting, RPC portmap pcnfsd request UDP, Encoded, 42.206.52.244, ->, 100.10.1.1, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 1, Recon, Footprinting, RPC portmap sadmind request UDP, Encoded, 241.31.165.89, ->, 100.10.1.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 2, Intrusion, Other, WEB-ATTACKS kill command attempt, Encoded, 13.133.194.205, ->, 100.10.1.1, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 2, Intrusion, Other, INFO Connection Closed MSG from Port 80, Encoded, 147.63.220.20, ->, 100.10.1.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 3, Intrusion, User, WEB-MISC Tomcat servlet mapping cross site scripting attempt, Encoded, 100.10.1.1, ->, 100.60.2.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 4, Escalation, Service, EXPLOIT ebola USER overflow attempt, Encoded, 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 5, Intrusion, Other, WEB-PHP phpMyAdmin db_details_importdocsql.php access, Encoded, 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.7, ->, FAIL
Attack 1, Step 5, Intrusion, Other, WEB-PHP shoutbox.php directory traversal attempt, Encoded, 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 6, Goal, Dos, RPC mountd UDP umountall request, Encoded, 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 7, Goal, Backdoor, BACKDOOR MISC Linux rootkit satori attempt, Encoded, 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 8, Misc, Other, POLICY FTP RETR 1MB possible warez site, Encoded, 100.60.3.1, ->, 100.60.3.3, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 9, Goal, Dos, RPC mountd UDP umountall request, Encoded, 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 10, Goal, Pilfering, ORACLE create table attempt, Encoded, 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 11, Goal, Dos, POP3 DELE negative argument attempt, Encoded, 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 12, Misc, Other, POLICY FTP CWD possible warez site, Encoded, 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.7, ->, SUCCESS
Attack 1, Step 13, Goal, Dos, RPC mountd UDP umount request, Encoded, 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.7, ->, SUCCESS
Appendix G: Process Refinement Testing Setup - Multiple Attack Scenarios

This section shows the inputs to the simulation methodology used to create the multiple attack scenarios for testing Process Refinement. The resulting attack output from the simulation is also shown. The screen shots below shows the input for Multiple Attack Scenario 1 and the table summarizes the inputs for Scenarios 2-5.

Multiple Attack Scenario 1

![Automatic Attack Generation]

**Scenario Name:** Multiple Attack 1

Please specify the number of attacks to generate: 3

**Target IP:** 100.50.4.1

**Efficiency of Hacker (0-1):** 0.9

**Delay This Attack By (mins):** 10

**Average Delay Time Between Steps:** 10

**Percent of time to encode step (0-1):** 0

**Noise:**
- **Alerts Per Hour:** 100
- **Type of Noise:**
  - 100
  - **Percentage:**
  - **Time to Run After Last Attack is Complete:** 5

**Goal on Target:** Pilfering

**Stealth Level of Hacker (0-1):** 0.9

This is a measure of the hackers tendency to achieve goals on machines other than the target. More goals will occur on other machines besides the target with a low stealth score.

This is a measure of how quickly the hacker will penetrate and achieve the goal. A higher efficiency score with provide more direct attacks and higher success rates.
Simulation Inputs for Scenarios 2-5

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Attack 1</th>
<th>Goal</th>
<th>Efficiency</th>
<th>Stealth</th>
<th>Attack Delay</th>
<th>Step Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 2</td>
<td>100.40.2.1</td>
<td>Backdoor</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Attack 2</td>
<td>100.20.1.1</td>
<td>DoS</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Attack 3</td>
<td>100.60.3.2</td>
<td>DoS</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>100.40.3.3</td>
<td>Pilfering</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Attack 2</td>
<td>100.50.4.4</td>
<td>Backdoor</td>
<td>0.9</td>
<td>0.9</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Attack 3</td>
<td>100.60.3.2</td>
<td>Pilfering</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>100.80.3.1</td>
<td>DoS</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Attack 2</td>
<td>100.50.4.6</td>
<td>Backdoor</td>
<td>0.9</td>
<td>0.9</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>Attack 3</td>
<td>100.60.3.4</td>
<td>Pilfering</td>
<td>0.9</td>
<td>0.9</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>100.50.4.1</td>
<td>Backdoor</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Attack 2</td>
<td>100.30.2.2</td>
<td>Pilfering</td>
<td>0.9</td>
<td>0.9</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Attack 3</td>
<td>100.70.3.1</td>
<td>Pilfering</td>
<td>0.9</td>
<td>0.9</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Scenario 1 Results

Attack 3, Step 1, Recon, Footprinting, RPC portmap RQUOTA request UDP, , 237.85.139.21, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 1, Recon, Footprinting, WEB-IIS /StoreCSV/InstantOrder.asmx request, , 21.49.174.116, ->, 100.10.1.2, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 1, Recon, Footprinting, OTHER-IDS ISS RealSecure 6 event collector connection attempt, , 63.220.20.112, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, OS, NETBIOS DCERPC Messenger Service buffer overflow attempt, , 202.85.115.142, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, Other, WEB-PHP forum_details.php access, , 11.104.16.80, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 3, Escalation, Service, WEB-COLDFUSION exepeval access, , 100.10.1.2, ->, 100.40.2.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 4, Intrusion, Root, WEB-COLDFUSION cfmlsyntaxtaxcheck.cfm access, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 1, Misc, Other, POLICY FTP MKD / possible warez site, , 192.70.172.66, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, Other, WEB-IIS as_web4.exe access, , 120.191.192.102, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 5, Intrusion, Other, WEB-IIS ASP contents view, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 3, Escalation, OS, EXPLOIT SCO calserver overflow, , 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, Root, MS-SQL/SMB shellcode attempt, , 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Intrusion, User, WEB-CGI upload.pl access, , 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 5, Intrusion, User, WEB-CGI ksh access, , 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, FAIL
Attack 3, Step 4, Intrusion, Root, WEB-COLDFUSION exampleapp access, , 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Intrusion, User, WEB-CGI LWGate access, , 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt (ADMv2), , 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 5, Intrusion, Other, INFO FTP no password, , 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 6, Intrusion, Root, MS-SQL/SMB sp_password password change, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 6, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 7, Goal, Pilfering, POLICY FTP anonymous login attempt, , 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 7, Goal, Backdoor, BACKDOOR Doly 2.0 access, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 7, Escalation, OS, EXPLOIT nips x86 Solaris overflow, , 100.50.4.4, ->, 100.50.4.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 8, Escalation, Service, SMTP RCPT TO overflow, , 100.50.4.4, ->, 100.50.4.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 9, Goal, Dos, WEB-FRONTPAGE dwvssr.dll access, , 100.50.4.4, ->, 100.50.4.3, Prob of Success =, 0.9, ->, SUCCESS
Scenario 2 Results

Attack 3, Step 1, Recon, Footprinting, RPC portmap nlockmgr request TCP, , 6.139.234.110, ->, 100.20.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 1, Intrusion, Other, WEB-PHP forum details.php access, , 149.21.117.231, ->, 100.20.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 2, Escalation, Service, NETBIOS SMB Data Service Session Setup AndX request username overflow attempt, , 110.130.59.158, ->, 100.20.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 1, Misc, Other, WEB-CLIENT Javascript URL host spoofing attempt, Encoded, 162.161.110.25, ->, 100.20.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 3, Intrusion, Root, IMAP auth overflow attempt, , 100.20.1.1, ->, 100.70.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 4, Intrusion, Other, WEB-CLIENT RealPlayer arbitrary javascript command attempt, , 100.20.1.1, ->, 100.70.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 5, Escalation, Service, EXPLOIT ISAKMP forth payload certificate request length overflow attempt, , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 6, Intrusion, Other, WEB-PHP gallery arbitrary command execution attempt, , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Intrusion, Root, WEB-PHP read_body.php access attempt, , 17.16.203.97, ->, 100.20.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, Root, WEB-PHP rolis guestbook arbitrary command execution attempt, Encoded, 231.147.63.220, ->, 100.20.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 7, Goal, Dos, NETBIOS SMB DCE/RPC NTLMSSP invalid mechlistMIC attempt, , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 3, Goal, Dos, WEB-CLIENT Microsoft wmf metafile access, Encoded, 19.149.130.203, ->, 100.20.1.1, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 3, Goal, Dos, EXPLOIT ICQ SRV_MULTI/SRV_META_USER email overflow attempt, Encoded, 21.87.62.112, ->, 100.20.1.1, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 3, Goal, Dos, WEB-MISC negative Content-Length attempt, Encoded, 116.151.19.30, ->, 100.20.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 3, Misc, Other, POLICY FTP MKD possible warez site, , 27.160.20.159, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, Root, RPC portmap proxy integer overflow attempt UDP, , 44.167.1.193, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt, , 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Goal, Backdoor, BACKDOOR Satan'sBackdoor.2.0.Beta, , 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS

Scenario 3 Results

Attack 2, Step 1, Misc, Other, POLICY FTP MKD / possible warez site, , 136.23.194.103, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 1, Intrusion, Other, WEB-PHP b2 cafelog gm-2-b2.php remote command execution attempt, , 70.172.66.23, ->, 100.10.1.2, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 1, Recon, Enumeration, WEB-CGI fileseek.cgi access, , 27.85.33.1, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 1, Intrusion, Other, WEB-PHP b2 cafelog gm-2-b2.php remote command execution attempt, , 61.224.156.96, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, Service, NETBIOS SMB Data Service Session Setup AndX request username overflow attempt, , 67.67.46.89, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, Other, WEB-PHP forum_details.php access, , 4.110.103.71, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 3, Escalation, Service, MS-SQL xp_proxiedmetadata possible buffer overflow, , 100.10.1.2, ->, 100.40.2.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 4, Intrusion, Root, MS-SQL/SMB xp_enumresultset possible buffer overflow, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 3, Intrusion, Root, MS-SQL/SMB xp_reg* registry access, , 100.10.1.3, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 5, Intrusion, Other, SHELLCODE x86 stealth NOOP, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, User, WEB-IIS unicode directory traversal attempt, , 100.10.1.3, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt (ADMv2), , 100.40.2.1, ->, 100.40.3.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 6, Escalation, Service, FTP CEL overflow attempt, , 100.50.3.1, ->, 100.50.4.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 7, Intrusion, Other, FINGER remote command pipe execution attempt, , 100.50.3.1, ->, 100.50.4.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 8, Goal, Dos, WEB-FRONTPAGE dvwwsr.dll access, , 100.50.3.1, ->, 100.50.4.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 2, Escalation, OS, NETBIOS DCERPC Messenger Service buffer overflow attempt, , 121.2.109.74, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 9, Goal, Backdoor, BACKDOOR QAZ Worm Client Login access, , 100.50.3.1, ->, 100.50.4.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Goal, Pilfering, FTP RNFR ././ attempt, , 100.40.2.1, ->, 100.40.3.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 3, Intrusion, Root, WEB-PHP read_body.php access attempt, , 100.10.1.2, ->, 100.60.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 4, Intrusion, Other, WEB-PHP phpMyAdmin db_details_importdocsql.php access, , 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 5, Misc, Other, POLICY FTP RETR 1MB possible warez site, , 100.60.2.1, ->, 100.60.3.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 6, Escalation, OS, NETBIOS SMB DCERPC Remote Activation bind attempt, , 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 7, Escalation, Service, WEB-MISC changepw.exe access, , 100.60.3.3, ->, 100.60.3.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 8, Goal, Pilfering, ORACLE create table attempt, , 100.60.3.3, ->, 100.60.3.2, Prob of Success =, 0.9, ->, FAIL
Attack 3, Step 8, Goal, Pilfering, ORACLE sys.all_users access, , 100.60.3.3, ->, 100.60.3.2, Prob of Success =, 0.9, ->, SUCCESS

Scenario 4 Results

Attack 3, Step 1, Intrusion, Other, WEB-PHP forum_details.php access, , 237.85.139.21, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 1, Misc, Other, POLICY FTP MKD / possible warez site, , 192.70.172.66, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 1, Recon, Enumeration, FTP LIST directory traversal attempt, , 27.85.33.1, ->, 100.20.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, Service, WEB-MISC cwmail.exe access, , 56.114.61.224, ->, 100.20.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 2, Escalation, OS, NETBIOS DCE/RPC Messenger Service buffer overflow attempt, , 120.191.192.102, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 3, Intrusion, Root, WEB-MISC oracle portal demo access, , 100.10.1.2, ->, 100.60.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, Other, WEB-PHP forum_details.php access, , 129.36.132.247, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 4, Intrusion, Other, WEB-PHP shoutbox.php directory traversal attempt, , 100.60.2.1, ->, 100.60.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 3, Escalation, Service, MS-SQL xp_proxiedmetadata possible buffer overflow, , 100.10.1.2, ->, 100.40.2.2, Prob of Success =, 0.9, ->, FAIL
Attack 3, Step 5, Misc, Other, POLICY FTP CWD possible warez site, , 100.60.2.1, ->, 100.60.3.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 6, Escalation, OS, NETBIOS SMB DCE/RPC Remote Activation bind attempt, , 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 3, Escalation, Service, RPC snmpXdmii overflow attempt TCP, , 100.10.1.2, ->, 100.40.2.2, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 4, Escalation, Service, EXPLOIT x86 Linux mountd overflow, , 100.10.1.2, ->, 100.40.2.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 4, Intrusion, Root, FTP SITE EXEC attempt, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 5, Intrusion, Other, WEB-IIS ASP contents view, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 6, Escalation, OS, NETBIOS SMB DCE/RPC Remote Activation bind attempt, , 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 6, Escalation, Service, POP3 EXPLOIT x86 BSD overflow, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 7, Escalation, Service, SMTP MAIL FROM sendmail prescan too many addresses overflow, , 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 8, Goal, Pilfering, ORACLE create table attempt, , 100.60.3.3, ->, 100.60.3.4, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 7, Intrusion, Other, INFO FTP no password, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 3, Intrusion, Root, WEB-PHP rolis guestbook arbitrary command execution attempt, , 100.20.1.3, ->, 100.70.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 7, Intrusion, Other, WEB-CGI SWSoft ASPSeek Overflow attempt, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 7, Intrusion, Other, WEB-CLIENT readme.eml autoload attempt, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 7, Intrusion, Other, SHELLCODE x86 NOOP, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 8, Goal, Dos, ICMP Large ICMP Packet, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 9, Goal, Backdoor, BACKDOOR Infector.1.x, , 100.50.3.1, ->, 100.50.4.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, User, ; flow:to_server established; uricontent:, , 100.20.1.3, ->, 100.70.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Misc, Other, WEB-MISC nc.exe attempt, , 100.70.2.1, ->, 100.80.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Goal, Dos, SMTP XEXCH50 overflow attempt, , 100.70.2.1, ->, 100.80.3.1, Prob of Success =, 0.9, ->, SUCCESS

Scenario 5 Results

Attack 3, Step 1, Recon, Footprinting, WEB-CGI quickstore.cgi access, , 85.139.21.162, ->, 100.20.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 1, Recon, Footprinting, WEB-IIS /StoreCSVs/InstantOrder.asmx request, , 200.118.193.153, ->, 100.10.1.2, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 1, Recon, Enumeration, WEB-CGI fileseek.cgi access, , 192.70.172.66, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 1, Recon, Footprinting, RPC portmap rwalld request TCP, , 48.229.96.83, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, User, WEB-CGI simplestmail.cgi access, , 20.105.87.182, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 2, Escalation, Service, NNTP ihave overflow attempt, , 231.191.23.162, ->, 100.20.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, User, WEB-MISC VsSetCookie.exe access, , 205.144.52.53, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 3, Intrusion, Other, WEB-PHP PayPal Storefront arbitrary command execution attempt, , 100.20.1.2, ->, 100.20.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 4, Misc, Other, POLICY poll.gotomypc.com access, , 100.20.1.2, ->, 100.70.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 5, Escalation, OS, NETBIOS SMB DCERPC Messenger Service buffer overflow attempt, , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 3, Recon, Enumeration, WEB-IIS fpccount access, , 251.80.62.225, ->, 100.10.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 4, Intrusion, User, WEB-IIS unicode directory traversal attempt, , 26.14.140.245, ->, 100.10.1.1, Prob of Success =, 0.9, ->, FAIL
Attack 2, Step 4, Intrusion, User, WEB-IIS achg.htr access, , 16.215.16.39, ->, 100.10.1.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 6, Misc, Other, POLICY vncviewer Java applet download attempt, , 100.70.2.1, ->, 100.80.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 5, Intrusion, Root, MS-SQL/SMB xp_showcolv possible buffer overflow, , 100.10.1.1, ->, 100.30.2.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, OS, NETBIOS DCERPC Messenger Service buffer overflow attempt, , 29.167.230.60, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 7, Intrusion, User, WEB-CGI story.pl arbitrary file read attempt, , 100.70.2.1, ->, 100.70.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 3, Escalation, OS, EXPLOIT SCO calserver overflow, , 100.10.1.2, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, Root, MS-SQL/SMB xp_displayparamstmt possible buffer overflow, , 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 4, Intrusion, Root, MISC Insecure TIMBUKTU Password, , 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 6, Goal, Pillfering, FTP RNFPR ././ attempt, , 100.10.1.1, ->, 100.30.2.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Intrusion, User, NETBIOS SMB C$ access, , 100.40.2.1, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 8, Escalation, Service, WEB-CGI MDaemon form2raw.cgi access, , 100.70.2.1, ->, 100.70.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 9, Goal, Pilfering, MISC rsyncd overflow attempt, , 100.70.2.1, ->, 100.70.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt, , 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 7, Goal, Backdoor, BACKDOOR QAZ Worm Client Login access, , 100.50.3.1, ->, 100.50.4.1, Prob of Success =, 0.9, ->, SUCCESS
Appendix H: Single Attack - Process Refinement Recommendations

The following tables show the Process Refinement recommendations from the Process Refinement model during different stages of the attack. The tables are labeled such that the first number represents the attack scenario and the second number represents the step at which the recommendation occurs. For example, the table A1,1 shows the recommendation obtained during attack scenario 1 at attack step 1. A one in the table indicates that the information was obtained and a blank indicates that the information was not obtained.

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>A1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td></td>
<td>26</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1</td>
<td>27</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>28</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>1</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>1</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>1</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A2.1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A2.2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A2.3</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A2.5</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td></td>
<td>26</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1</td>
<td>27</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>28</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>1</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>1</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>1</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3.1</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0</td>
<td>20</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>2</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>3</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>4</td>
<td>24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A3.2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A3.4</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A3.5</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A3.6</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4,1</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A4,2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A4,3</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A5,1</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A5,2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A5,3</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix I: Multiple Attack – Process Refinement Recommendations

The following tables show the Process Refinement recommendations from the Process Refinement model during different stages of the attack. The tables are labeled such that the first number represents the attack scenario and the second number represents the point at which the recommendation occurred. For example, the table M1,1 shows the recommendation obtained during attack scenario 1 at the first action that occurred. The table M1,6 shows the recommendation for attack scenario 1 after the sixth action in time has taken place. A one in the table indicates that the information was obtained and a blank indicates that the information was not obtained.

<table>
<thead>
<tr>
<th>M0.0</th>
<th></th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1.1</th>
<th></th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1.3</th>
<th></th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1.6</th>
<th></th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td></td>
<td></td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td></td>
<td></td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td></td>
<td></td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td></td>
<td></td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td></td>
<td>25</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td></td>
<td>26</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1</td>
<td>27</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>28</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>1</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>1</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>1</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M2.1</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>25</td>
<td>2</td>
<td>13</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>26</td>
<td>3</td>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>27</td>
<td>4</td>
<td>15</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>28</td>
<td>5</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>29</td>
<td>6</td>
<td>17</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>18</td>
<td>30</td>
<td>7</td>
<td>18</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>31</td>
<td>8</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>32</td>
<td>9</td>
<td>20</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>33</td>
<td>10</td>
<td>21</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>34</td>
<td>11</td>
<td>22</td>
<td>34</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>23</td>
<td>35</td>
<td>12</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>24</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M2.3</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>25</td>
<td>2</td>
<td>13</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>26</td>
<td>3</td>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>27</td>
<td>4</td>
<td>15</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>28</td>
<td>5</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>29</td>
<td>6</td>
<td>17</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>18</td>
<td>30</td>
<td>7</td>
<td>18</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>31</td>
<td>8</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>32</td>
<td>9</td>
<td>20</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>33</td>
<td>10</td>
<td>21</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>34</td>
<td>11</td>
<td>22</td>
<td>34</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>23</td>
<td>35</td>
<td>12</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>24</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M2.14</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>25</td>
<td>2</td>
<td>13</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>26</td>
<td>3</td>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>27</td>
<td>4</td>
<td>15</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>28</td>
<td>5</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>29</td>
<td>6</td>
<td>17</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>18</td>
<td>30</td>
<td>7</td>
<td>18</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>31</td>
<td>8</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>32</td>
<td>9</td>
<td>20</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>33</td>
<td>10</td>
<td>21</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>34</td>
<td>11</td>
<td>22</td>
<td>34</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>23</td>
<td>35</td>
<td>12</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>24</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M2.15</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>13</td>
<td>25</td>
<td>2</td>
<td>13</td>
<td>25</td>
<td>3</td>
<td>13</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td>3</td>
<td>14</td>
<td>26</td>
<td>4</td>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td>5</td>
<td>15</td>
<td>27</td>
<td>6</td>
<td>15</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td>7</td>
<td>16</td>
<td>28</td>
<td>8</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td>9</td>
<td>17</td>
<td>29</td>
<td>10</td>
<td>17</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td>11</td>
<td>18</td>
<td>30</td>
<td>12</td>
<td>18</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19</td>
<td>31</td>
<td>13</td>
<td>19</td>
<td>31</td>
<td>14</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>20</td>
<td>32</td>
<td>15</td>
<td>20</td>
<td>32</td>
<td>16</td>
<td>20</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>21</td>
<td>33</td>
<td>17</td>
<td>21</td>
<td>33</td>
<td>18</td>
<td>21</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>22</td>
<td>34</td>
<td>19</td>
<td>22</td>
<td>34</td>
<td>20</td>
<td>22</td>
<td>34</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>23</td>
<td>35</td>
<td>21</td>
<td>23</td>
<td>35</td>
<td>22</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>24</td>
<td>36</td>
<td>23</td>
<td>24</td>
<td>36</td>
<td>24</td>
<td>24</td>
<td>36</td>
</tr>
</tbody>
</table>
### M3.1

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>23</td>
<td>23</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>24</td>
<td>24</td>
<td>1</td>
<td>24</td>
</tr>
</tbody>
</table>

### M3.4

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### M3.6

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### M3.7

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>23</td>
<td>23</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>24</td>
<td>24</td>
<td>1</td>
<td>24</td>
</tr>
</tbody>
</table>

### M3.18

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>16</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>18</td>
<td>18</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>19</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>23</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>23</td>
<td>23</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>24</td>
<td>24</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>6</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>9</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>12</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>6</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>9</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>12</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>6</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>9</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>12</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>6</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>9</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>12</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>6</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>9</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>12</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>5</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>6</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>7</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>8</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>9</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>10</td>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>12</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>36</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
<th>M</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>1</td>
<td>1</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>36</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix J: Process Refinement Failure

This section shows the simulation inputs for creating the three attacks that highlight Process Refinement’s weakness. The results from running the simulation under the parameters shown are also shown. The recommendations made by Process Refinement for this scenario are also shown.

**Attack Scenario Inputs**

<table>
<thead>
<tr>
<th>Attack Scenario Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automatic Attack Generation</strong></td>
</tr>
<tr>
<td><strong>Scenario Name:</strong> Process Refinement Failure</td>
</tr>
<tr>
<td><strong>Please specify the number of attacks to generate:</strong> 3</td>
</tr>
<tr>
<td>**Attack 1</td>
</tr>
<tr>
<td><strong>Target IP:</strong> 100.40.3.6</td>
</tr>
<tr>
<td><strong>Goal on Target:</strong> Pilfering</td>
</tr>
<tr>
<td><strong>Efficiency of Hacker (0-1):</strong> 0.9</td>
</tr>
<tr>
<td><strong>Stealth Level of Hacker (0-1):</strong> 0.9</td>
</tr>
<tr>
<td><strong>Delay This Attack By (mins):</strong> 5</td>
</tr>
<tr>
<td><strong>Average Delay Time Between Steps:</strong> 5</td>
</tr>
<tr>
<td><strong>Percent of time to encode step (0-1):</strong> 0</td>
</tr>
<tr>
<td><strong>Noise:</strong> Alerts Per Hour: 1000</td>
</tr>
<tr>
<td><strong>Type of Noise:</strong> Percentage: 100</td>
</tr>
<tr>
<td><strong>Time to Run After Last Attack is Complete:</strong> 5</td>
</tr>
<tr>
<td><strong>Done</strong></td>
</tr>
</tbody>
</table>
Simulation Results

Attack 1, Step 1, Recon, Enumeration, WEB-CGI fileseek.cgi access, , 121.66.161.139, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 1, Misc, Other, POLICY FTP MKD / possible warez site, , 136.23.194.103, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 2, Intrusion, Other, WEB-PHP ttCMS header.php remote command execution attempt, , 71.107.120.17, ->, 100.10.1.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 3, Escalation, Service, WEB-FRONTPAGE dwvssr.dll access, , 100.10.1.2, ->, 100.40.2.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 4, Intrusion, Root, MS-SQL/SMB xp_setsqlsecurity possible buffer overflow, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 2, Escalation, Service, NNTP sendsys overflow attempt, , 50.198.18.64, ->, 100.10.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 5, Intrusion, Other, WEB-IIS ASP contents view, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, FAIL
Attack 1, Step 3, Intrusion, Root, WEB-FRONTPAGE orders.txt access, , 100.10.1.3, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 5, Intrusion, Other, SHELLCODE x86 stealth NOOP, , 100.40.2.2, ->, 100.50.3.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 6, Escalation, Service, SMTP RCPT TO overflow, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 4, Intrusion, User, WEB-CGI redirect access, , 100.10.1.3, ->, 100.40.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 5, Misc, Other, DNS EXPLOIT x86 Linux overflow attempt (ADMv2), , 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 7, Intrusion, Other, FINGER remote command pipe execution attempt, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 1, Step 6, Goal, Pilfering, POLICY FTP anonymous (ftp) login attempt, , 100.40.2.1, ->, 100.40.3.6, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 1, Recon, Enumeration, WEB-PHP phpbb quick-reply.php access, , 70.172.66.23, ->, 100.20.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 8, Goal, Dos, DDOS mstream agent to handler, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, FAIL
Attack 3, Step 2, Intrusion, User, WEB-IIS as_web4.exe access, , 160.244.138.131, ->, 100.20.1.3, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 8, Goal, Dos, DDOS TFN client command LE, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 2, Step 9, Goal, Dos, DOS ath, , 100.50.3.1, ->, 100.50.4.5, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 3, Escalation, OS, NETBIOS DCERPC ISSystemActivator path overflow attempt little endian, , 100.20.1.3, ->, 100.70.2.1, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 4, Escalation, OS, NETBIOS SMB DCERPC Messenger Service buffer overflow attempt, , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, SUCCESS
Attack 3, Step 5, Goal, Backdoor, BACKDOOR DeepThroat 3.1 Connection attempt [4120], , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, FAIL
Attack 3, Step 5, Goal, Backdoor, BACKDOOR FsSniffer connection attempt, , 100.70.2.1, ->, 100.80.3.2, Prob of Success =, 0.9, ->, SUCCESS
Process Refinement Recommendations

<table>
<thead>
<tr>
<th></th>
<th>F0</th>
<th></th>
<th>F1</th>
<th></th>
<th>F3</th>
<th></th>
<th>F4</th>
<th></th>
<th>F5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>25</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td>2</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td>3</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td>4</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td>5</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td>6</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>31</td>
<td>7</td>
<td>1</td>
<td>19</td>
<td>31</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>32</td>
<td>8</td>
<td>1</td>
<td>20</td>
<td>32</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>21</td>
<td>33</td>
<td>9</td>
<td>1</td>
<td>21</td>
<td>33</td>
<td>9</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>22</td>
<td>11</td>
<td>1</td>
<td>23</td>
<td>12</td>
<td>24</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>23</td>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>F3</th>
<th></th>
<th>F4</th>
<th></th>
<th>F5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
<td>S1</td>
<td>S2</td>
<td>M</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>25</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>26</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>27</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>28</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>29</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>30</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>31</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>32</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>33</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>24</td>
</tr>
</tbody>
</table>
Appendix K – Simulation and Process Refinement Files Stored on CD

This section contains a CD with files for the simulation methodology and the Process Refinement model. The files on the CD are described below.

Cyber Attack Simulation Methodology Files:

- Folder Entitled “Cyber Attack Simulation”
  - Cyber Attack Simulation model file without network setup (Cyber Attack Simulation.doe)
    - Can be used to create a simulated network and run attacks
  - Cyber Attack Simulation input files (ListOfActions.txt; SnortAlertDefs.txt; DragonAlertDefs.txt; SnortPriorityDefs.txt; HTTPInspectDefs.txt; ActionsInStages.txt; GuidanceTemplate.txt)
    - Used to setup the attack actions possible, alert definitions, and automatic attack generation necessary inputs
  - Network Simulation template development file (Network Simulation.tpl)
    - Used to make edits, additions, or deletions from the Network Simulation template
  - Network Simulation template execution file (Network Simulation.tpo)
    - Used to open the Network Simulation template in the Arena software

- Folder Entitled “Example”
  - Cyber Attack Simulation model file with a sample network created (Cyber Attack Simulation Example.doe)
    - Can be used to run attack scenarios
- Cyber Attack Simulation input files (ListOfActions.txt; SnortAlertDefs.txt; DragonAlertDefs.txt; SnortPriorityDefs.txt; HTTPInspectDefs.txt; ActionsInStages.txt; GuidanceTemplate.txt)
- Network Simulation template development file (Network Simulation.tpl)
- Network Simulation template execution file (Network Simulation.tpo)
- Example Attack Scenario Input File (InputFile_Attack Example.txt)
  - Can be used to load a previously generated attack scenario

**Process Refinement Files:**

- Folder Entitled “Process Refinement”
  - Process Refinement OPL project file (Process Refinement Project.prj)
    - Used to load model files and data files in OPL
  - Process Refinement OPL model file (Process Refinement.mod)
    - Base optimization model that can have data inputs

- Folder Entitled “Example”
  - Process Refinement OPL project file (Process Refinement Project.prj)
  - Process Refinement OPL model file (Process Refinement.mod)
  - Process Refinement OPL data file with variable input blank (No Variable Input.dat)
    - A data file that can be used with the Process Refinement OPL model but needs to have the variable input information specified
  - Process Refinement OPL data file with variable input specified for an example (Variable Input Specified.dat)
- A data file that can be used with the Process Refinement OPL model to obtain a recommendation