PETRI NET BASED MODEL FOR PROTOCOL DAMAGE DETECTION AND PROTECTION

By

ADITI GODSE

Bachelor of Science in Information Technology

Rajiv Gandhi University

India

2004

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2007

PETRI NET BASED MODEL FOR PROTOCOL

DAMAGE DETECTION AND PROTECTION

Thesis Approved:

Dr. Johnson Thomas

Thesis Advisor

Dr. N. Park

Dr. John. P. Chandler

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGEMENT

I sincerely thank my advisor Dr. Johnson P. Thomas, for his encouragement, guidance and support he provided throughout the entire duration of my thesis. I would also like to thank my committee members Drs. Nohpill Park and John P. Chandler for serving on my committee.

The Computer Science Department, staff and faculty alike deserve special mention for the continual assistance in the technical areas that I required throughout the entire period of my education here.

TABLE OF CONTENTS

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER I

INTRODUCTION

Ad Hoc wireless networks have greatly increased information sharing and accessing amongst the general public, but these networks do not have any infrastructure and have limited resources. Due to their infrastructureless characteristics and limited resources these networks face two difficult challenges:

- Providing quality of service
- Securing these networks as they are vulnerable to attacks

Today, many efficient routing protocols exist for Ad Hoc wireless networks. However, it is very difficult to measure the quality of service of security provided by existing protocols. In [7] the author has proposed a finite state machine approach to measure the quality of service or security in relation to another protocol. Although this is not an absolute measure, it is a useful indicator of the quality of service or security provided by one protocol in comparison with another protocol.

In this thesis we propose to extend this approach by modeling protocols as Petri Nets. Petri nets provide several advantages over finite state machines. In particular, Petri Nets can indicate the current state of the protocol as defined by the distribution of tokens in the net. Secondly Petri Nets allow us to reason about the net and derive useful properties of the system.

In our proposed approach we look at four key issues:

- To define a measure for the matching of two nets. Here one net is protocol petri net and the other one is the attack petri net. This is achieved by determining closest match between the states and transitions of the two Petri nets such that the attacker's goal is reachable.
- To find similar markings in protocol petri net model and attack petri net. This will ensure the existence of vulnerable states which will be of benefit to the attacker and cause maximum damage to the user.
- To determine how much damage an attack can cause to the protocol. We analyze what resources, i.e. data and actions, are exposed to the intruder, and the sensitivity of the information compromised. Based on this analysis we assign the cost of damage to the user.
- Once the damage is estimated, various protective measures are implemented to protect the protocol from an attack by reducing or eliminating the number of exposed vulnerable states and resources.

The proposed model is divided into three main phases.

- Modeling. The purpose of modeling is to be able to define petri net models for network protocols and attacks and further identify their properties and reason about them. Here we model the following:
  - Petri net model of protocol
  - Petri net model of attack
  - Determine the reachability graph of the attack Petri net

- Categorization of resources in protocol model. Each resource identified in protocol and attack net is categorized into different abstract dimensions, which define an attack surface.
- Pay off function assigns costs to the categories. The payoff function helps determine the maximum matching of states in protocol net with states of attack net.

➢ Damage Estimation: The objective of this phase is to find the states in the protocol petri net that are of maximum benefit to the attacker.

- Matching of protocol and attack petri nets.
- Yields common states and resources. The most vulnerable states and resources in protocol petri net are obtained.
- Determine the damage for the substructure using pay off function and categories. Categories are determined by classifying the resources based on their properties. Some resources are targets of attack while some are enablers that just enable an attack whereas others are carriers that propagate the attack to the targeted resource. A detailed description is given in chapter 4.

➢ Protection. The objective here is to protect the protocol against intrusions by defining two protective measures:

- Changing the access rights of the vulnerable resources.
- Changing the order of availability of resources.

The rest of the thesis is outlined as follows:

In chapter 2, we present a literature review of previous work done in area of using both Petri nets and FSM to represent protocols. We also review the work done in the field of measurement of security of protocols using colored petri net models. In chapter 3, we provide a brief introduction to the ordinary petri nets and their analysis. In chapter 4, we provide the details of the proposed model for modeling, damage estimation and protection of the network protocol. We model petri nets for an ad hoc network protocol along with the attack that can be mounted on this protocol. This chapter also provides the various attacks possible on the physical layer, transport layer along with suitable defense mechanisms identified. In chapter 5, we provide the details of the damage estimation tool that matches the protocol petri net and attack petri net.

CHAPTER II

REVIEW OF LITERATURE

This chapter provides an overview of the key concepts involved in measuring the security of ad hoc network protocols using finite state machines as well as Petri nets. This chapter illustrates the previous work undertaken and completed to understand the metric of an attack surface. The concept- attack surface, attack dimensions are made use of in subsequent chapters. The chapter provides a concise description of the previous work attempted in the measuring the security of protocols using different methods.

2.1 Ad hoc Network Protocols

An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. Their nodes have only limited bandwidth and limited battery power. Due to the limited transmission range of wireless network interfaces, multiple network "hops" may be needed for one node to exchange data with another across the network. In recent years, a variety of new routing protocols targeted specifically at this environment have been developed, but little performance information on each protocol and no realistic performance comparison between them is available.

Ad hoc network routing can be classified into proactive and reactive ones. The former continuously makes routing decisions so that routes are immediately available when packets need to be transmitted with no regard as to when and how frequently such routes are desired. An example protocol is the Destination Sequence Distance Vector (DSDV) protocol.

The reactive protocols determine routes on an on-demand basis; when a node has packet to transmit, it queries the network for a route. Example protocols include the Ad hoc On-Demand Distance Vector (AODV) and the Secure Ad hoc On-Demand Distance Vector (SAODV) protocols.

2.2 Measurement of security of Protocols using Finite State Machines

Some work has been done on the performance comparison of ad hoc protocols based on metrics such as routing overhead, packet delivery ratio, end-to-end delay, path optimality, and throughput. Perkin et. al. [12] focused on only comparing the two on-demand routing protocols which are AODV and DSR. In other research, like [2], Broch et. al. extended the ns-2 simulator to model the behavior of the IEEE 802.11 wireless LAN standard and the wireless transmission channel. They used their ns-2 extension to compare the packet delivery ratio, routing overhead and path optimality of DSDV, TORA, AODV and DSR.

Our work for objective-II of matching the two petri nets based on resource availability and access rights is an extension to the theoretical work on defining security of Ad-Hoc network protocols and QoS Surfaces. The authors in [4] have theoretically defined a metric to determine whether one routing protocol of Ad Hoc network is

relatively more secure than another. They have counted the network's attack opportunities and used this count as an indication of the network's attackability (likelihood that it will be successfully attacked). The network's attack surface is described along four abstract dimensions: attack targets, attack triggers, attack carriers, and access rights. Intuitively, the more exposed the attack surface, the more likely the network could be successfully attacked, and hence the more insecure it is. Thus, one way to improve network security is to reduce its attack surface. Based on the metric suggested in [14] we propose to design a solution such that given a network protocol, it automatically determines how secure the protocol is and how much QoS it provides.

According to the existing research in [7] the wanderer protocol is a simple protocol that enables us to deploy the idea of an attack surface on ad hoc networks. Each node maintains a list of neighbors available and not the whole network topology. Although packets are broadcast, one neighbor is chosen from the list as the receiver and only that node retransmits the received packet. A random choice of a node is made from the list of neighbor nodes to be designated as the receiver. This random choice is governed by the random function which is usually application-oriented.

A node may receive the same packet many times; each node keeps track of the number of times it has forwarded a particular packet. A number, called the duplicate-forwarding parameter, determines the number of times a packet can be forwarded by a node. This duplicate forwarding parameter is set by the system design specifications. The intrinsic working details of the wanderer protocol are represented as a state transition diagram of the protocol.

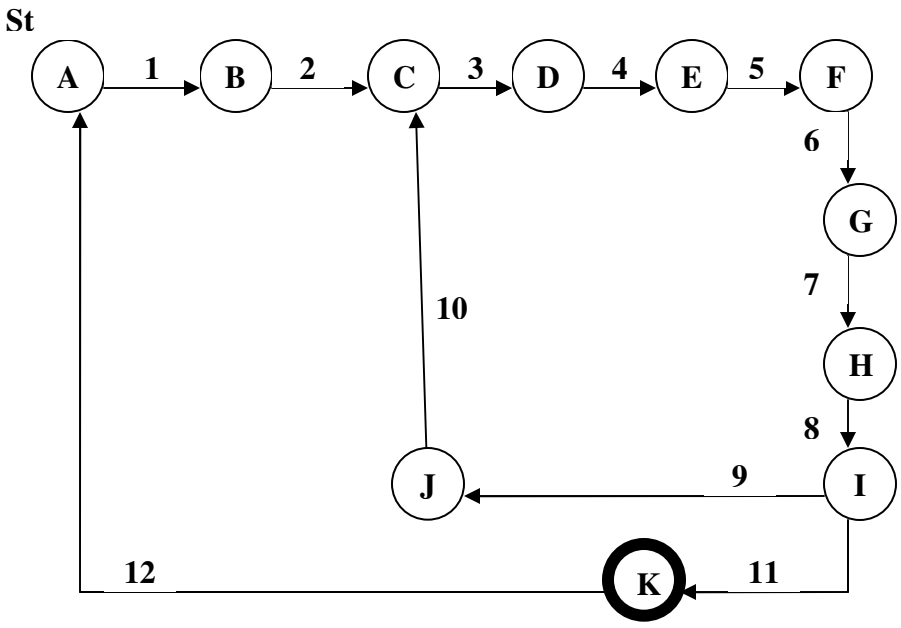

Figure 1: State Transition Diagram for Wanderer Protocol in [7].

State A represents the initial state of the network. No transmissions take place in this state. The network just waits for some events to take place so that transmission of data packets is necessary to report the events. At stage B, the request message from the base station is received by the leaf node. This stage also corresponds to the necessity of leaf nodes having to transmit data packets to report the occurrence of events to the base station. At Stage C a leaf node has the required information for transmission. The information to be transmitted will be suitably wrapped into data packets in the following stages, which will be interpreted by the base station as the occurrence of certain events. Stage D corresponds to the availability of the neighbor list in the leaf node. A neighbor list is required so that the 'receiver' can be chosen to be the next node in the routing path. E: This stage is responsible for packing the information requested into packets. These packets can be interpreted by the base station to correspond to various scenarios and

events.

The receiver is determined using the random function from the neighbor list. The policy used in the wanderer protocol is (Source ID + Receiver ID + Packet). At stage G the ID of the source leaf node, ID of the receiver chosen using the random function are attached to the packet so that only the receiver can transmit the packets to its neighbors. At stage H, the packet is transmitted to all the neighbor nodes present in the neighbor list. At next stage, I, the source node ID is stripped from the packet so that the receiver for the next stage of transmission will not be the source node. The receiver has the packet ready for further transmission. All the IDs which were initially packed with the data packet are stripped off at stage J. Thus the packet is ready for transmission to the next node on the routing path with- (Source ID +Intermediate ID+ new Receiver ID + Packet).

K is the final stage of the routing protocol, where the required packet has been received at the base station. After the packet has been received at the base station, the network enters the initial state A.

2.3 Measurement of security of Protocols using Petri-Nets

There has been some research done in comparing the performance of Ad-Hoc network protocols under varying network scenarios. Xiong et al. [12] proposed a topology approximation (TA) mechanism to address the problem of mobility and perform simulation of AODV routing protocol based on Petri nets. In [12] a stochastic Petri net based approach to model and analyze ad hoc network is suggested. Zhang and Zhou [3] present a SPN model of ad hoc networks and analyze its performance as a function of the

successful delivery ratio. In [13] an analysis of performance of TCP over MANET is done using fuzzy timing high level Petri nets.

Others have used colored Petri nets in the development of protocols for Ad hoc networking. In a joint research project [13] between the Centre for Pervasive Computing and Ericsson Telebit A/S, formal description techniques in the form of Colored Petri nets (CP-nets or CPNs) have been used in the development of protocols for ad-hoc networking based on Internet Protocol v6 (IPv6) . CPN model specifying Edge Router Discovery Protocol (ERDP) has been developed and integrated. The use of message sequence charts and simulation have been used to analyze and investigate its behavior in detail. State spaces have been used to verify important properties of the protocol.

In [10] Aly has generated a model using CP-Nets to show how it can be used to analyze security protocols. It is based on generating an explicit CP-Nets specification for the protocol and identifying insecure states that may or may not occur and performing a backward state analysis to test if each insecure state is reachable or not.

Summary

The approach of representing protocols in petri net form and creating an attack surface to determine features of a protocol can be extended further. This approach can be verified to provide a feasible solution to detect vulnerabilities in a system, and hence enhance the security associated with the system.

CHAPTER III

PETRI NETS

In this section, we give a brief introduction to ordinary Petri net theory through some standard definitions and examples from [9].

A Petri net is a graphical and mathematical modeling tool and provide a framework for modeling and analyzing both the performance and the functionality of distributed and concurrent systems. It consists of *places*, *transitions*, and *arcs* that connect them. *Input arcs* connect places with transitions, while *output arcs* start at a transition and end at a place. There are other types of arcs, e.g. *inhibitor arcs*, which we will use in our protection model. An inhibitor arc from place *p* to transition *t* disables *t* in any marking where *p* is not empty. Graphically, an inhibitor arc connects a place to a transition and is drawn with a small circle instead of an arrowhead. Since, one use of inhibitor arcs is to model limited resources; we use them in our protection model to limit the number of resources available to the attacker. Places can contain *tokens*; the current state of the modeled system (the *marking*) is given by the number (and type if the tokens are distinguishable) of tokens in each place. Transitions are active components. They model activities which can occur (the transition *fires*), thus changing the state of the system (the marking of the Petri net). Transitions are only allowed to fire if they are *enabled*, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places). When the transition fires, it removes

tokens from its input places and adds some at all of its output places. The number of tokens removed / added depends on the cardinality of each arc.

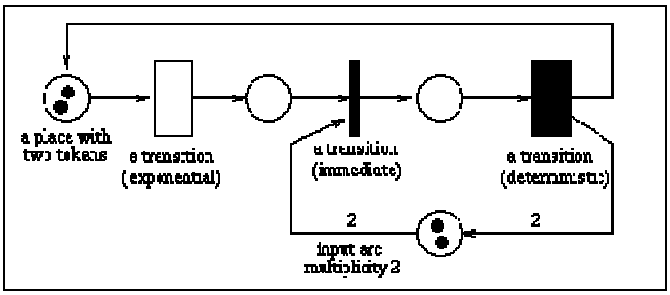

Figure 2: Example of a Petri Net

The net can be represented as a 5 tuple (P, T, F, Min, l) where:

- P is a set of places,

- T is a set of transitions, where P intersection T =∅,

- F is a $\subseteq ((P \times T) \cup (T \times P)$ is the flow relation, where T is $\subseteq$ range (F),

- $M_{in} \subseteq P$ is the initial marking,

- *l* is called a labeling function.

Definition 1*: A marking m of a* PN (Petri Net) is a function m: P→ N. It gives the number of tokens contained in each place p∈P. A token can be represented by *a dot.* The initial state of a system is defined by *a* net marking called the *initial marking.* Figure 3a shows a PN with its initial marking:

M_0 (Pl) = M_0 (P2) =1, M_0 (P3) = O.

The marking can be more briefly expressed as *a* column vector: $M_0 = (1\ 1\ 0)^T$.

Definition 2: A transition t *is **enabled*** wrt a marking *m* iff: for all p∈P, m(p) ≥I (p, t).

In figure 4a, only transition t_1 is enabled, and t_2 in figure 4b.

Definition 3 ***(execution rule) :*** Firing an enabled transition consists of removing I(p, t) tokens from each input place p and adding O(p, t) tokens to each output place p. Figure 5 shows the marking of the PN after firing the enabled transition t_1. The marking reached is $M_\ell = (0\ 0\ 1)^T$; in general, $\forall\ p \in P$, $M_\ell(p) = M_0(p) + O(p, t) - I(p, t)$.

Let M be the marking reached from M_0 by applying the firing sequence *s*. If y is the count vector of s (i .e. y represents the number of times each transition has been fired in s), then m can be expressed by the state equation $M = M_0 + Ay$,

where $A = O - I = [a(p, t)]$, $a(p, t) = O(p, t) - I(p, t)$.

A is called the ***incidence matrix*** of the Petri net in example of figure 3, And $s = t_1\ t_2\ t_1$, i.e., $y = (2\ 1)^T$, leads to the marking $m = mo + Ay = (0\ 0\ 1)^T$.

$$A = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figure 3: Incidence Matrix of Petri net [8]

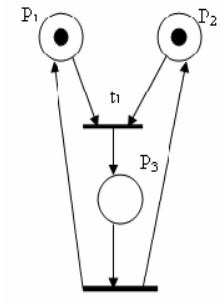

Figure 4: Marked Petri Net before firing transition t1 [8].

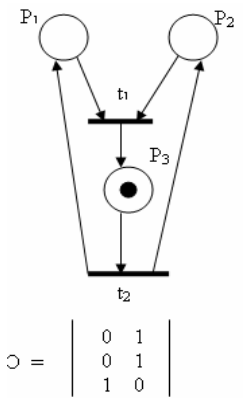

Figure 5: Marked Petri Net after firing transition t1 [8].

Definition *4 (reachability set) :* The reachability set *R(m)* for a PN with marking m is the set of all markings of PN which can be reached from m by firing a finite number of transitions of PN .

The reachability set of the PN in figure 3 is R(mo)=(1 10) ^{T}, (0 0 1) ^{T}) .

Analysis of Petri Nets

The following are some of the properties and questions that have been studied in the literature about Petri nets. Later we will use these properties to analyze the behavior of the Petri net model for protocols and attacks.

1) A deadlock in a PN occurs when a marking is reached where no transitions in the net can be fired from that point on.

2) A PN is live with respect to a marking m if, for any marking in R(m), it is possible to fire any transition in the net. Liveness guarantees the absence of deadlocks.

3) A PN is reversible or proper with respect to a marking m if for every m'∈R(m), m∈R(m'). Reversibility guarantees that the system modeled by the PN can re-initialize

itself. This is very important for automatic error recovery. It is easy to show that the PN in figure 3 is live and reversible. Therefore it is deadlock-free.

4) A PN is bounded with respect to a marking m if there exists a finite number k such that for any marking in R(m) the number of tokens in each place of the PN under that marking is less than k. When k=1, the PN is safe.

5) A PN is (structurally) conservative if, for any initial marking m 0, there exists a weighting vector w>0 Є N, n=|P| and for all m∈R(m_o), w^T m=w^T m_o i.e., the sum of the tokens weighted by w is constant. The weighting vector w is called *a P-invariant* or *S-invariant of* the PN. It can be proved from the state equation that w is an P-invariant of a PN iff w^T A=O where A is the incidence matrix of the PN.

6) A PN is (structurally) consistent if, for any initial marking m, there exists a firing sequence s, called a cyclic firing sequence with respect to m, such that m –s→*m. The count vector of a cyclic firing sequence of a PN is called a *T-invariant* of the net. Similarly, from the state equation, a T-invariant x of a PN must satisfy equation: Ax=O

For the PN defined in figure 3, the P-invariants are (n 1 n2 n1 + n2) ^{T}, n_1, n2>0, n_1 n_2#0. The net is obviously safe and bounded. The P-invariants are (n n) ^{T}, n>O.

CHAPTER IV

METHODOLOGY

Protocols consist of a number of independent concurrent protocol entities that may proceed in many different ways depending on, for example, when packets are lost, timers expire, and processes are scheduled. Protocol consists of resources which are either system resources or data resources. A resource can be a hardware entity like printer, user ID, data packet, password etc.

This makes design and specification of protocols a difficult task. In order to address the problem of security of protocols, a number of approaches such as key management techniques and cryptographic approaches to name a few have been proposed to improve the security.

To improve the security of protocol, we propose a model that first determines the vulnerability present in protocol. We propose a model to dynamically determine if the protocol can be attacked and propose a protective approach for improving security. Our model is divided into three phases namely: modeling, damage estimation, and protection.

A main motivation for the use of petri nets in representing protocols and attacks is the possibility to formally state and decide certain desirable properties, such as liveness (i.e., the system can never lock up) and boundedness (i.e., loosely, the system cannot accumulate indefinitely large amounts of work). The major advantage of using Petri nets is the short description and analysis of the systems consisting of a number of different processes having a similar structure and behavior.

- Modeling. The purpose of modeling is to be able to define petri net models for network protocols and attacks and further identify their properties and reason about them. Here we model the following:
  - Petri net model of protocol
  - Petri net model of attack
  - Determine the reachability graph of the attack Petri net
  - Categorization of resources in protocol model. Each resource identified in protocol and attack net is categorized into different abstract dimensions, which define an attack surface.
  - Pay off function assigns costs to the categories. The payoff function helps determine the maximum matching of states in protocol net with states of attack net.
- Damage Estimation: The objective of this phase is to find the states in protocol petri net that are of maximum benefit to the attacker.
  - Matching of protocol and attack petri nets.
  - Yields common states and resources. The most vulnerable states and resources in protocol petri net are obtained.
  - Determine the damage for the substructure using pay off function and categories
- Protection. The objective here is to protect the protocol against intrusions by defining two protective measures:
  - Changing the access rights of the vulnerable resources.
  - Changing the order of availability of resources.

4.1 Modeling

4.1.1 Petri net model of protocol

Modeling of protocols in Petri net form has following advantages:

1. Is easy to use and understand

2. Is a powerful modeling and analysis tool

3. Can be automated

4. Shares common properties (boundedness, liveness, completeness, proper termination and deadlock-freeness) with protocols.

5. Can be easily transformed into state transition diagrams.

The petri net model of any protocol can be represented as follows:

- Places: States of protocol are represented by places. A change in place means that either a new resource is added to the mapping, a resource was deleted, or a resource changes in value. We assume each place-transition is atomic. An execution of a petri net is the alternating sequence of places and transitions executions: $P_0 \times T_1 \rightarrow P_1 \times T_2 \rightarrow \ldots\ldots\ P_{i-1} \times Ti \rightarrow P_{i\ldots}$
- Transitions: activities or actions in protocol are represented by transitions.
- Tokens: tokens are the resources, i.e. processes and data that are available in the network, for example, Message, User name, data, password etc.

Wanderer Protocol

The wanderer protocol has been chosen to illustrate the working of our model. The wanderer protocol is a simple protocol that enables us to deploy the idea of petri net based model to detect the protocol's vulnerability towards an attack.

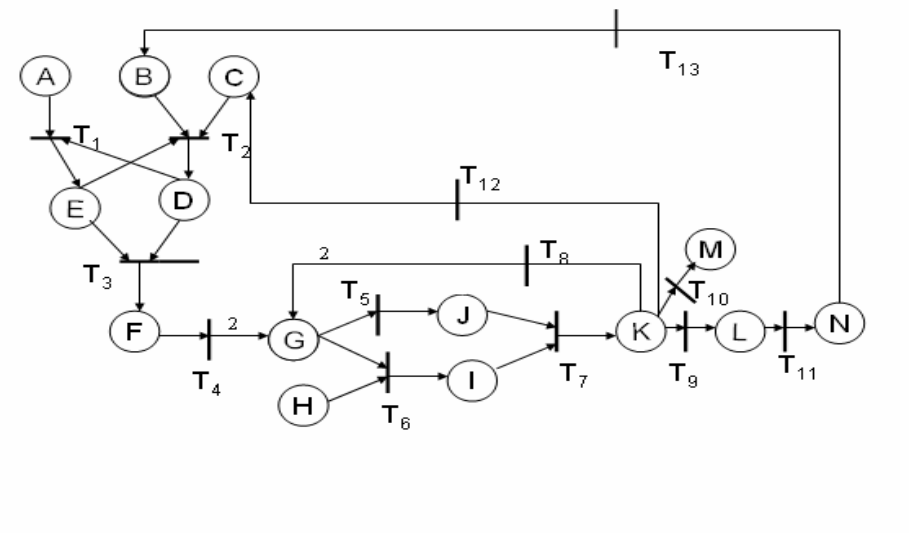

Figure 6: Petri Net model of Wanderer Protocol.

- $P_{Wanderer}$ = {Set of states in the protocol}

  = {A, B, C, D, E, F, G, H, I, J, K}

- $T_{Wanderer}$ = {Set of actions that take a resource from one state to another}

  = {T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9}

- Tokens: tokens represent the resources, i.e. processes and data of the protocol, i.e. User name, password, data, message, sender ID, receiver ID.

Working details of the wanderer protocol as described as below:

States:

- **A:** This is the state which represents that a data is ready. No transmissions take place in this state. The sensor network just waits for some events to take place so that transmission of data is necessary to report the events.
- **B:** The leaf nodes are waiting in this state for a request message from a base station.

- **C:** This state represents a base station. Upon receiving a message from the base station the leaf node enters the next stage to transmit the requested information through transition T2.
- **D:** This is the stage which represents that the request message from the base station has been received by the leaf node.
- **E:** The information to be transmitted will be suitably wrapped into data packets in transition T1 and data packets are available at stage E.
- **F:** The source node information, i.e. node ID is added to the packet to prevent the looping of packet between two nodes.
- **G:** At this stage, the neighbor list is available in the leaf node. A neighbor list is required so that the 'receiver' can be chosen to be the next node in the routing path.
- **H:** A random function is used to determine the receiver from the neighbor list. Care is taken so that the sender is not again chosen to be the receiver else the packet may just loop across the two nodes without traversing in the forward direction.
- **I:** Once the Receiver is determined, its Receiver ID is also retrieved and stored at this place so that it can be attached to the data packet.
- **J:** The (Source ID +Data packet) is available at this place to be attached to the Receiver ID.
- **K:** The ID of the receiver chosen using the random function is stored along with the Source Id and Data packet. This ID is added to the packet so that only the receiver can transmit the packets to its neighbors. This is the policy used in the wanderer protocol. (Source ID + Receiver ID + Packet). The packet is transmitted to all the neighbor nodes

present in the neighbor list. Each neighbor node receives the transmitted packet but confirms its identity as the receiver before transmitting the packet again.

- **L:** The source node ID is stripped from the packet so that the receiver for the next stage of transmission will not be the source node. The receiver node is taken from the packet so that only the receiver node has the packet ready for the next stage of transmission.
- **M:** The receiver has the packet ready for further transmission. All the IDs which were initially packed with the data packet are stripped off. Thus the packet is ready for transmission to the next node on the routing path. The packet is now added with the original source ID, the current node ID, the new Receiver node ID.

  (Source ID +Intermediate ID+ new Receiver ID + Packet).
- **N:** This is the final stage of the routing protocol where in the sensor network, the required packet has been received at the base station. After the packet has been received at the base station the network enters the initial state C.

Transitions:

$T_{1:}$ The information to be sent is collected and wrapped into data packets.

T_2: Upon receiving a message from the base station the leaf node enters the next stage to transmit the requested information.

T_3: The Source ID of the leaf node and data packet are attached together.

$T_{4:}$ Neighbor node list is created and updated.

T_5: The Source ID and Data packet is made available at the next state.

T_6: Using random function over the neighbor node list, a receiver is determined.

T_7: Receiver ID is attacked to the Source ID and Data packet, so that only the receiver can transmit the packets to its neighbors. This is the policy used in the wanderer protocol.

(Source ID + Receiver ID + Packet).

T_8: The data packet is transmitted to all the neighbors of the transmitting node.

T_9: Check if the current node is the receiver node ID

T_{10}: Attach the node's ID as the current intermediate node ID

T_{11}: If the destination is the base station then the sensor network reaches its final state

T_{12}: Timeout makes the base station, request for information from the sensor node

T_{13}: The information is ready for transmission

4.1.2 Petri net model of attack

Similarly, we model the attack as a petri net. An attack is modeled as a sequence of executions of actions that ends in a state that satisfies the adversary's goal. The petri net model of any attack can be represented as follows:

- Places: states of attack are represented by places. A change in place means that either a new resource is deleted, or a resource value is changed. We assume each place-transition is atomic. An execution of a petri net is the alternating sequence of places and transitions executions: $P_0 \times T_1 \rightarrow P_1 \times T_2 \rightarrow \ldots\ldots\ P_{i-1} \times Ti \rightarrow P_{i\ldots}$
- Transitions: activities or actions in attack are represented by transitions.
- Tokens: tokens are the resources, i.e. processes and data that the intruder aims to attack or use for an attack, for example, unauthenticated message, User name, unencrypted data and password etc.

Sinkhole attacks

In a sinkhole attack, the adversary lures all traffic from a particular area to a compromised node, creating a metaphorical sinkhole with the adversary at the center. Sinkhole attacks can enable many other attacks (selective forwarding, for example).

In sinkhole attacks the compromised nodes are made to act as sink holes so that the data packets are dropped or selectively forwarded to avoid suspicion. Over a period of time all the nodes in the network can be compromised to bring down the whole network. The compromised nodes are present as different nodes in the neighbor list thus affecting the normal functioning of the random function. Resources required to launch sinkhole attack:

- Unencrypted Message sender ID– Attack looks for nodes that transmit data and have unencrypted ID. If this ID is available and is unencrypted

- Unencrypted data – Attacks looks for a node that has unencrypted data.

- Message receiver – Node that receives the message.

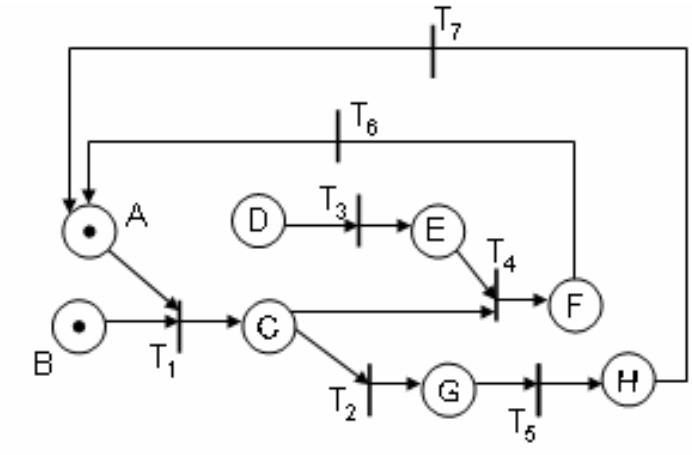

Figure 7: Sinkhole Attack Petri net Model

- $P_{sinkhole}$ = {Set of states in the attack}

  = {A, B, C, D, E, F, G, H}

- $T_{sinkhole}$ = {Set of actions that take a resource from one state to another}

  = {T_1, T_2, T_3, T_4, T_5, T_6, T_7}

- Tokens: tokens represent the resources, i.e. processes and data of the protocol, i.e. User name, password, data, message, sender ID, receiver ID.

States:

A – The initial state of the sensor network, packets have to be transmitted.

B – Source ID of the node that sends the data or information.

C – HELLO/ Sybil attack so compromised nodes are present to be available on the neighbor list.

D – Neighbor node list is available for choosing the receiver, which may have compromised nodes from the Sybil/HELLO attacks. The random function chooses the compromised node.

E – The packet is transmitted to the receiver, a compromised node has been chosen as the receiver.

F – Sinkholes thus created affect the normal functioning of the network.

G – The packet is transmitted to an uncompromised node that has been chosen as the receiver.

H – The packet reaches the destination (base station).

Actions:

T_1– The base station has requested information from the sensor nodes and data packet an Source ID available.

T_2 – The Source ID and Data packet is made available at the next state.

T_3– The neighbor list is created. The compromised receiver is chosen from the neighbor list using the random function available at the node.

T_4– Receiver ID is attached to the Source ID and Data packet, so that only the receiver can transmit the packets to its neighbors. (Source ID + Receiver ID + Packet).

T_5– There are intermediate transmissions between the nodes, before the packet reaches the base station.

T_6– Compromised nodes relay HELLO packets to indicate their false presence as neighbors to the source node. There are intermediate transmissions between the nodes, before the packet reaches the base station. The final transmission of the packet to the base station takes place.

T_7 – After transmission of the packet the sensor network enters the initial stage A.

4.1.3 Determine the reachability graph of attack Petri net

The reachability graph of a net helps identify the path from one state to the final state. Determining the reachability graph of protocol petri net ensures that when matching the two nets (namely protocol and attack net) we obtain the best possible match that will be of maximum benefit to the attacker and cause maximum damage to the user.

4.1.4 Categorization of resources in protocol model

Each resource in a protocol and attack net is associated with a category. The categorization of resources helps us identify interesting properties of the resources to characterize their attackability.

Following are the four categories we define for resources:

- Targets

These are the processes or data resources that are used by the adversary to gain control modify and compromise in a system. A target is a distinguished process or data resource in the network that plays a critical role in the attacker achieving his goal. Example of data targets are sensitive data, password, data carried by messages.

- Enablers

These are the other processes or data resources that are used by the adversary to gain control over the targets. We use the term enablers for any accessed process or data resource that is used as a means of the attack but is not signed out to be a target. Message sender, message receiver can be termed as enablers.

- Channels

An adversary gains control over the resources through communication channels. A channel is a means of communicating information from a sender to a receiver. Communication Messages are the means by which the attacker gains access to the targets on system. Carriers deliver attack from one node to another in the network, for example, message containing unencrypted password.

- Access rights

Control over processes or data resources is subject to constraints imposed by the system's set of access rights. These rights are associated with each user, process and data resource. The more generous the access rights, the larger the probability of attack. We also represent conditional access rights based on number of conditions and combinations of data and processes.

Access rights are associated with all resources. These access rights determine the type of access available on a particular resource. Conceptually we model these rights as a relation, suggestive of Lampson's original access control matrix [6]:

$$Access \subset Principals \times Res \times Rights$$

Where $Principals = Users \cup Processes$, $Res = Processes \cup Data$.

When required, to represent conditional access rights, we can extend the above relation with a fourth dimension *Conditions* for conditional access rights,

$$Access \subset Principals \times Res \times Rights \times Conditions$$

Following table shows the common resources that can be identified as candidates for attack. The table also shows the access rights and categories these resources fall into. Access rights are represented by single letters as follows:

Read is represented by a 'R', Write is represented by a 'W' and Read/Write is represented by 'B'. Send is represented by 'S' and Receive is represented by a 'R'.

Resources	Representation	Targets	Enablers	Carriers	Access Rights
Message contains password	M			✓	Read (R)
Message contains data	D			✓	Read /Write (B)
Unencrypted User Password	P	✓			Write (W)
Username	U		✓		Read (R)
Authenticated Sender	A		✓		Send (S)
Unauthenticated Sender	S			✓	Send (S)
Unauthenticated Receiver	T		✓		Receive (R)
Authenticated Receiver	R		✓		Receive (R)

Table 1. Resource categorization and representation

Resource identification for wanderer protocol and Sinkhole attack:

All the resources that are potential targets of attack are identified in the wanderer protocol. All these resources are then categorized based on the attacker's goal. The access rights for these resources are expressed as a Read, Write, Send, Receive or a Read/Write operation. Similarly, we identify the resources that attacker is looking for to execute its attack. Resource Categorization of Wanderer protocol and Sinkhole attack has been given in the following tables:

Resources	Representation	Targets	Enablers	Carriers	Access Rights
Message contains sensitive data	D	✓			Read /Write (B)
Encrypted password	P	✓			Read (R)
Node List	N			✓	Write (R)
Unauthenticated Sender ID	S		✓		Send (S)
Unauthenticated Receiver ID	T		✓		Receive (R)

Table 2. Resource categorization and representation for Wanderer Protocol

Resources	Representation	Targets	Enablers	Carriers	Access Rights
Message contains sensitive data	D	✓			Read /Write (B)
Node List	N			✓	Write (R)
Unauthenticated Sender ID	S		✓		Send (S)
Unauthenticated Receiver ID	T		✓		Receive (R)

Table 3. Resource categorization and representation for Sinkhole Attack

4.1.5 Pay-off function assigns costs to the categories

A payoff function F: Category → [0, 1] assigns payoffs to each resource. Payoffs represent the likelihoods of damage. A category with a high payoff indicates that resources of that class are more likely to be attacked and can damage the system than resources of an attack class with a lower payoff. Our approach for defining a payoff function is to quantify the "damage" the adversary can effect if resources in a given attack class are compromised, e.g., in terms of cost to repair the system. We assign a payoff of '1' to the resource that has the highest risk of causing damage and '0' to the resources that have lowest or no risk of a causing damage. The payoff is also defined for access rights and sequences the resources follow in the attack Petri net. For example a combination of unencrypted username and unencrypted password input at same time is more unsafe than a username and encrypted password or just a username input at different stages in protocol.

We define payoff to the three main factors that influence the vulnerability of a protocol. These factors are Resources, Access rights and Sequence.

Resources:

If a resource exists in both the attack and protocol net then we assign a cost of 1 to this match. This would guarantee that the resources needed to launch the attack are available in the protocol net, further increasing the chances of an attack. Resource types could be an unencrypted message, data or a password or it could be a username.

Access Rights:

Once we have matched the resources in both the attack and protocol net we would like to look for their access rights. If these also match, again a payoff of 1 is

assigned to the matching access rights, further guarantying a stronger possibility of launching an attack. Access rights are as follows: send (S), receive (C), write (W), read (R), read-write(B)

4.2 Damage Estimation

4.2.1 Matching

A protocol 'P' is vulnerable to an attack 'A' if there exists a set of common resources and conditions in the protocol and attack petri net such that the attack can be launched on the protocol.

Starting with the initial transition, all incoming states are explored for this transition, then the states incoming to the next transition, and so on. To find all possible states we use breadth-first search technique. If the graph is inherently bounded, its reachability graph will always have a finite amount of states.

To explain this in detail let us consider a protocol petri net and an attack petri net. In order to launch an attack the attack petri net requires specific transition to be fired with some specific conditions and resources. The access rights of these resources are also taken into consideration. If the protocol petri net satisfies all the necessary conditions like resources and access rights that are necessary to launch an attack, then we can say that protocol is vulnerable to the attack.

As shown in figure 8 the attack can be launched by firing transition T_1. The firing of transition T_1 requires all the resources and conditions to be satisfied for state 'A_1'. The petri net model of protocol here has place S_1 that satisfies all the conditions and resources

requirements for launching attack. Thus, we aim at finding the best possible matches in both the protocol and attack petri net.

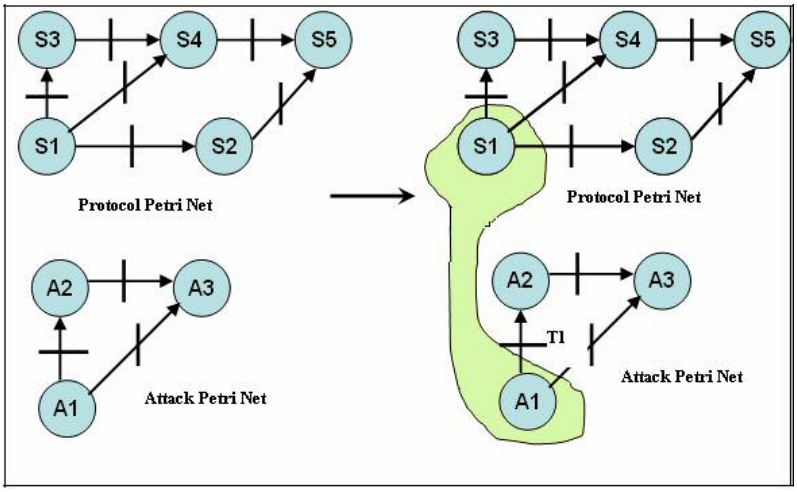

Figure 8. Finding a matching state in protocol to launch attack.

4.2.2 Matching of Two Petri Nets

We assume that the two petri nets are state minimized. This assumption also helps to find the markings that will be of maximum benefit to the attacker and cause maximum damage to the user.

4.2.2.1 Algorithm description

We propose an algorithm to find matching resources, sequence and access rights between the attack petri net and protocol petri net. The algorithm works as follows:

In the attack net, the transition that initiates the attack is identified. All the states for the given transition and transitions prior to this transition are identified. Identify all the resource types required by each state and their respective access rights for each transition.

For each resource identified in attack net, find similar matching resource type in protocol net. Using the payoff function, assign the cost of damage to each of this resource matching found in the protocol net. If there is a match found for a resource, find if the resource's access rights in the protocol net also match with that of the attack petri net. For each state in protocol net, we look for a state with maximum damage cost for resources, and access rights. Repeatedly search for all the states in protocol unless all identified resources in attack net are searched for in all states of protocol petri net.

The solution to the matching would result into either of the following:

- None of the two: resource type and access rights match: There is then an almost negligible possibility of an attack.
- If the protocol have same the type of resources available as that of an attack and if the protocol states (representing resource types) have the same access rights, then there is a higher chance of an attack .

4.2.2.2 Algorithm

Algorithm Protocol Damage Estimator

Input: Protocol and attack petri net namely P and A labeled with resource types and access rights.

Output: States with maximum vulnerability identified along with their estimated damage cost for resource type and access rights.

1. Initialize the protocol matrix $P1_{ij}$ and $P2_{ij}$ with resources and access rights for resources of the protocol net respectively.

set $P1_{ij} = \ell_{\mathbf{1}}(P)$ and set $P2_{ij} = \ell_{\mathbf{2}}(P)$;

where ℓ_1 and ℓ_2 are labeling functions for resources and access rights respectively.

1. Initialize the attack matrix $A1_{ij}$ and $A2_{ij}$ with resources and access rights for resources of the protocol net respectively.

   set $A1_{ij} = \ell_1(A)$ and set $A2_{ij} = \ell_2(A)$;

2. While Current_Transition < = Attack Transition;

   Attack Transition is the transition that initiates the attack.

   Initialize ja = 0; where ja is the total number of attack states.

   for s=1…m where m is the total number of protocol states.

   for jp=1…n where n is the total number of protocol resources.

   If A1[current_transition][ja] $\neq$ 0;

   If A1[current_transition][ja]=P1[s][jp];

   Then increment cost_resource(s) by 1

   Check if access rights also match,

   If A2[current_transition][ja]=P2[s][jp];

   Then increment cost_accessrights(s) by 1

   Increment ja by 1;

   Calculate the total cost in percentage using the formula:

   cost_resource(s)′ = (cost_resource(s) /total_resources)* 100;

   cost_accessrights(s)′ = (cost_accessrights(s) /total_resources)* 100;

   Subtract 1 from current_transition.

   End.

Let $f_1,.....f_n$ be all the matches which are possible based on resources between the two petri nets p_1 and p_2. The cost of damage f^* is defined by:

$$\text{Cost}(f^*) = \max_{f_1,...f_n} \{\text{cost}(f)\}$$

Similarly, the cost of damage based access rights will be a $\max\{\text{cost}(f)\}$ for $f = 1$ to n. This cost is termed as the "Cost of damage to the user" or "Benefit to the attacker".

4.2.3 Yields common states and damage cost for protocol

The tool for matching petri nets will yield a set of states that are most vulnerable to an attack. Our tool for matching the two petri nets gives the total payoff value for protocol petri net which is termed as the "Cost of damage to the user" or "Benefit to the attacker". This payoff value is the percentage of matching found for resources and access rights. The higher the percentage value of resource or access rights for a state, the higher is the probability of an attack being successful. If for any state, there is a 100% resource match and 100% access rights match, then the attack can surely be launched on this protocol through this state and the damage depends upon the category the resources belong to. Analysis of this cost is done in the following section and some protective measures are suggested to reduce the number of resources exposed to the attack.

The Wanderer protocol and Sinkhole attack petri nets were input to the damage estimation tool and the following results were obtained from the matching of protocol and attack petri nets. 100% resources matched and 80% access rights matched at transition T3 for states E and D in the wanderer protocol net. This states that the sinkhole attack can be easily launched on the wanderer protocol since all the resources that are

potential targets of an attack are completely exposed to the sinkhole attack. In the following section, we analyze the wanderer protocol to prevent it from being attacked.

4.3 Protection

If an intruder state is no longer reachable from any state of the protocol petri net, we can state that the protocol is safe with respect to that particular attack.

The Network is unsafe if we obtain states in protocol net that are similar to the attack net, and hold resources that are required to launch an attack. Using our model we can obtain the states in protocol net and resources that match with the resources in the attack net and thus obtain the damage cost. This will enable us to find out the minimal set of measures (state, resources and access rights and sequence of availability) that must be prevented to guarantee that the adversary cannot achieve his goal. The aim here is to avoid all those measures that can lead to attacker's goal.

To define the protective measures we will first define the attack surface. The categories (Targets, enablers, carriers and access rights) all together comprises of an attack surface of a protocol. This means that the more the resources from all the categories, the larger is the attack surface and higher the chances of an attack on the protocol. The protective measure is defined as an action that reduces the attack surface and thus reduces the chances of protocol being attacked by the intruder.

Intuitively, if there are more resources in category Targets, the higher the chance of an attack being successful. The more the enabler, the larger is the chance of an attack or chance of an attack being successful, the more the carriers, and the larger is the chance

of an attack being successful. But above all, more generous the access rights, higher are the chances of an attack being successful.

4.3.1 Protection against the intrusion

There can be two different approaches to protect the protocol from an attack.

- We can control the constraints imposed by access rights on all resources.
- Eliminate the identified vulnerability
  - Controlling constraints imposed by access rights on all resources

One approach to reduce the attack surface is to change the limit on accessing the resources. Each resource has an access right associated with it. If the constraint on accessing the resources is restricted to minimal access, the attacker will not get exactly the same conditions as required to launch the attack and would fail to do so. For example, if we have a resource as "unencrypted data" with access rights as "Read" in a state. Assuming that for a specific attack, our tool finds that this state is the weakest state and has high vulnerability to an attack, then changing the access rights on unencrypted data from "Read" to "protected" can thus reduce the probability of attack through this state and further fail the attacker's goal to damage the protocol.

  - Eliminate the identified vulnerability

Changing the amount of resources available at a particular transition will reduce the availability of the identified vulnerability. The protocol petri net can be modified to allow fewer resources to be available to the attack net which will reduce the attack surface and thus reduce the probability of attack on the protocol. In order to achieve this some extra states, transitions and inhibitor arcs are introduced into the petri net of the protocol such

that the resources are never directly accessible to the attacker. The inhibitor arc is represented as an arc with a ball head. The arc is always from a place p to a transition t. The meaning is that if p has a token it will hinder t from firing, i.e., opposite to that of a normal arc. This will also change the order in which the resources can be available to the attacker. The attacker will thus fail to get all the resources at the same time.

Algorithm Protection Model

Input: Protocol Petri net P labeled with resource types and access rights for all states and transitions. Assume V_t, V_s and V_{t-1} as set consisting of transitions and states.

Output: Eliminate the identified vulnerability in protocol by reducing the order of availability of resources in the protocol.

1. Identify the transitions that are vulnerable to an attack. Let these transitions be represented as elements of set V_t. Determine the states that are input to these identified vulnerable transitions and represent them as elements of set V_s. Assume all transitions incoming to states in V_s as elements of set V_{t-1}.
2. V_t' is set of transitions added later on to the protocol with some encryption operation. A decryption operation is also required at transition to retrieve resource values and further perform the protocol operation normally and securely.
3. If $\sum V_s = 1$ i.e. the total input states to the transition in V_t is 1,

   Then change the access rights of resource types in that state.

   Else If the number of states entering a transition is more that one i.e. $\sum V_s > 1$ then introduce inhibitor arc from every state in V_s to the previous transition of another state i.e. the transitions in set Vt_{-1}. If there are n states entering into the transition then total of n-1 inhibitor arcs are inserted.

4. Once all inhibitor arcs are inserted, another transition is added to the state above the identified vulnerable transition. V_t' transition ensures that if two resources are available at this stage they are in encrypted form. When all the resources are received at the vulnerable transition in encrypted form, the original protocol operation on that transition can be executed along with a decryption operation to retrieve the resources in their original state.
5. Once all the inhibitor arcs are added, we have avoided the chance of availability of resources at the same time. But there needs to be an encryption for each resource.
6. Add a transition for each state with any encryption operation that restricts the access to resource in this state. Assume these transitions as member of set Vt´.
7. Once all the resources from all incoming states are received in encrypted form they can be decrypted together and the protocol operations can be performed securely.

End.

Definition: **Protection Algorithm:** *This algorithm ensures that the resources needed by an attacker are never available at the same time***.**

The Following example illustrates the above algorithm and resource protection. In figure 9 consider a petri net such that resources *A* and *B* are available in state S_1 and S_2 and are input to transition T_3 at same time. Both *A* and *B* must be available at the same time to enable the transition T_3. Assume that transition T_3 is some operation '*x*'. An intruder also needs the two resources A and B at the same time to launch the attack. A protective

measure to avoid the attack is to change the order of availability of resources *A* and *B* in protocol net.

State S_1 and S_2 has resource *A* and *B* respectively. An inhibitor arc is added from both states S_1 and S_2 to transition T_1 and T_2 respectively.

Adding this arc ensures that transition T_1 will take place only when state S_2 is empty(i.e. no resources in S_2) and T_2 is enabled only when state S_1 is empty(i.e. no resources in S_1).

An operation- *hide(R),* where *R* is the resource, encrypts the resources such that the intruder cannot access it.

The final transition T_3 of protocol is modified to include an extra step of unencrypting the resources from hide operation, such that: *unhide(hide(R)) = R,* where R is a resource. Transition T_3 is defined as *unhide()* + operation 'x'.

Definition: *The original protocol functioning remains the same even after adding extra states and transitions and inhibitor arcs.*

Assume that *hide()* and *unhide()* does nothing. Then in figure 9b at transition T_3 we have *hide(A) = A and hide(B)=B* (since we assumed hide does nothing). This condition is similar to the condition in figure 7a at transition T_3, both *A* and *B* are available at same time. Hence, the original property and working of protocol is retained irrespective of adding inhibitor arc.

Assume that states S_1 and S_2 are empty at the initial stage. Since S_1 is empty and resources for enabling T_2 are available, transition T_2 is fired and now state S_2 has resource *B*. Transition T_1 is not fired unless S_2 is empty. So, T_5 is fired and resource *B* goes to state S_4. Now, when resources for enabling transition T_1 are available it can be fired, as S_2 is also

empty, once T_1 is fired, resource A is available in S_1 and further after transition T_4 it is available in S_4. Even though both resources are available in states S_4 and S_5, they are hidden or encrypted. In transition T_3, these resources are unencrypted or operation unhide is performed and normal operation of T_3 is executed.

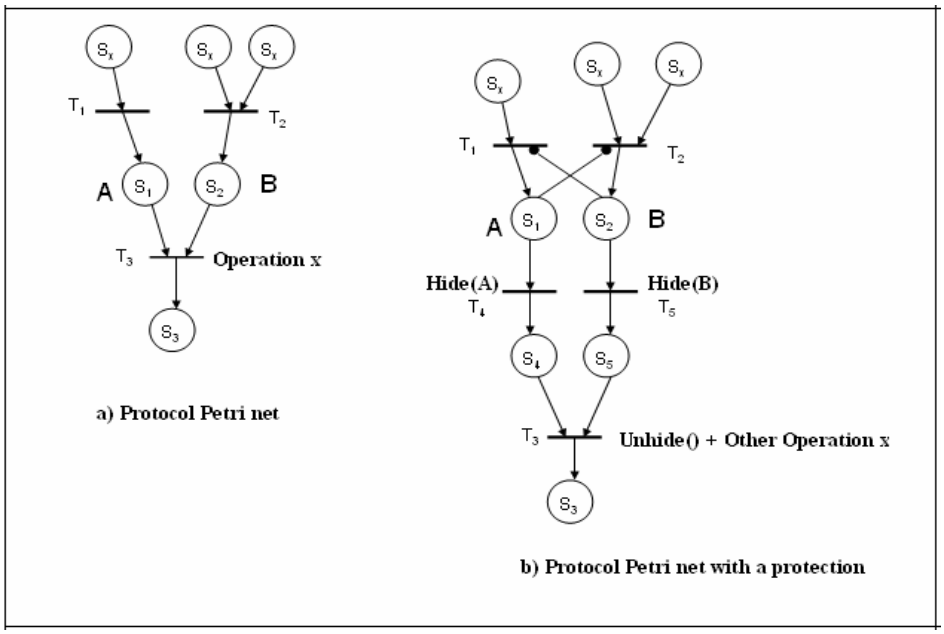

Figure 9. Protective Measures

Eliminate the identified vulnerability in Wanderer Protocol

We consider reducing the order of availability of resources in the wanderer protocol to illustrate our protective measure. The following figure shows the wanderer protocol with inhibitor arcs. Inserting the inhibitor arc guarantees that the resources are not available at same time. Following is a detailed description of the working of protocol. We use the second approach of changing the order of resource availability by adding inhibitor arcs. The following figure shows how the protective measure can reduce the wanderer protocol vulnerability towards a sinkhole attack. The hide operation at

transition T1′ is the encryption operation that encrypts the data packet, unhide operation at transition T2′ can be a decryption method to retrieve the data packet.

Hide: Upon receiving a message from the base station the leaf node enters the next stage to transmit the requested information. The message obtained from the base station is decrypted with the individual key already assigned in stage A.

Unhide: The packet is transmitted to all the neighbor nodes present in the neighbor list. Each neighbor node receives the transmitted packet but confirms its identity as the receiver before transmitting the packet again. The transmitted packet is decrypted using some decryption mechanism.

When the protected protocol and attack was again input to the tool, the resources found were 50% at one state and 50% at another state and this proves that the protective measure reduced the attack surface to a greater extent. This shows that the protocol is now more secure(compared to 100% and 80% respectively previously).

T1 is fired only when there is a resource in state A but not in sate D′. Similarly, T2 is fired only when there are resources in state B and C but not in state E′. Thus, at any given point both state E′ and D′ cannot have resources i.e. either of the two states is empty. Assume that initially only state A has a resource available. Then transition T1 is fired since state D′ is empty. Firing changes the state of resources from state A to state E′. Now, let's say that B and C each have a resource available. But in this marking transition T2 cannot be fired because there is a resource in state E' and the inhibitor arc will allow firing of transition T_2 only when state E' is empty. Transition T_2' is enabled and the hide operation is executed i.e. the encryption of resource is done and the resource changes its state to E''. Now, since state E' is empty, T_2 can be fired and the resource changes its state

to D' and further firing of transition T_2' changes its state to D''. Thus, the resources (data packet and Source Id) are never available at same time. They are available at same time just before transition T_3 but since they are encrypted, the intruder will fail to launch the sinkhole attack.

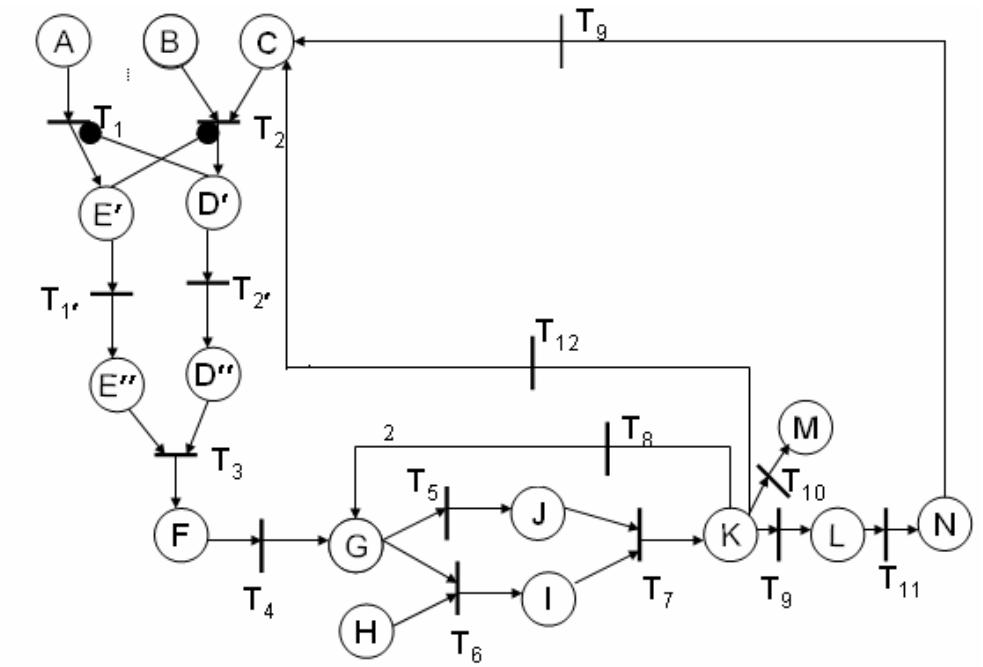

Figure 10. Modified Wanderer Protocol with protection.

CHAPTER IV

DAMAGE ESTIMATION TOOL

A tool has been developed to determine the damage cost to the protocol with respect to an attack. We evaluate our system by implementing test data which contains both normal and malicious data. Normal data refers to the protocol petri net and malicious data is the attack petri net. The data was run through our tool and the damage cost estimates were found for the wanderer protocol.

The tool uses the above mentioned algorithm to find the matching states and give damage cost estimation for resources and access rights that are probable candidates for launching an attack. This tool has been implemented using object oriented programming, C++. Following figure shows the System diagram for the tool.

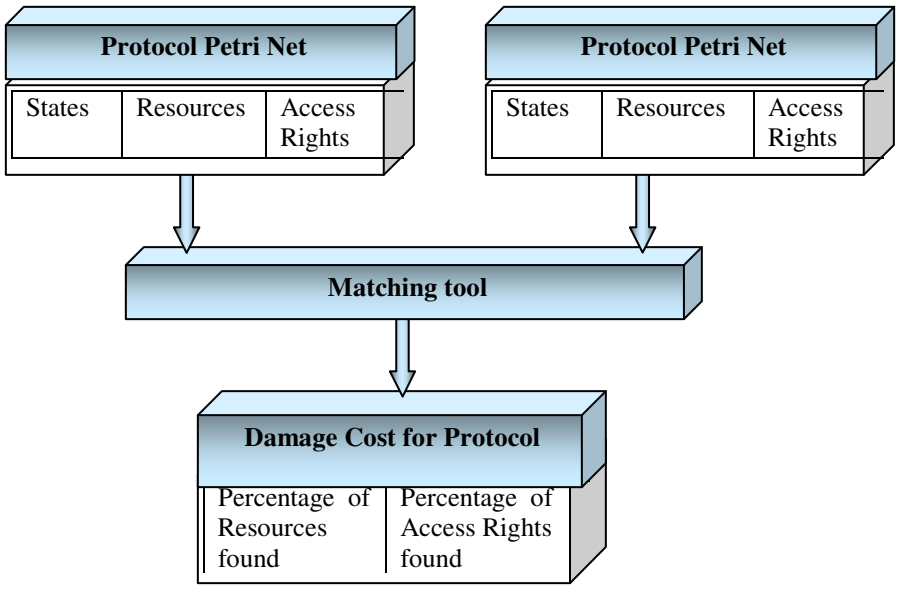

Figure 11. System Diagram for Tool.

Input: Input to the tool is the protocol petri net and the attack petri net i.e. the total number of states and resources these nets consist of and markings for each state that represent the total number of resources required for each state. Each entry also represents the access rights for that resource in that particular state. An example of our input description is given by figure 12. Each row represents a state in the protocol net and each column represents a resource. Hence, $(R_1, R_2 \ldots R_m)$ is m number of resources in a protocol petri net and $(S_1, S_2 \ldots S_n)$ are n number of states in the protocol petri net, where m and n are the total number of resources and states in a petri net. Access rights are represented by their symbolic representation (see table 2). Since, an access right is linked to each resource; an entry *'x'* in S_iR_j denotes that there exists a j^{th} resource in i^{th} state with access rights *'x'*.

A similar representation is given for the attack net. The symbolic representation of all resources and their access rights can be derived from the table as shown below. A table can be defined for each resource and access rights that provides a detailed description of all available resources in a protocol and attack petri net.

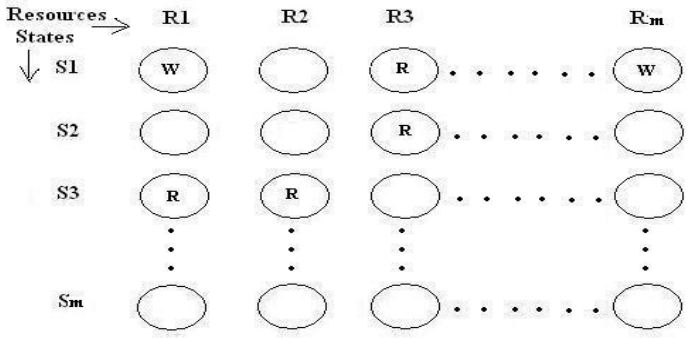

Figure 12. Example input file: resource allocation for different states.

Resources	Representation	Targets	Enablers	Carriers	Access Rights
Message contains password	M			✓	Read (R)
Message contains data	D			✓	Read /Write (B)
Unencrypted User Password	P	✓			Write (W)
Username	U		✓		Read (R)

Table 4. Example table for resources and their Categorization.

Let us assume that transition *T1* requires all resources incoming via state *S1*. From figure 12, we see that attack requires resource *R1* and *R3* with access rights as Write (W) for resource R1 and *read* (R) for resource *R3*. The algorithm would look for resources *R1* and *R3* in protocol net. Once it finds these two resources it also checks for the access rights. From the protocol and attack figure, we find that all resources and access rights of attack net for state 1 matched with the protocol net; thus giving damage cost of 100% for resources and 100% for access rights and 100% for sequence.

CHAPTER VI

CONCLUSION

In this thesis we investigate the vulnerabilities present in protocols and calculate the amount of damage that can arise if these vulnerabilities are exploited by a malicious node. Our approach proves that protocol and attacks can be modeled as Petri net models and the matching can be applied to these nets to identify the damage cost and suggest protective measures. The resources that can be potential targets of an attack are identified as; categories.

For the experimental validation of our proposed model, we have developed a tool to correlate the protocol Petri net and the attack Petri net to identify the worst vulnerability in the protocol and a payoff function was applied to matches found for attack and protocol resources. This payoff indicated that closer the matching of the resources and access rights between protocol and attack, the higher the chances of successfully launching an attack.

Once the weak link in the protocol is identified, we propose approaches to reduce or eliminate the identified vulnerability. Two protective measures are suggested to protect the protocol and reduce the vulnerable states. The measures suggested are verified by implementing an example protocol. The modified protocol is also modeled as a Petri net and again run through our damage estimation tool.

Future work related to our approach will involve identifying more categories for damage estimation and matching the structures of two petri nets. This will include identifying the vulnerabilities and classifying them efficiently across the different dimensions. Further work on combining and synchronizing the best features of multiple protocols to create an improved protocol would be another avenue for future work.

REFERENCES

[1.] Adel Hlaoui, Shengrui Wang, "A New Algorithm for Inexact Graph Matching" 16th International Conference on Pattern Recognition (ICPR'02), Vol. 4, 2003.

[2.] Josh Broch , David A. Maltz , David B. Johnson , Yih-Chun Hu , Jorjeta Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp.85-97, 1998.

[3.] Zhang Congzhe, Zhou Mengchu, "A Stochastic Petri net Approach to Modeling and Analysis of Ad Hoc Network", In Proceedings of ITRE 2003 International Conference on Information Technology: Research and Education, pp.152-156, August 2003.

[4.] C. Xiong, T. Murata, and J. Tsai, "Modeling and Simulation of Routing Protocol for Mobile Ad Hoc networks Using Colored Petri Nets," Research and Practice in Information Technology, Australian Computer Society vol. 12, pp.145-153, 2002.

[5.] Michael Howard, Jon Pincus, and Jeannette M. Wing, "Measuring Relative Attack Surfaces", Workshop on Advanced Developments in Software and Systems, 2003.

[6.] Zhengming Shen, Johnson P. Thomas, "Measurement of Security of Ad Hoc Network Protocols". Private Communication.

[7.] Arasavel Ramaraj, "Modeling of sensor networks-Attack Surface, QoS Surface", MS Thesis, Department of Computer Science, Oklahoma State University, 2006.

[8.] V. Ramamurthy, “A Petri net reduction algorithm for protocol analysis”, http://delivery.acm.org/10.1145/20000/18191/p157-ramamoorthy.pdf, [last accessed September 18, 2006].

[9.] Fei- Yue Wang, Kevin Glidea and Alan Rubenstein “A colored Petri net model for Connection Management Services in MMS”, Center for Manufacturing Productivity and Technology Transfer.

[10.] Khaled Mustafa, Salah Aly, “Protocol verification and Analysis using colored Petri nets”, Technical Report, Cairo University, July 2003.

[11.] Madhav Neel, “Correctness and error detection in a partial order model of distribution program executions”, PhD Dissertation, Department of Computer Science, Stanford University, 1993.

[12.] Congzhe Zhang, Mengchu Zhou, “A Stochastic Petri Net Approach to Modeling and Analysis of Ad Hoc Network”, http://www.stevens.edu/wireless/research/ petrinet/zhang-zhou-petrinet-adhoc.pdf, [last accessed – April 28, 2006].

[13.] Lars Michael Kristensen, “Using colored Petri nets in the development of protocols for ad-hoc networking”, http://tfs.cs.tu-berlin.de/~gschroet/int04/krist.ps [Last accessed – May 30, 2006].

[14.] C. E. Perkins, E.M Royer, “Ad Hoc on- Demand Distance Vector Routing”, http://www.cs.ucsb.edu/~ebelding/txt/aodv.ps [last accessed - April 20, 2006].

[15.] Pratyusha Manandhata, Jeannette M. Wing, “Measuring a System’s Attack Surface”, http://www.cs.cmu.edu/afs/cs/project/ calder/www/tr04-102.pdf [last accessed – April 10, 2006]

VITA

ADITI GODSE

Candidate for the Degree of

Master of Science

Thesis: PETRI NET BASED MODEL FOR PROTOCOL DAMAGE ESTIMATION AND PROTECTION

Major Field: Computer Science

Biographical:

Personal Data: Born in Vidisha, Madhya Pradesh, India, November 3, 1982, the daughter of Mr. N.R. Godse and Mrs. Lata Godse.

Education: Obtained Senior High School Diploma from All Saints' School, Bhopal, India, in May 2000; completed Bachelor of Engineering in Computer Science from Rajiv Gandhi University, Bhopal, India, May 2004; fulfilled requirements for the Master of Science Degree at Oklahoma State University in May, 2007.

Experience: Graduate Assistant in Department of Distance Learning (CEPD), Oklahoma State University, from January 2005 to December 2006.

Technical Support Engineer, eClinicalWorks, MA, from May 2006 to August 2006.

Name: Aditi Godse Date of Degree: May, 2007

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: Petri Net based Model for Protocol Damage Detection and Protection.

Pages in Study: 49 Candidate for the Degree of Master of Science

Major Field: Computer Science

Abstract:

In this thesis we investigate the vulnerabilities present in protocols and the damage that can arise if these vulnerabilities are exploited by a malicious node. In particular, we model protocols using Petri nets. Petri nets allow us to simulate the protocols and reason about them. Attacks are also modeled using Petri nets. We develop a tool to correlate the protocol Petri net and the attack Petri net to identify the worst vulnerability in the protocol and a payoff function is applied to measure the potential damage. Once the weak link in the protocol is identified, we propose approaches to reduce or eliminate the identified vulnerability. The modified protocol is also modeled as a Petri net.

ADVISOR'S APPROVAL: Dr. Johnson P. Thomas