IJICTR International Journal of Information & Communication Technology Research

Volume 4- Number 3- June 2012 (45- 58)

Cyber Attack Simulation for Operational Security Evaluation Using Coloured Petri Nets

Mehrdad Ashtiani
School of computer Engineering
Iran University of Science and Technology
Tehran, Iran
m_ashtiani@comp.iust.ac.ir

Mohammad Abdollahi Azgomi
School of computer Engineering
Iran University of Science and Technology
Tehran, Iran
azgomi@iust.ac.ir

Received: February 15, 2012- Accepted: May 17, 2012

Abstract— **Today, cyber attacks to computer networks have turned into a real challenge for network administrators. A wide range of methods have been used for attack modeling and security quantification. The most important drawback of the existing methods is that they are not based on real security-related information of networks. Our aim has been to overcome this drawback by using high-level modeling techniques and real security relevant information of systems. In this paper, we use coloured Petri nets (CPNs) for attack modeling. One of the objectives of this paper is to show the power and flexibility of CPNs for high-level attack modeling. In our work, the important elements of networks involved in cyber attacks, such as hosts, attackers, intrusion detection and prevention systems, servers and firewalls are modeled as reusable CPN sub-models. In other words, with the help of hierarchy and the abstraction provided by CPNs, we have proposed a framework for modeling and evaluation of the impacts of cyber attacks on networks. Through an illustrative example, we have modeled a sample network and some attack scenarios by using the security-relevant information extracted from open source vulnerability database (OSVDB). Finally, we have evaluated some security measures of a sample network.**

Keywords: Cyber attacks, attack modeling, coloured Petri nets (CPNs), availability evaluation, simulation, CPN Tools.

I. INTRODUCTION

One of the everyday challenges of network administrators is to determine the impacts of different attacks on their networks. They want to know how much their systems are vulnerable to different attacks and evaluate the status of the security of their systems. Constructing models of networks using real security-related information and simulating them can help network administrators to answer these "*what if*?" questions.

In this paper, coloured Petri nets (CPNs or CP-nets) [1, 2, 3] are used for modeling network elements and simulating the cyber attack processes using CPN Tools [4]. One of the objectives of this work is to present a high-level modeling framework through which various networks with different equipment and configurations can be modeled by connecting ready-to-use modeling elements. In this approach, we have tried to consider the parameters affecting the time and success of different attacks. The following real security-related information is used as parameters of the models:

1. The exploit information is extracted from source vulnerability database (OSVDB). Parameters, such as, access vector, access complexity, required operating system, the required open ports and related services and different exploit impacts on confidentiality, availability and integrity (CIA) are considered.

2. The patching level of the network clients and host devices.
3. The defensive elements, such as firewalls and intrusion detection/prevention systems (IDSs/IPSs) are also considered.

In the proposed framework, three levels of skills are considered for attackers and network administrators. Based on the selected skill of attacker and administrator, the attack and defense patterns will be different. In modeling, considering the defensive element's behavior is very crucial to have a more valid model regarding the modeling of cyber attacks. Real networks consist of many defensive elements, such as firewalls and IDSs in different layers. This work tries to include the role of these elements in the proposed framework.

On the other hand, the impacts of attacks are categorized in three main aspects: confidentiality, integrity and availability (CIA). One of the advantages of using CPNs is the ability to distinguish between these aspects. Any of these aspects can be represented by a different colour set. Each attack can have a complete or partial impact on one or more aspects. These aspects can also have impacts on each other. For example, the impact on integrity can lead to the impact on availability. This is shown in Figure 1.

A wide range of analytic, simulative and game theoretic methods have been used for attack modeling and security quantification. The most important drawback of the existing methods is that they are not based on real security-related information of computer systems and networks. Our aim has been to overcome this drawback by using a high-level modeling formalism ad real exploit information and security patching level of systems in a unified framework.

The rest of this paper is organized as follows. In Section 2, we will provide a brief background for the cyber attack modeling process. In Section 3, related works are explained. In Section 4, the proposed framework for modeling cyber attacks, using CPNs is introduced. In Section 5, the evaluation results of the framework for different scenarios are presented. Finally, some concluding remakes are mentioned in Section 6.

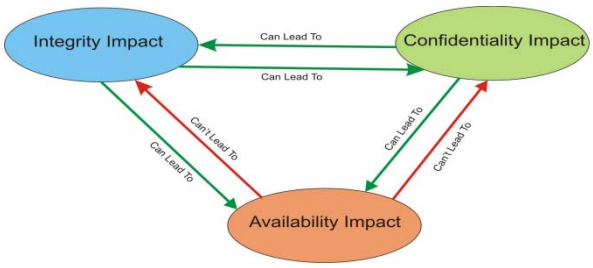

Figure 1. The impacts of security aspects on each other

II. BACKGROUND

For determining the details of attack process modeling we have to talk about the details and the important parameters of this process. Attackers look for the security vulnerabilities in their target systems. After gathering the information about vulnerabilities, they use appropriate exploits in order to gain access. As stated in [5], the steps involved in an intrusion process are as follows:

1. Information gathering: in this step attackers try to get as much information as possible about the target systems. Activities involved in this step are as follows:
   - Operating system fingerprinting, and
   - Port and service enumeration.
2. Determining vulnerabilities of services running on the target machine.
3. Selecting and sending an exploit in order to gain access to the target system.
4. After the intrusion, the attacker can take the advantage of the system based on his goal and desire.

It is worth mentioning that based on the chosen vulnerabilities and the selected exploits, the impact of intrusion will be different. Exploits can have different effects on one or more of the three aspects we mentioned earlier (*i.e.*, CIA). These effects can be partial or complete. For example, an exploit can have partial impact on confidentiality, integrity or availability.

In order to model the attack process correctly, we need to employ defensive elements in the network. Most of today's networks use firewalls and intrusion detection/prevention systems. Firewalls are usually the first layer of defense against malicious traffics. These devices are placed in the edge of networks. Firewalls may implement one of the two access policies: *open access policy* (in which all ports are open except the black list) and *closed access policy* (in which all ports are closed except the white list) [5]. In the proposed framework, we model the closed access policy which is more common.

IDS/IPSs are the next layer of defense. These systems are generally divided into two categories:

1. *Signature-based*: these systems keep a signature list of all known public exploits (*a.k.a*, attack patterns). After receiving any packet, they try to match the payload of the packet to a signature in their database. If a match is found, the packet is considered to be malicious and appropriate alerts will be produced. Snort [6] is one of the most famous IDSs in this category.
2. *Anomaly-based*: these systems try to profile the behavioral models of "normal traffics" over time. Then, the incoming traffics will be checked against this profile. If the traffic's behavior is different from the profile, then the traffic is suspicious to be malicious and appropriate alerts will be generated.

For modeling IDSs and IPSs in our framework, we have used the first category. This is because signature-based IDS/IPSs are more common in the industry and

tend to have far less false positives and false negatives compared to anomaly-based IDS/IPSs.

The important thing here is that IDSs are passive elements, which means that they are not capable of changing or blocking the incoming traffic and thus they cannot prevent attacks. This is why IPSs were introduced. Some of the most important usages of IPSs are as follows:

1. Shutting down or restarting servers: this is very useful for denial of service (DoS) and distributed DoS (DDoS) attacks.
2. Dropping malicious packets.
3. Blocking senders of malicious packets by changing the rules of firewalls.
4. Providing solution if one exists for the targeted vulnerability.

III. RELATED WORK

The existing literature on modeling and simulation for security evaluation are briefly reviewed in this section.

The existing works on modeling attacks and intrusions may be classified into two main categories:

1. Analytical modeling techniques (Markov chains, stochastic processes, etc.).
2. Simulation models (using simulation languages, tools or frameworks).

Analytical modeling techniques in comparison with simulation techniques have serious challenges. For example, state-based models, such as Markov chains and Petri nets, have the big challenge of state space explosion and so, highly unrealistic assumptions and abstractions should be taken into account in order tackle this challenge. These models are very hard to scale. It is very hard to create analytical models and formulae for today's large-scale, multi-technology, multi-protocol networks. This will impose an upper bound to the size of the network and entities that can be modeled.

Many of the works are on the basis of creating the model and then employing model checking techniques [7, 8]. Although this approach produces very considerable results, because of the existing issues and drawbacks, such as state space explosion, its application is very limited due to the scale of the networks it can model. There are also different approaches for modeling the intrusion processes. Many of the works are based on attack trees and attack graphs [9, 10, 11]. Lots of works are also based on Markov chains [12, 13]. Even though these approaches have many advantages, they are very limited from the scalability and flexibility point of view.

Game theoretic approaches have also been used to model the attack and defense processes. These approaches are usually limited in terms of the scale and are basically more theoretical than practical for network designers and administrators [14, 15].

Availability evaluation is a result of integrating dependability and security evaluations. By applying the same techniques and methods of dependability evaluation in general (and especially, availability evaluation), we can evaluate a cyber attack based on the defined operational measures and metrics [16, 17, 18]. One of the aims of our work has been to show the result of a cyber attack on operational measures such as utilization, availability and responsibility.

Two main drawbacks of the existing methods mentioned above are as follows:

1. These modeling techniques are very low level for the network administrators to actually implement them for the security evaluation of their network. The best case for a network administrator is to have the common networking elements and the ability to connect them together as high-level entities to build the model of their own network.
2. The input data of most models are more hypothetical than actual. This will lead to unrealistic results. In the proposed framework, we use real-world exploit information to determine the output of launching an exploit. Relative information for exploits, such as access vector, access complexity, the required ports, service version, the required operating system and even patching level of systems are considered. For the IDS/IPS we use the rule signature database of Snort to produce the most valid behavior of these systems.

As the best of our knowledge, there are three main works done for attack modeling with coloured Petri nets (CPNs).

- In [19], a technique for converting attack trees into CPNs and vice versa is introduced. As reported in this paper, all the operations that can be specified with attack tress can also be modeled with CPNs.
- In [20], hierarchical coloured Petri nets (HCPNs) is used for modeling attacks in two layers. One of the layers is used for modeling more general attacks and the other layer is used for modeling specific attacks. The main focus of this work is on multi-stage attacks.
- In [21], CPNs is used for modeling the integrity of a system. In this paper, incidents are classified into intentional and accidental. These two categories and faults are modeled using different transitions with different firing probabilities assigned to them in the CPN model. All attacks in this paper are limited to the system integrity breaches and there is no distinction between attacks with different impacts.

On the other hand, simulation techniques are easier to use and can be scaled appropriately due to taking into account only the simulated behavior of network entities. There are a lot of papers on simulation of attack processes such as [22, 23, 24]. In the following, we briefly review some of the existing results in this area, which has inspired us for our CPN-based framework, which will be presented in the next section.

In [2], a framework for designing and simulation of a network using Arena software is introduced, which allows designing networks with different configurations and then running the simulation of an attack based on the exploits that are in hand. The simulation is based on the common steps introduced in the previous section. In [25], a distributed simulation framework for cyber attack simulation is introduced. In the simulation model, defensive elements (such as IPSs) are considered. Common tasks of IPSs is introduced and implemented in this simulation model.

In [21], the main focus is on the system integrity impact. The problem with this research is, the very high-level and unrealistic Petri net created for this purpose. The whole attack process is modeled by four states and does not take into account any real information about how attack works and how it can be successful. We are strongly against assigning such static probabilities for attack success without taking into account the process itself.

In the paper "The simulation of attack and defense in OPNET environment" [26], the same problem exists. The problem is in, not considering the actual attack process and real world information. The whole simulation is divided into five states. These states again are very high-level and do not have any meaning without the appropriate context. This will lead to very abstract representation of the network behavior.

The work which is closest to ours, in terms of using real world attack process information is [5]. The drawback of this work is that only the source IP and target IP of the exploit is considered. No defensive element is taken into account and there are no patching process and no usage of wide array of related exploitation parameters.

From the point of view of the modeling language that was chosen, our work is similar to the work reported in [20]. But, the differences are worth mentioning, which are discussed below.

Although this approach uses the HCPN model, there is no detailed process underlying this high-level model. The attacker is modeled with only one place without the details of how the attacker works. The same statement is true for the file server (and servers in general) in this work. It is not even clear how this model actually works; because it seems that there are errors in the model itself (as notified by the red highlighting of the simulation components).

With regarding the descriptions of the benefits of our model, it can be seen that we provide a much more realistic and detailed simulation for different network elements (including the defensive elements). The same reasoning is valid for the work reported in "Attack modeling with coloured Petri net" [19].

IV. ATTACK MODELING WITH COLOURED PETRI NETS

The main open problem that we try to address in this paper is "*modeling the actual attack process with real world parameters*" in order to create a high-level framework of ready and easy to use components for the network administrators to simulate their networks. The following is the summary of the contributions of this paper:

1. The scalability of the simulation framework for being used for modeling potentially large networks.
2. Using real world data for different entities (such considering many exploitation parameters, using real world exploit signatures for IDSs and trying to simulate the actual path that attackers take through the attack process).
3. Considering different skill levels for attackers and modeling their behavior in respect to the selected skill.
4. Providing modularization through the use of hierarchical models in HCPNs.
5. Considering defensive elements in the simulation along with attackers in order to increase the simulation fidelity.

At first, let us assume that we want to model the network shown in Figure 2. In order to model this network, we intend to model each element's behavior separately and then model the whole network by composing the modeled elements in the highest abstraction level as reusable components. This approach gives us the power to model various networks by using reusable sub-models.

The created model for the network is shown in Figure 3, which is constructed by CPN Tool .This is the highest abstraction level in the hierarchy. Each primitive is modeled as a transition. In the lower abstraction levels, each of these transitions consists of several other sub-transitions and sub-models. In this model, the host computer (and the attacker) sends its packets to the server based on a Poisson process. These packets go through the firewall. If the packet is sent to an open port on the server, the firewall will let the packet pass (otherwise, the packet will be dropped).

Currently, there are two general types of policies for firewalls:

1. Open access policy: In this kind of access policy, all ports are open, except the ones mentioned in the black-list.
2. Closed access policy: In this policy, all ports are closed, except the ones that are directly specified as open (i.e., in the white-list).

The main reason that we have considered closed access policy firewalls is because in the real world networks the dominant policy for firewalls is closed access policy. In other words, administrators prefer to close all ports and explicitly allow incoming traffic to certain ports.

The second layer of defense is IPS. The IPS does a signature matching on the received packets. If the packet is matched, the appropriate alert will be generated and based on the skill level specified for the administrator, a suitable prevention mechanism will be employed. If no match is found, the packet will be sent to the server. After the reception of packet, the server processes the packet and generates the response for the sender (based on HTTP Request/HTTP Response mechanism which is used in the Internet).In the following, we are going to describe the sub-models constructed for each element.

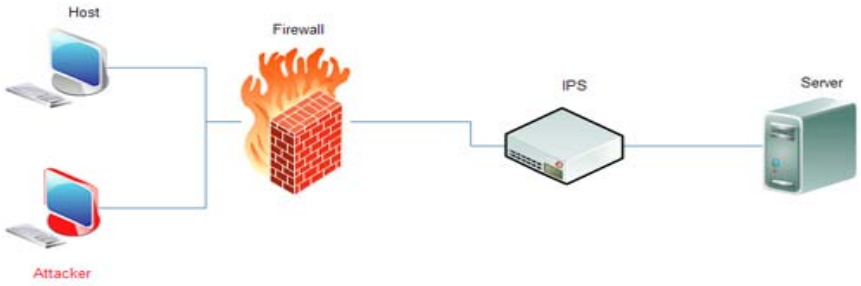

Figure 2. Basic network for modeling the attack process

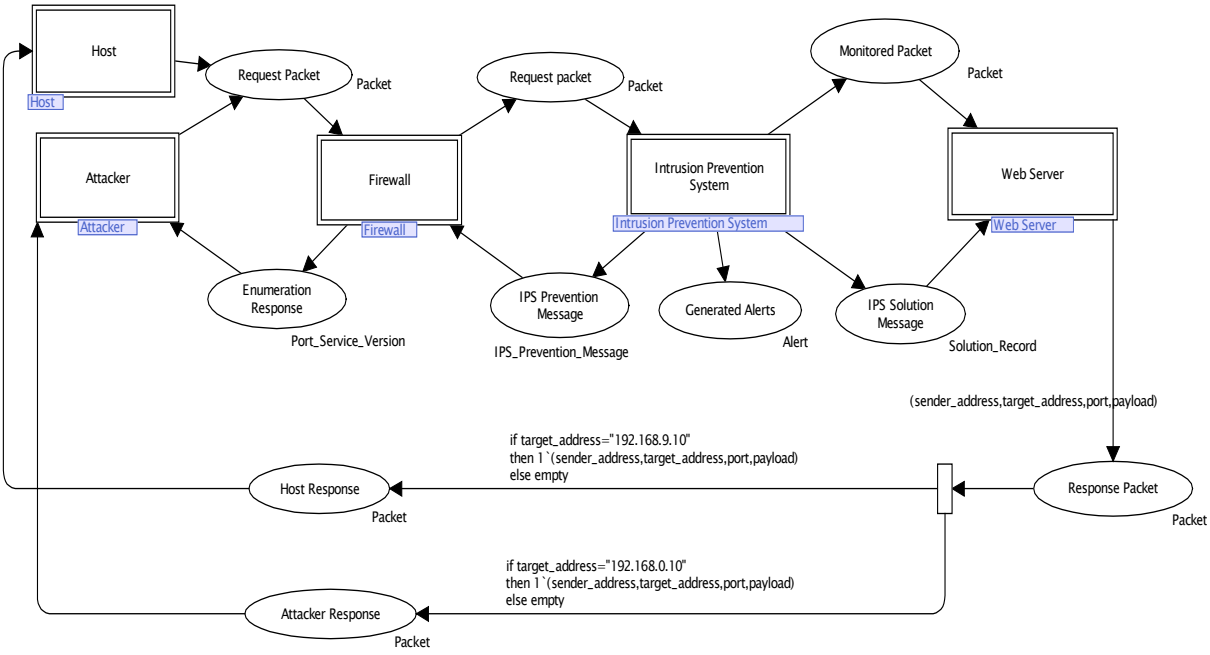

Figure 3. The model created for the network of Figure 2

A. *the sub-model for the host*

As shown in Figure 4, the host computer generates packets based on exponential distribution over time. The requested packets are placed in Send Request Packet place and the response packets are received in Receive Response Packet place. The structure of the packets is shown in Figure 5. This structure only stores the relevant information in attack modeling.

B. *the sub-model for the attacker*

For modeling attackers, three levels of skill (low, medium and high) are considered. The skill parameter is an important factor in exploit selection process. The success of an attack is highly dependent on the exploit selection method.

Before talking about the exploit selection method based on attacker skill, we introduce a categorization of exploits based on the exploits published in OSVDB [27]:

1. *Public exploits*: these exploits are widely accessible in exploit databases in the Internet. Complete information about these exploits and their usage is available to public.
2. *Commercial exploits*: these exploits are written by professional attackers or security teams and are commonly integrated in commercial vulnerability testing products. These exploits are

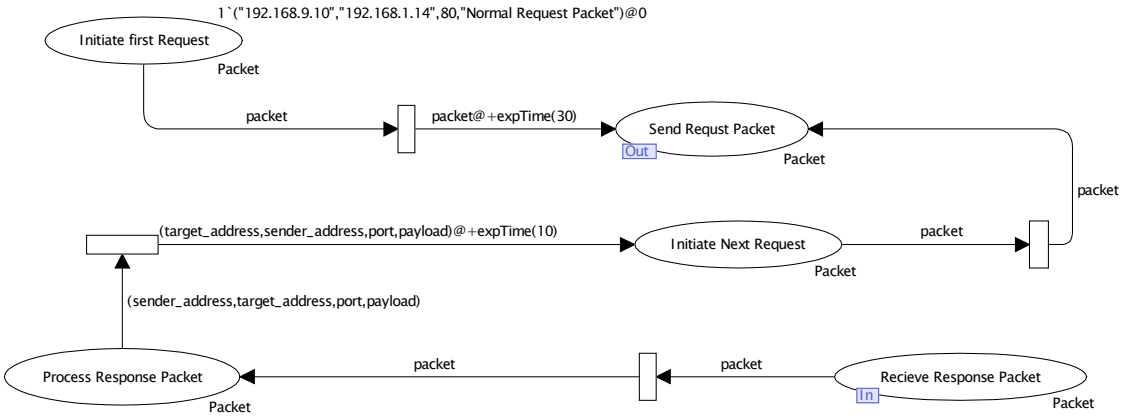

Figure 4. The CPN sub-model for the host

usually very effective and the information about them is very limited in public.

3. *Private exploits*: these exploits are written by very skilled attackers. These codes are commonly referred to as zero-day exploits, which are kept completely hidden from public and usually will remain in the underground world of hackers. Only after a while, these exploits will be released to public.

Each one of these exploits mentioned above will be selected by different attackers with different skills.

The skill categorization of attackers is based on the following definitions:

1. *Attacker with low skill level*: these attackers (commonly known as script-kiddies) usually do not have programming knowledge nor they are familiar with the steps required to gain access to their target systems. These attackers most of the times rely on blind use of exploits and tools written by professional attackers. Therefore, the two most important characteristics of these attackers are as follows:
   - Relying on public exploits, and
   - Random selection of public exploits.
2. *Attackers with medium skill level*: these attackers are partially familiar with professional hacking tools and the steps required gaining access to target machines. They are still not familiar with programming knowledge required to create new exploits based on vulnerabilities they find in their target. Therefore, the following characteristics are assumed for these attackers:
   - Completely familiar with the information necessary to gain access.
   - Completely aware of the steps required for a successful intrusion.
   - Familiarity with professional hacking tools and thus the ability to use commercial exploits.
   - No professional programming knowledge and thus not being able to create new exploits (private exploits).
3. *High-skilled attackers*: these attackers have a deep knowledge of vulnerability detection and writing new exploits. They are completely familiar with professional hacking tools and the steps required gaining access to their target systems. These attackers usually use their private archive of exploits. The summary of characteristics of such attackers is as follows:
   - Completely familiar with the information required for gaining access.
   - Completely familiar with the steps required for successful intrusion.
   - The ability to create zero-day exploits (private exploits).

The steps required to do a successful exploitation of the vulnerability is shown in Figure 6. These are the steps that medium-skilled and high-skilled attackers do in order to successfully gain access to their target systems. Selecting exploits based on operating system information or open ports and their respective services will drastically increase the possibility of success for these attackers. Using private exploits will give high-skilled attackers the ability to bypass signature based IDS/IPS devices.

Figure 5. The packet structure for the sub-model

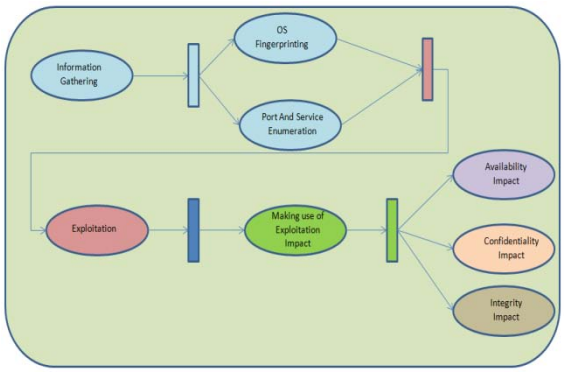

Figure 6. Common steps required to gain access to target systems by attackers

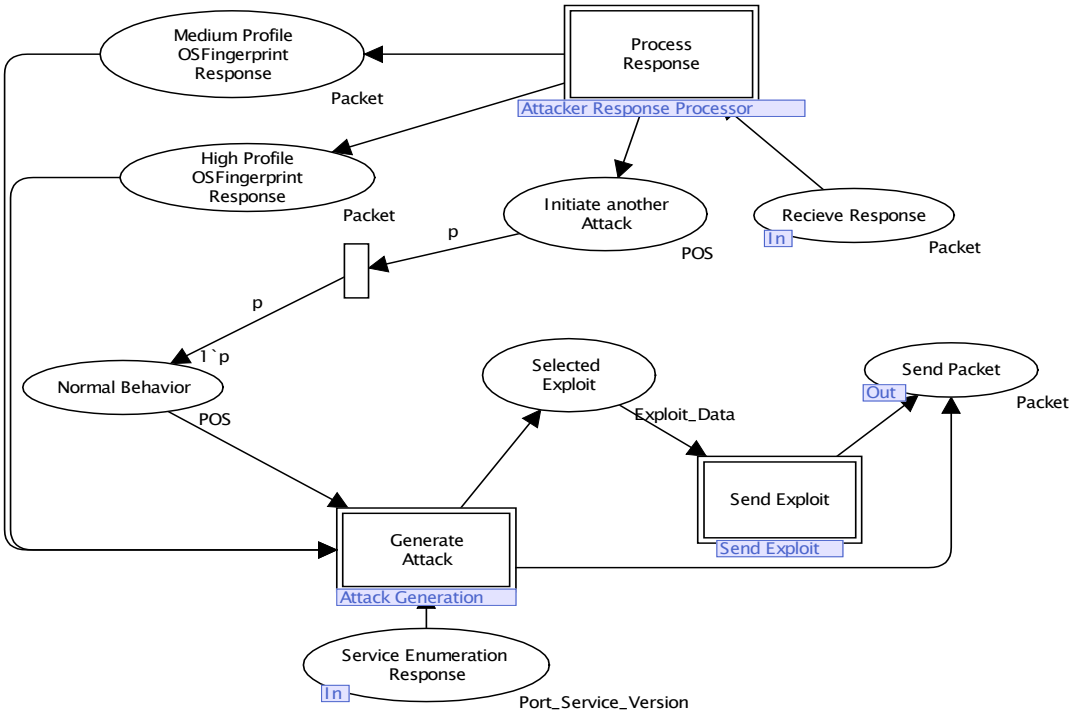

Figure 7. High-level model of attacker

Two main sections of the model are Attack Generation and Response Processing. In the Attack Generation section, based on the definitions of attackers skill provided in the previous part, three levels of skill is defined. One of the transitions based on the selected skill will be executed. The attack generation sub-model is shown in Figure 8.

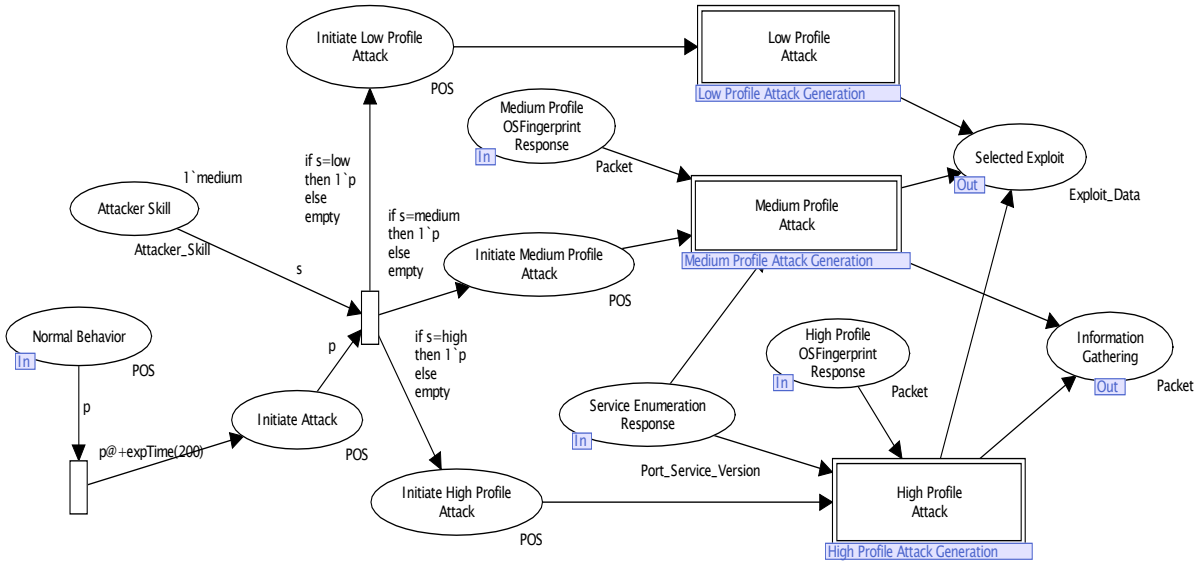

Figure 8. Attack generation sub-model

The modeling of the *Low Profile Attack* transition is based on the definition of low-skilled attacker. The modeling is based on random selection of public exploits. This is shown in Figure 9 and Figure 10.

The *LExploit_Set* initial marking is defined as below. The exploit colour set consists of exploit signature, exploit type (Public, Private and Commercial), Access Complexity, Operating System required for this exploit to be effective and the set of ports, service and service version this exploit is written for. The information about exploits is extracted from OSVDB.

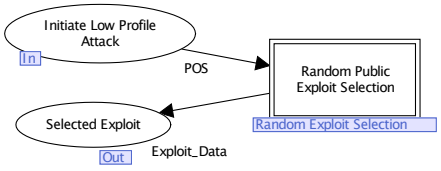

Figure 9. Low profile attack generation

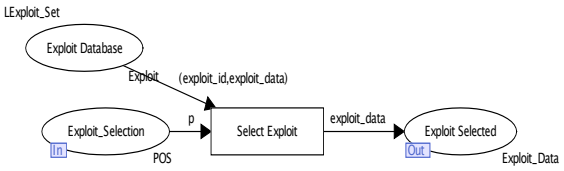

Figure 10. Low profile exploit selection

The following is part of the initial marking for the exploit database place of the low profile attacker:

```
val LExploit_Set=1`(1,("Microsoft IIS URL Redirection Malformed Length
DoS", Public, Low, "windows XP", (80,"IIS","5.0"))) ++
1` (2, ("IPTables FTP Stateful Inspection Arbitrary Filter Rule Insertion",
Public, Low, "Linux",   (21,"FTP","2.4"))) ++
1` (3, ("Eudora SMTP Server Reply Overflow", Public, Medium, “windows",
(25,"SMTP","2.5"))) ++
1` (4, ("RealPlayer Crafted .au File Handling Divide-By-Zero Application
DoS", Public, Low, “windows", (243,"RealPlayer","10.1"))) ++
1` (5, ("EasyMail Objects EasyMail.SMTP.6 ActiveX (emsmtp.dll)
AddAttachment Method Overflow", Public, High, "Linux",
(25,"SMTP","1.3.4"))) ++
1` (6, ("Microsoft IIS FTP Server Crafted Recursive Listing Remote DoS",
Public, Low, "windows", (21,"FTP","0.4"))) ++
1` (7, ("Apache HTTP Server Header Parsing Space Saturation DoS", Public,
Low, "Linux", (80,"Apache","2.0.4"))) ++
1` (8, ("Apache HTTP Server on Cygwin Encoded GET Request Arbitrary
File Access", Public, Low, "Linux", (80,"Apache","1.3.27"))) ++
1` (9, ("Microsoft IIS aexp2b.htr Password Policy Bypass", Public, Low,
"windows NT 4.0”, (80,"IIS","4.0")));
```

The colour set for exploits is defined as follows:

```
colset Exploit_Signature=string;
colset Exploit_ID=int;
colset Disclosure=with Public| Commercial| Private;
colset Access_Complexity=with High | Medium | Low;
colset Port=int;
colset Service=string;
colset Service_Version=string;
colset Port_Service_Version=product Port * Service * Service_Version
timed;
colset Exploit_Data= product Exploit_Signature* Disclosure
*Access_Complexity*Operating_System
Port_Service_Version;
colset Exploit= product Exploit_ID * Exploit_Data;
```

This is only a small portion of the exploits that can be defined in the public exploits set. But everything will be different for the attackers with medium and high skill levels. These attackers follow the required steps of every successful intrusion and based on the information that they have gathered they start selecting their exploits. This sub-model is shown in Figure 11.

For the exploit selection process in this level, all the information about the target system’s operating system, open ports, services and their respective versions are considered. The exploit will be selected based on this information. The only difference between medium skilled and high skilled attackers is in the set of exploits they can select. High skilled attackers have access to private exploits along with

commercial and public ones, but medium skilled attackers can only choose between public and commercial exploits. Figure 12 shows the sub-model for exploit selection in these levels of skill.

C. *The sub-model for intrusion prevention systems*

As stated before, for this framework, signature-based IDS/IPSs are modeled. The IPS checks whether each received packet's payload matches a signature in the database. If a match is found, the IPS can do several preventive tasks such as dropping or blocking the sender. The model constructed for IPS is shown in Figure 13. The IPS model consists of two main parts. One of them is Signature Matching and the other is Prevention. In the prevention part, different methods based on the skill of network administrator can be chosen to prevent attacks.

If the skill of administrator is low, then shutting down or resetting the server will be selected. In the medium skill level, the packet will be dropped and the sender's source address will be blocked by adding the address to the black list of firewall. In the high skill level, all the above actions can be done plus if a solution exists, such as a security patch, workaround or upgrade, then the solution will be provided to the server. In the signature matching part, we have tried to model Snort rule set format. The following is an example:

```
1` ("Microsoft IIS URL Redirection Malformed Length DoS", "WEB-IIS IIS URL Redirection DOS Attempt") ++
1` ("IPTables FTP Stateful Inspection Arbitrary Filter Rule Insertion", "FTP Arbitrary Filter Rule Inspection")
```

D. *The model for servers*

After the server receives an incoming packet, two things may happen. If the packet is normal, the server will generate appropriate response and will send the response to the sender. If the packet is malicious (a packet with an exploit in its payload), which has passed all the defensive elements in the network it should be determined whether it is effective on the server or not. This task will be done in Exploit infiltration success probability sub-model of the server model. If it is determined that the exploit is effective, then the exploitation stage will begin. The modeling of exploitation stage is done in Exploiting Stage sub-model. Finally, based on the extracted information of the received exploit, its impact on confidentiality, integrity and availability will be determined. The Solution Processor sub-model is responsible for the reception and processing of incoming solutions from IPS. The model of the server is shown in Figure 14.

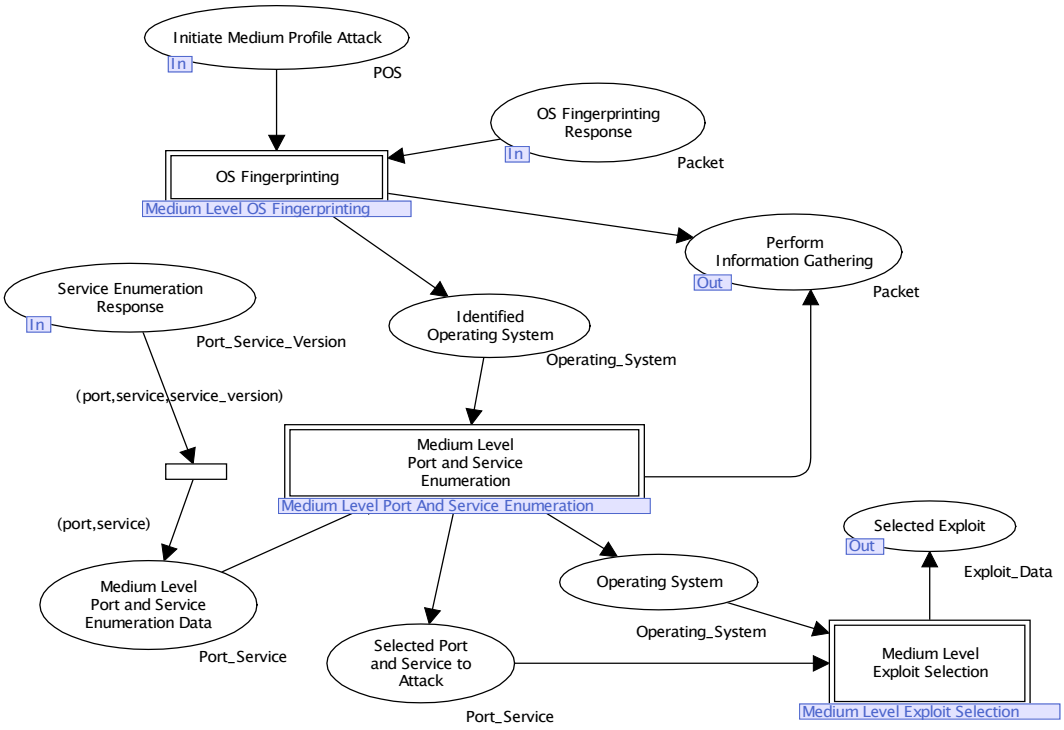

Figure 11. High-level modeling of attack process for medium level attackers

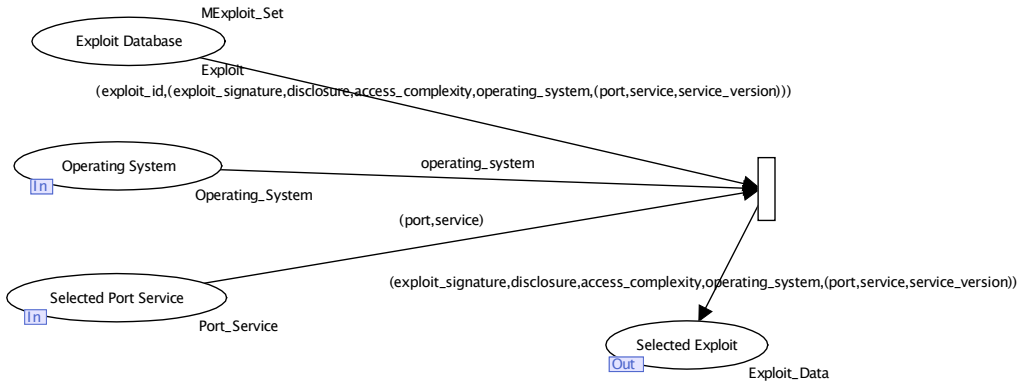

Figure 12. The exploit selection model for medium and high-level attackers

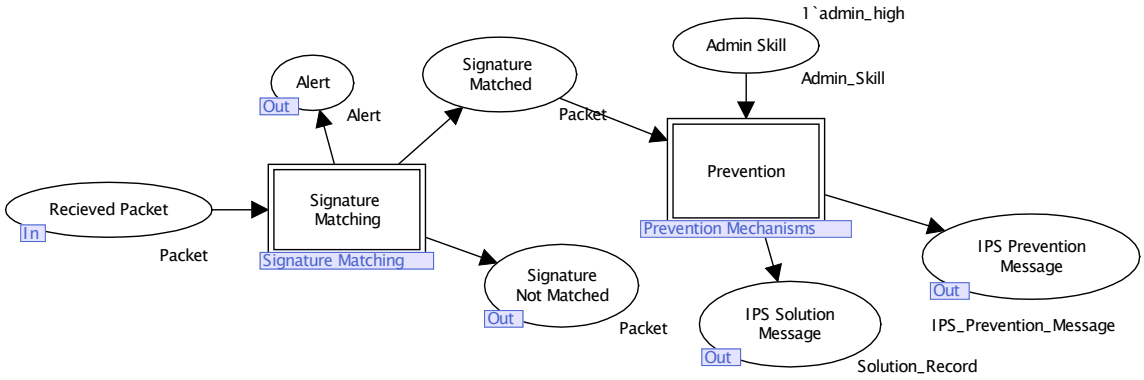

Figure 13. The IPS model

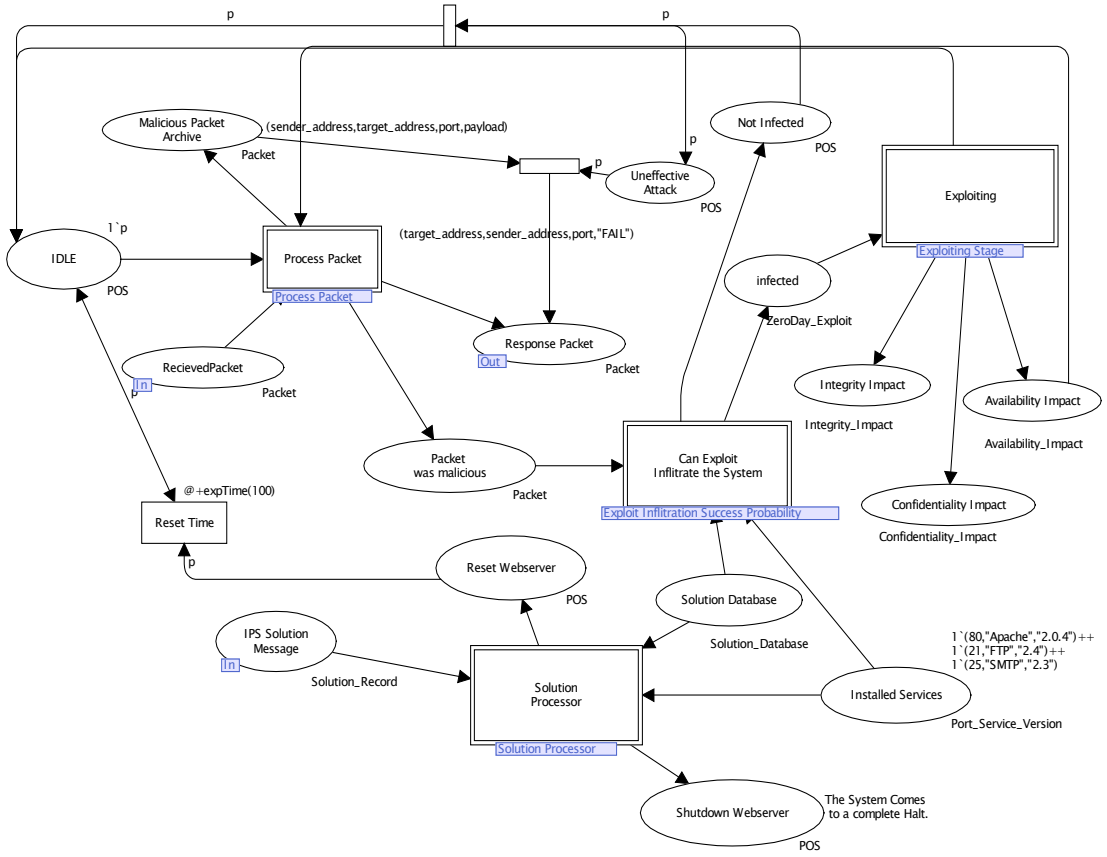

Figure 14. The high-level model of server

One of the most important parts of the model for the server is the Exploit infiltration success probability. For determining the possibility that the exploit is effective on the target system, many parameters should be considered. These parameters are operating system, open ports, installed services, service's version, the access vector of exploit and the solution set (including patches, workaround, upgrades) installed on the target system. The model created for this part is shown in Figure 15. After the reception of an incoming malicious packet, the following checks will be done:

1. Checking the access vector of the exploit: access vector (as categorized in OSVDB) can be Remote, Local, Physical Access required and Adjacent Network. For example, an exploit that has an Adjacent Network access vector can only be effective if it was sent from a machine of the same subnet as the target system.
2. Checking the operating system compatibility between the operating system required by the exploit to be effective and the operating system of the target system.
3. Checking the open ports and installed services (with their respective versions)
4. Checking the solutions installed for the targeted vulnerabilities in the target system. For example if the targeted vulnerability is patched, then the exploit will not be effective even if other parameters match.

After the exploit is determined to be effective, the exploitation stage will start. In this stage, based on the access complexity of the exploit it will take some time for the exploit to create its impacts. This parameter based on the extracted information from OSVDB is divided into low, medium and high categories. The model constructed for this stage is shown in Figure 16.

After the execution of exploit, with regard to its effect, one or more of the confidentiality, integrity or availability aspects of the target system will be completely or partially affected. The effect on the availability of the system will directly influence the operational measures of the system. (As stated before, integrity and confidentiality compromisation can have indirect effect on availability and thus the operational measures). The following definitions are presented in the CVSS scoring system for partial and complete availability impacts [28, 29]:

1. *Partial availability impact*: "There is reduced performance or interruptions in resource availability. An example is a network-based flood attack that permits a limited number of successful connections to an Internet service."
2. *Complete availability impact*: "There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable."

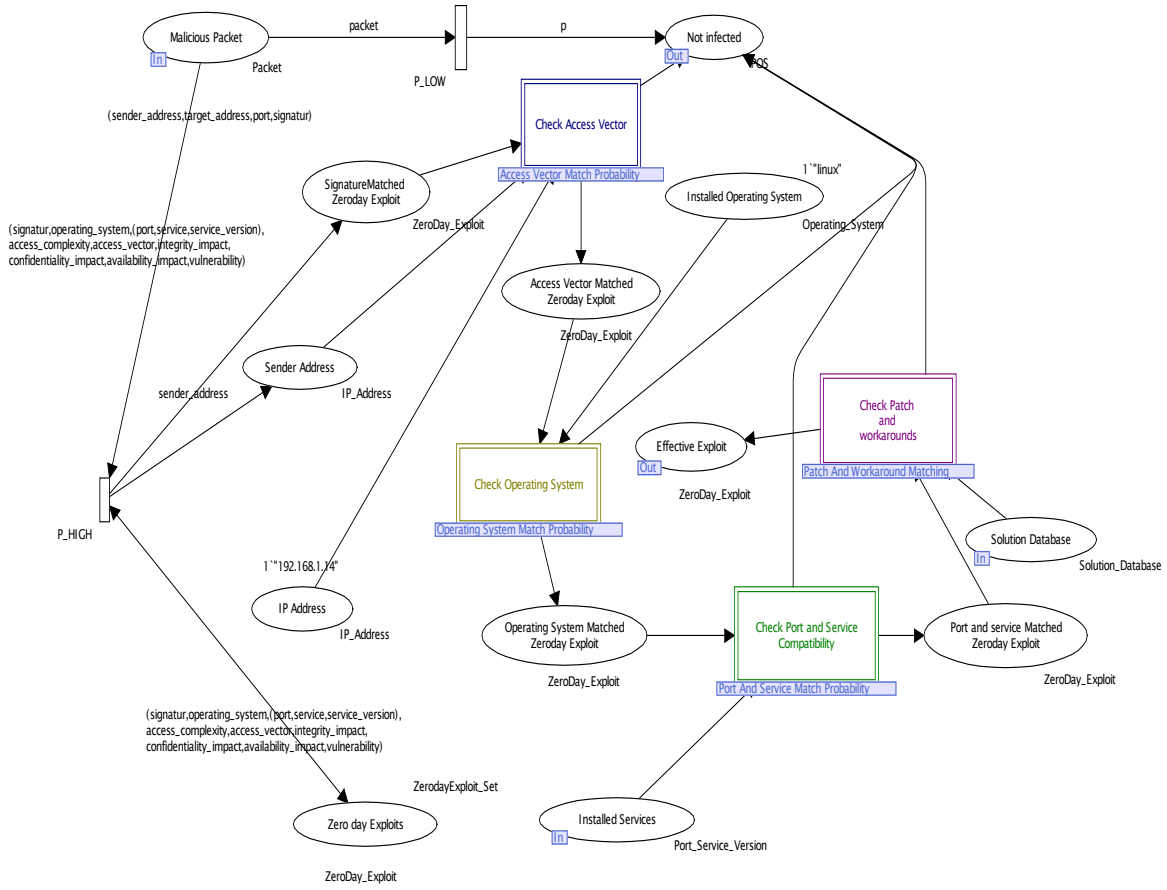

Figure 15. Exploit infiltration success probability model

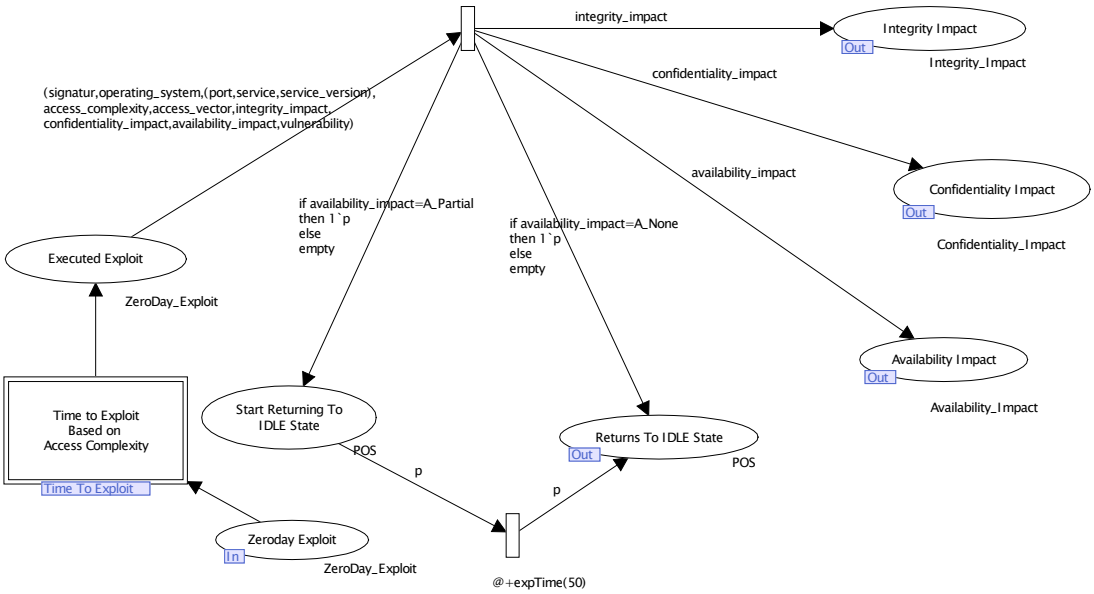

Figure 16. Exploitation stage model

Based on the above definition, the model shown in Figure 17 is presented for partial availability impact. After the partial availability impact, the server will enter to an ON/OFF process. This is shown in Figure 18. The server will be accessible in brief periods of time, whereas most of the time it is in the off state (denial of service state). The length of ON periods is randomly generated with an exponential distribution.

V. MODEL EVALUATION

For evaluating the model and computing the operational measures, we have to consider different scenarios. After selecting the scenario and configuration of the network elements, CPN Tool is used to simulate the model and with the help of data collectors, different measures are extracted. For each of the following scenarios, consider the network of Figure 2. The server's operating system is Linux and FTP, Apache and SMTP services are running on ports 80, 21 and 25, respectively. Firewall will only allow the traffics sent to the ports 21 and 80.

A. Different Attackers vs. Different Administrators

In this section we present the results of simulation for four scenarios. These scenarios are as follows:

- Low-skilled attacker vs. medium skilled admin
- Medium-skilled attacker vs. medium skilled admin
- High-skilled attacker vs. medium skilled admin (without any patches)
- High-skilled attacker vs. high-skilled admin (with incremental patching)

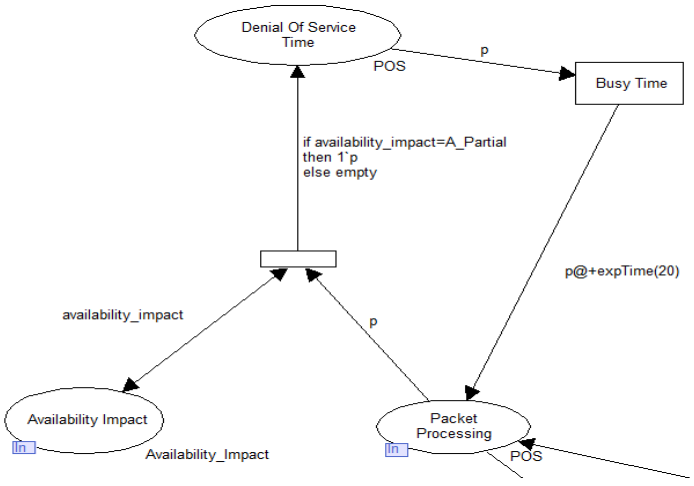

Figure 18. Modeled ON/OFF process for partial availability impact

Figure 17. Modeling partial availability

Access Vector	Access Complexity	Authentication	Confidentiality	Integrity	Availability
Local	High	Multiple Instance	None	None	None
Adjacent Network	Medium	Single Instance	Partial	Partial	Partial
Remote	Low	None	Complete	Complete	Complete
1.0	0.71	0.704	0.0	0.275	0.0

Figure 19. A sample exploit's information extracted from OSVDB [26]

In the first scenario, attacker has access to a set of public exploits. All the information about these exploits is extracted from OSVDB. For example, in Figure 19 some of the information about "Microsoft IIS aexp2b.htr Password Policy Bypass" exploit extracted from OSVDB is shown.

The results of repeating the simulation for 10 times for the first scenario are presented in Figure 20. The 10 repetitions are for gaining more dependable results. Because, the exploits are blindly selected and are almost always public exploits all the packets will be either dropped by firewall or blocked by IPS. The results of the second, third and fourth scenarios are shown in the Figures 21, 22 and 23, respectively. In the fourth scenario, it is assumed that the server is without any patches or upgrades in the initial state. As the simulation continues, the IPS will provide solutions for vulnerabilities and thus preventing future attacks against those.

B. *Availability Analysis*

One of the purposes of attack modeling is the evaluation of the impacts of successful attacks on different machines on the network. Some of the useful measures that can be calculated for these kinds of attacks are (but not limited to): service time of servers, availability of servers or hosts, queue length on servers, response time of servers, waiting time of hosts, etc. In figures 24 through 26, some of the results taken from an attack that has partial availability impact are presented.

Another important measure for network administrators is the packet queue length fluctuations during the attack. In order to have more realistic results, we consider the network shown in Figure 27. The constructed model for this network is shown in Figure 28.

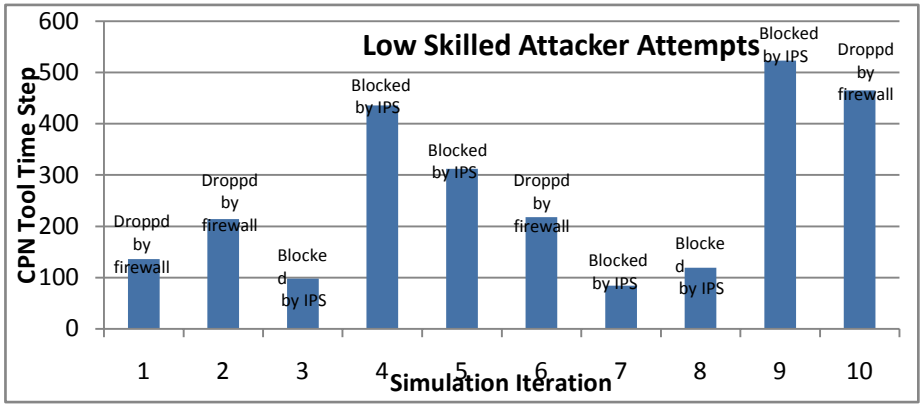

Figure 20. Low-skilled attacker vs. medium-skilled admin

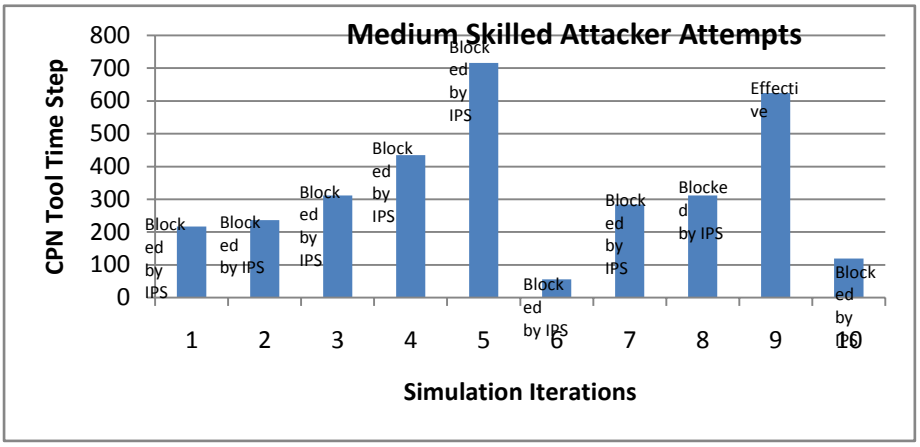

Figure 21. Medium-skilled attacker vs. medium-skilled admin

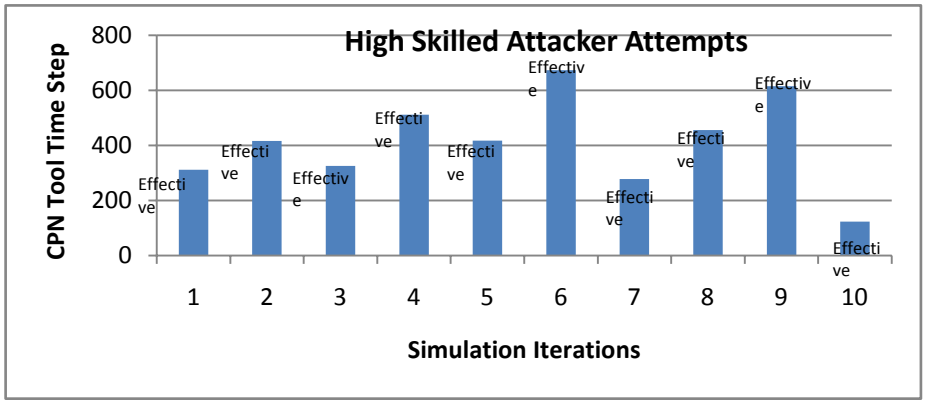

Figure 22. High-skilled attacker vs. medium skilled admin (without any patches)

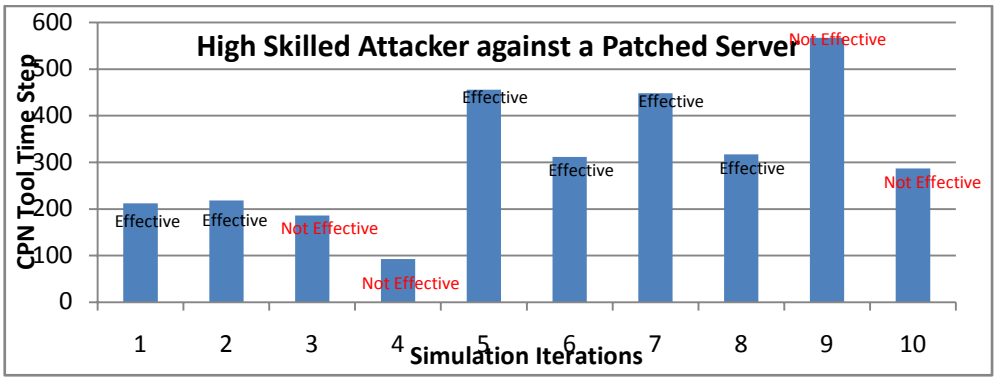

Figure 23. High-skilled attacker vs. high-skilled admin (with incremental patching)

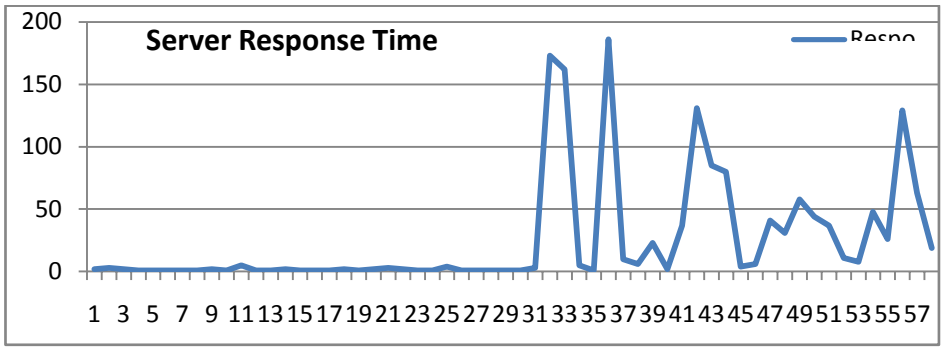

Figure 24. Server response time during an attack that has partial impact on availability

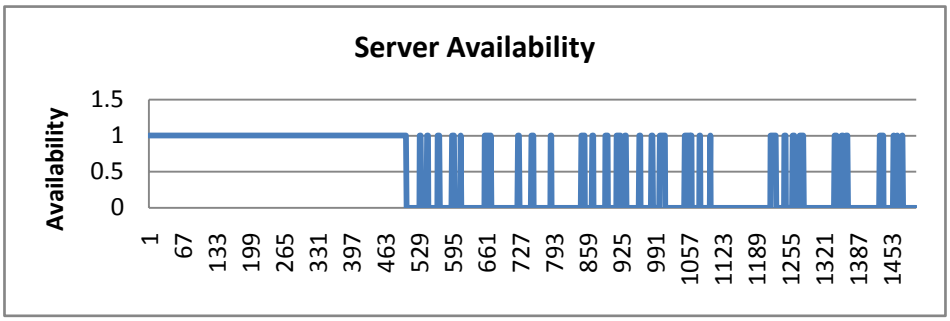

Figure 25. Server availability during an attack that has partial impact on availability

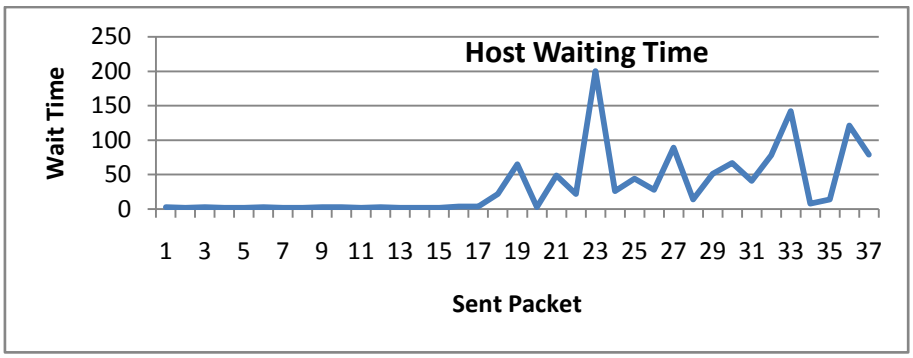

Figure 26. Host waiting time during an attack that has partial impact on availability

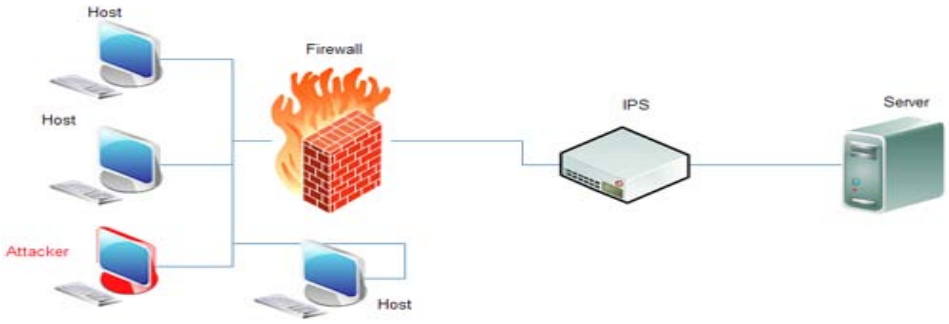

Figure 27. Network with three hosts

The queue length measure during an attack which has partial availability impact is shown in Figure 29. The same measures for an attack that has complete availability impact are presented in the Figures 30 through 33. (Like before, for the packet queue length measure, the network shown in Figure 27 is considered).

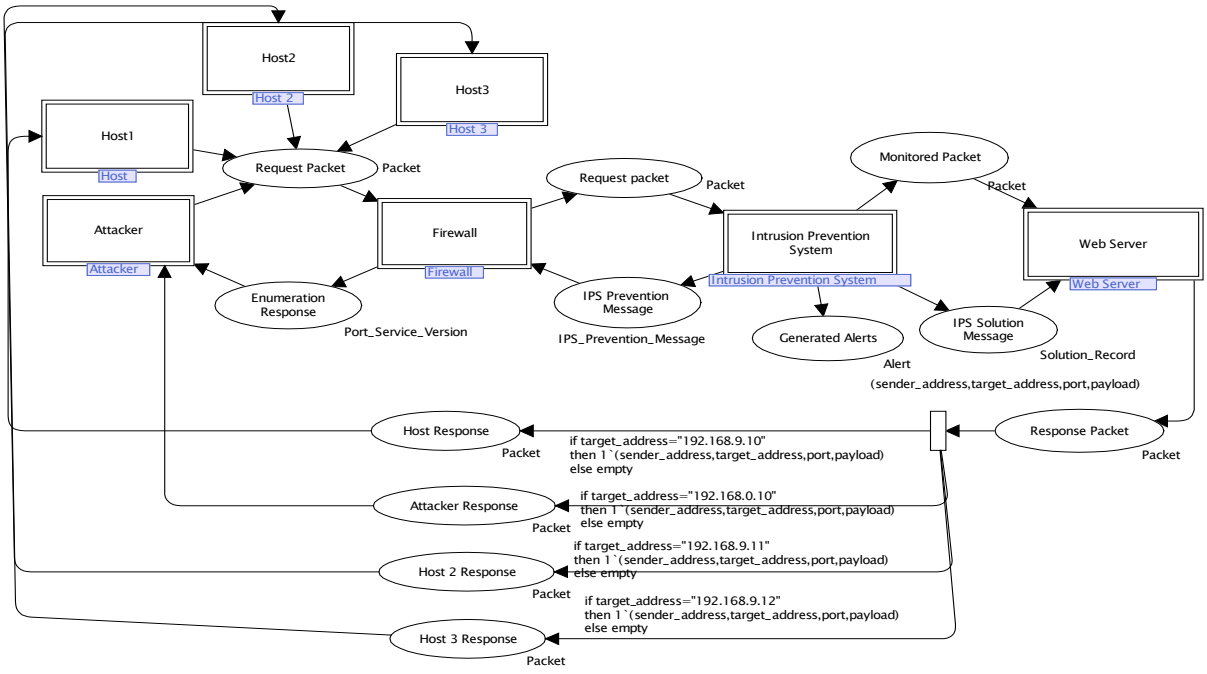

Figure 28. The model for the network shown in Figure 27

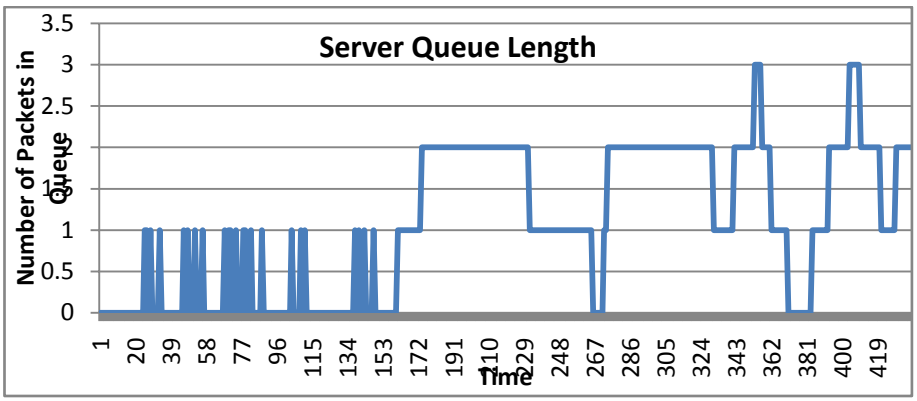

Figure 29. Server queue length during an attack that has partial availability impact

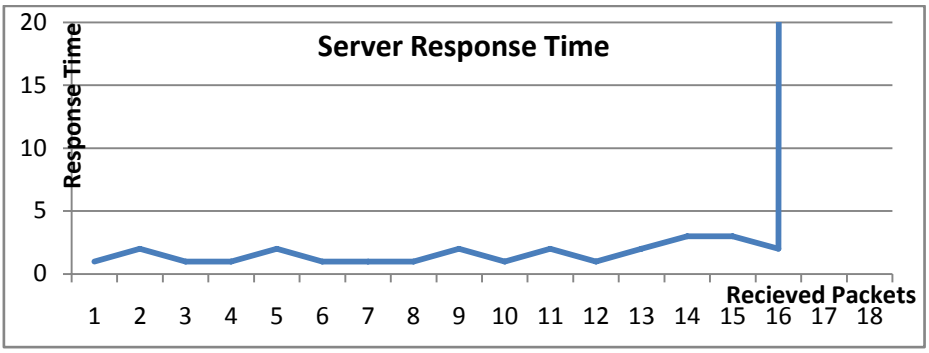

Figure 30. Server response time during an attack that has complete availability impact

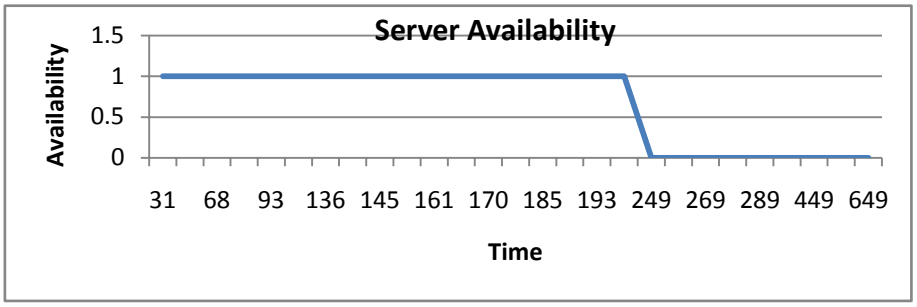

Figure 31. Server availability during an attack that has complete availability impact

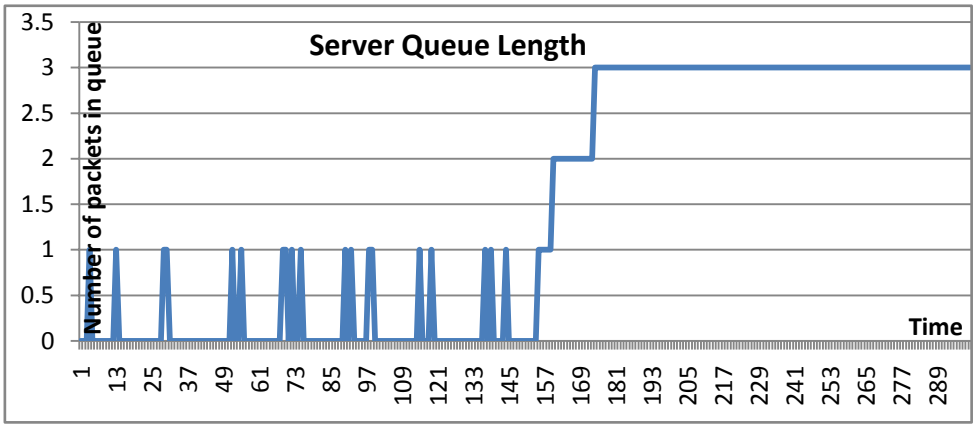

Figure 30. Server queue length during an attack that has complete availability impact

VI. Advantages and Disadvantages of the Proposed Framework

The benefits of the proposed simulation framework in comparison with existing works in the domain are as follows:

1. Simulation of attack processes as close as possible to the real world. Many existing works in this field considers only very high-level and abstract attacking processes. For example, they divide the attacking process into multiple steps with different statically assigned probabilities without taking into account the real steps that attackers take.

2. Considering different skill levels for attackers. Based on different skills of attackers, the process of intrusion will differ greatly. For example, a very unskilled attacker (commonly known as script kiddies or pink hat hackers) relies mostly on tools written by others without completing the required steps of a successful attack (steps such as the information gathering, enumeration, etc.). This will reduce the chance of success. The same statement is correct about medium-skilled attacker that cannot develop his own custom exploits (known as zero-day exploits). One of the purposes of our work has been to simulate the skill level of attackers and the effect of skill on the success of the attempts.

   Using real world payloads. One of the drawbacks of the existing simulative approaches of attack process modeling is the usage of toy data and toy scenarios, with no (or less) real world information. This will lead to imprecise results. One of the benefits of the proposed simulation framework is the ability to use real world information in the simulation process. In other words, the framework takes into account important detailed parameters that may make or break an attack. These parameters are as follows: (a) Exploit's access vector, (b) Exploit's required operating system, (c) Exploit's required services and service versions, and (d) Open ports, and the patch level of the target system.

3. Considering defensive elements in the simulation environment. Many existing simulative approaches in this context do not take into account the presence of defensive elements, such as firewalls and IDSs. This will create an unrealistic simulation environment. We have tried to simulate these elements, too. However, in some cases this is done in a simple manner, e.g. the anomaly-based IDSs are not considered. But, in using IDS in the simulation, we have tried our best to simulate it as close as possible to the real world. For this purpose, we have used the Snort real-world signatures and have fed them into the simulation, in order to have a precise simulation behavior.

4. considering different (not ON/OFF) impacts for the exploitation process. Many simulations in this context have considered the success of an attack like an ON/OFF process whether the attack is successful or not and whether the system is compromised or not. In real world (as mentioned in most of the vulnerability databases, such as OSVDB) the impacts of exploits are different. An exploit may have a partial availability impact. Or it may have a complete integrity impact. These impacts are also modeled in our simulations. For example, a system that has a complete integrity impact can be used by the attacker as a stepping point to reach other systems. Or a system that has a partial availability impact may follow a randomly distributed ON/OFF periods in its availability.

5. Modularization of simulation entities. One of the benefits of the proposed simulation framework is the possibility of the creation of high-level simulation entities with the use of hierarchies of coloured Petri nets that can be used very easily by end users. In opposite, one of the drawbacks of many existing simulation works is the very complex process which makes it difficult for a real world user to make something out of (i.e., no abstraction is used to make transparent the complicated calculations). In the proposed framework, users can drag and drop high-level entities and just wire them together in order to simulate their network. In an existing work which has used hierarchical coloured Petri nets (HCPNs) [19, 20, 21], there is no real use of the abstraction that these hierarchies can provide.

The drawbacks of the proposed framework are as follows:

1. The feeding process of real world data extracted by OSVDB and Snort is not automatic and thus is time consuming. This will result in a large amount of time spending on entering exploit's data, IDS signatures, open ports, installed services and operating system information in the model.

2. The creativity of the attackers in their attack is not considered. Usually the attack process by skilled attackers consists of a normal phase (which is simulated in detail) plus a creative phase. In the creative phase, attackers usually use very sophisticated activities that are very hard to model. This is a very common drawback of all existing simulative approaches.

3. The anomaly-based IDSs are not modeled in the current simulation framework. The reasons behind this are twofold. First, these systems are not main stream in commercial world and real world networks. Currently the most famous intrusion detection systems, such as Snort, are completely signature-based. Anomaly-based IDSs are still very much in the research and development phase and are not that mature to be used in real networks due to their very large false alarm rates and the challenges regarding building

the "Normal Model of traffic". Second, simulation of anomaly-based IDSs requires a very detailed research in order to propose its simulation behavior. This was out of the scope of our work. In other words, we were not concerned about how IDSs work or how are their performance in comparison to other types of IDSs. We mainly focused on attack process taken by the attacker. Although we have considered the simple and common forms of defensive elements, such as closed access policy firewalls and signature-based IDSs. This may be the topic of a future work for the proposed simulation framework.

VII. CONCLUSIONS

In attack modeling the security of computer systems and networks, many parameters should be considered. For example, the impacts of the selected exploits are different and should be categorized and weighted. Modeling the attacking process with hierarchical coloured Petri nets has many advantages over other existing approaches. The flexibility, scalability and reusability models and sub-models and the ability of modeling in different levels of abstraction are some of the advantages of CPNs.

In this paper, the important elements of computer networks involved in cyber attacks, such as hosts, attackers, intrusion detection and prevention systems, servers and firewalls have been modeled as reusable CPN sub-models. In other words, with the help of hierarchy and the abstraction provided by CPNs, we have tried to propose a framework for modeling and operational security evaluation of the impacts of cyber attacks on computer networks. Different networks, equipment and attack scenarios have been modeled by integrating CPN sub-models in the framework. By some illustrative examples, we have modeled sample networks and different attack scenarios using the exploit information extracted from open source vulnerability database (OSVDB). This has made the models to be parameterized using real-world information. By using the simulation features of CPN Tools to evaluate the model, some challenges of other formal approaches, such as state space explosion, do not exist in the proposed approach.

In this work, we have concentrated on the evaluation of availability. As a future work, we are adding some patterns and the necessary elements for evaluation of integrity. For example, DNS Spoofing is going to be added to this framework as an attack against the integrity of a DNS machine (this attack will indirectly lead to availability impact, too).

ACKNOWLEDGMENT

This research was supported by National Research Institute for Science Policy (NRISP) of Iran.

REFERENCES

[1] K. Jensen, Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science. Springer-Verlag, 1997.

[2] K. Jensen, Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 2, Analysis Methods. Monographs in Theoretical Computer Science. Springer-Verlag, 1997

[3] K. Jensen, Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 3, Practical Use. Monographs in Theoretical Computer Science. Springer-Verlag, 1997.

[4] "CPN Tool Home Page", URL: http://www.cpntool.org , visited: 2011/08/06

[5] M. Kuhl, J. Kistner, K. Costantini and M. Sudit, "Cyber attack modeling and simulation for network security analysis," Proc. of the 39th Conference on Winter Simulation, vol. 1, NJ, USA, 15-Dec, pp. 1180-1188, 2007.

[6] "Snort open source network intrusion detection and prevention system", URL: http://www.snort.org, visited: 2011/08/06.

[7] R. Ritchey and P. Amman, "Using model checking to analyze network vulnerabilities," Proc. Of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, May, pp. 156-165, 2000.

[8] R. Hewett, and P. Kijsanayothin, "Host-centric model checking for network vulnerability analysis," Proc. of the 24th Annual Computer Security Applications Conference, Washington, DC, USA, December 2008, IEEE Computer Society, pp. 225-234.

[9] V. Saini, Q. Duan and V.Paruchuri, "Threat modeling using attack trees," *Journal of Computing Science in Colleges,vol. 23, Issue.4,* pp. 124–131, 2008

[10] L.Wang, T. Islam, T.Long, A.Singhal and S.Jajodia, "An attack graph-based probabilistic security metric," Proc. of the 22nd Conference on Data and Application Security, vol. 5094, London, UK, 13-Jul, pp. 283-296, 2008.

[11] A. Xie, G. Chen, Y. Wang, Z. Chen, and J. Hu, "A new method to generate attack graphs," Proc. Of the 3rd IEEE International Conference on Secure Software Integration and Reliability Improvement, Shanghai, China. July 2009, pp. 401-406.

[12] J. Almasizadeh and M. Abdollahi Azgomi, "Intrusion process modeling for security quantification," Proc. of the 4th International Conference on Availability, Reliability and Security (ARES'09), March 16-19, Fukuoka Institute of Technology (FIT), Fukuoka, Japan, IEEE CS Press, 2009, pp. 114-121

[13] B. R. Haverkort, "Markovian models for performance and dependability modeling," Formal Methods and Performance Analysis (FMPA'00), LNCS, E. Brinksma, H. Hermanns, and J. P. Katoen, eds. vol. 2090, pp. 38-83, Springer, 2001.

[14] K. Sallhammar, S. J. Knapskog and B. E. Helvik, "Using stochastic game theory to compute the expected behavior of attackers," Proc. of the 2005 International Symposium on Applications and the Internet (Saint 2005).Trento, Italy, January 31 - February 4, 2005.

[15] K. Sallhammar and S. J. Knapskog, "Using game theory in stochastic models for quantifying security," Proc. of the 9th Nordic Workshop on Secure IT-Systems (NordSec 2004), Espoo, Finland, November 4-5, 2004.

[16] D. M. Nicol, W. H. Sanders and K. S. Trivedi, "Model-based evaluation: from dependability to security," IEEE Transactions on Dependable and Secure Computing, vol. 1, issue 1, pp. 48-65, Jan 2004.

[17] B.Haverkort, R.Marie, G.Rubino, and K. S Trivedi, "Performability modeling tools and techniques," Chichester, England: John Wiley & Sons, 2001.

[18] K. Sallhammar, B. E. Helvik and S. J. Knapskog, "On stochastic modeling for integrated security and dependability

evaluation," Journal of Networks, vol. 1, no.5, Sept/Oct 2006.

[19] S. Zhou, Z. Qin, F.Zhang, X. Zhang, W.Chen and J. Liu, "Coloured Petri net based Attack Modeling," Proc. of the 9th International Conference on Rough Sets, Data Mining and Granular Computing, vol. 2639, Chongqing, China, 26-May, pp. 583 2003.

[20] R. Wu , W. Li and H.Huang , "An attack modeling based on hierarchical coloured Petri nets," Proc. of the International conference on computer and electrical engineering.ICCEE, vol. 1, Phuket, Thailand, 20-Dec, pp. 918-921, 2008.

[21] S. H. Houmb and K. Sallhammar. "Modeling system integrity of a security critical using coloured Petri nets," Proc. of the 1st Inte'l Conf. on Safety and Security Engineering, Rome, Italy, June 13-15, 2005.

[22] R. Bye, S. Schmidt, K. Luther, and S. Albayrak, "Application-level simulation for network security". In Proc. of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, pp.113-127, 2008

[23] M. Liljenstamand, J. Liuand, D. Nicoland, Y. Yuanand, G. Yanand and C. G. Rinse, "The real-time immersive network simulation environment for network security exercises," In Proc. Of the Workshop on Principles of Advanced and Distributed Simulation, pp.119-128, 2005.

[24] A. Futoransky, F. Miranda, J. Orlicki and C. Sarrute, "Simulating cyber-attacks for fun and profit," Proc. Of the 2nd International Conference on Simulation Tools and Techniques, Brussel, Belgium,pp.415-425, 2009

[25] "NetSim: A Distributed Network Simulation to Support Cyber Exercises," In Proc. Of the Huntsville Simulation Conference, Huntsville, pp.110-116, 2004.

[26] "The simulation of attack and defence in OPNET environment", In Proc. Of the Computer Design and Applications Conference, pp. 312-315, 2010.

[27] "Open Source Vulnerability Database", URL: http://www.osvdb.org , visited: 2011/06/08

[28] "Common Vulnerability Scoring System", URL: http://www.first.org/cvss/ , visited: 2011/06/08

[29] S. Scarfone, and P. Mell, **"**An analysis of CVSS version 2 vulnerability scoring**,"** Proc. of the 3rd International Symposium on Empirical Software Engineering and Measurement, Washington, DC, USA, 2009, IEEE Computer Society, pp. 516-525.

Mehrdad Ashtiani received his B.Sc. and M.Sc. degrees in computer science from Iran University of Science & Technology (2008 and 2010, respectively), Tehran, Iran. His main research interests include network security andcomputer simulation. He has published several papers in international conferences.
Mr. Ashtiani is currently a Ph.D. student at the School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

Mohammad Abdollahi Azgomi received his B.Sc., M.Sc. and Ph.D. degrees in computer engineering (software) (1991, 1996 and 2005, respectively) from Sharif University of Technology, Tehran, Iran. His research interests include network and software security, and dependability and security modelling. He has published several papers in international conferences and journals.
Dr. Abdollahi Azgomi is currently a faculty member at the School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.