Composition of Cyberattack Models
Katia P. Mayfield
Dept. of Mathematics, Computer, and Natural Sciences, Athens State University
Athens, AL 35611
Mikel D. Petty, John A. Bland, and Tymaine S. Whitaker
Dept. of Computer Science, University of Alabama in Huntsville
Huntsville, AL 35899

Abstract
Cyberattacks are threats that every system developer and administrators need to be familiar with. The use of Petri Nets to model cyberattacks has been shown to provide additional knowledge on the planning stages of defense systems. There have been several proposed extensions to a Petri Net model to accommodate the simulation of these attacks. Petri Nets with Players, Strategies, and Cost was recently proposed to model individual cyberattacks on systems. This study introduces the formalism required to compose individual Petri Nets with Players, Strategies, and Cost models from a single system attack to a full system, which may include different methods of attacks being attempted, modeling a more realistic situation. The model composition described includes the sequential and parallel possibilities of multiple attacks. An example of a possible attack scenario is described in order to demonstrate the practical application of the current results.

Keywords: cyberattacks, Petri Nets, composition, models, replication

1. Introduction
There have been multiple areas of research conducted to study security threats. Knowledge of cyber attacks has been labeled as instrumental for the design of secure software [1]. Most cyberattacks have been identified to follow a pattern, which allows a model to be created to represent them in order to analyze their evolution and impact. Not only is the knowledge of these attacks important, but it is also essential to try to prevent them and understand how the attacker proceeds. MITRE Corporation developed a report called the Common Attack Patterns Enumeration and Classification (CAPEC) which documents over 400 different attacks, how they are performed, their goals, and mitigations [2].

Each attack described in the CAPEC reports can be performed by an individual independently from any other attack. However, some of the goals of different attacks can be achieved through different methods, i.e. there is more than one way to try and gain privileged information associated to a system. An attacker may prefer to run one attack over another due to their own knowledge of being able to control the process, however since he or she has options, these must be considered. The process of designing a model with the use of Petri nets, containing the strategic involvement of attackers and defenders, was demonstrated in a previous publication under the title Petri Nets with Players, Strategies and Cost (PNPSC) [4].

In a study not related to cyber security, Hamadi and Benatallah proposed a Petri net based algebra for modeling the composition of Petri nets representing Web services control flow [3]. They describe the semantics of algebra operators used to combine the Petri nets. Even though their research was geared towards Web services, the authors provided a general theoretical framework for Petri net composition.

Similar to Web services, where different services are combined to provide a user with whatever they need, the study of cyber security must assume that an attacker may also be able to combine different attack strategies to achieve their goal. Therefore, this study focuses on the composition of Petri nets with players, strategies and cost used to model different views of cyberattacks. The next section provides an overview of modeling attacks with Petri nets and a brief review of the operations proposed by Hamadi and Benatallah. Follows a discussion on the basic operations that will be initially used to allow for different cyberattack Petri nets to be composed into a larger model and the formal definitions of such operations. Section 4 provides an example of composing two different well-known attack models and is followed by a summary of this study.

2 Background
Petri nets are models that are mathematically developed (compatible) with a graphical representation [5]. They have been used to model
different applications since 1962. Petri nets implemented for this study are not basic, they take into consideration players and strategies (PNPS) where an attacker can make decisions, and a defender is also included since there are actions that can be detected and mitigated. Details of PNPS can be found in previous publications [4, 6]. A brief overview follows next.

A standard Petri net is formally defined as a 6-tuple $PN = (P, T, W, L, M_0, B)$, where
- $P = \{p_1, p_2, \ldots\}$; finite, non-empty set of places
- $T = \{t_1, t_2, \ldots\}$; finite, non-empty set of transitions
- $W = W_i \cup W_o$, where $W_i \subseteq (P \times T)$; set of arcs from places to transitions and $W_o \subseteq (T \times P)$; set of arcs from transitions to places
- $L: W_i \rightarrow Z^+ \cup \{-1\}$; the count (or weight) for each arc representing the number of tokens that can be transmitted, the value -1 implies a blocked arc from a place to transition
- $M_0: P \rightarrow Z^+ \cup \{0\}$; initial marking of tokens in places, with $0 \leq M_0(p) \leq B(p)$ for every $p \in P$
- $B: P \rightarrow Z^+ \cup \infty$; upper bound on tokens per place

The definition of Petri nets was the basis for the design of PNPS nets, which are formally defined as an 8-tuple $PNPS = (PN, G, \Theta, O, F, \Omega, \Gamma, \mathcal{X})$, where
- $PN$: standard Petri net, as defined above
- $G$: finite, non-empty set of players
- $\Theta$: partition of transition set $T$ into non-controlled and player-controlled transitions
- $O$: partition of place set $P$ into places observable by each player
- $F$: firing rates for non-player-controlled transitions
- $\Omega$: max firing rates for player-controlled transitions
- $\Gamma$: mapping from observable markings to firing rates for player-controlled transitions
- $\mathcal{X}$: mapping from marked end places to firing rate

PNPS was later expanded to allow cost analysis of defending against cyberattacks, resulting in a Petri net with Players, Strategies, and Cost, which is defined as a 3-tuple $PNPSC = (PNPS, \mathcal{F}, D)$
- $PNPS$: Petri net with Players, and Strategies as presented above.
- $\mathcal{F}: T \rightarrow Z^+ \cup \{0\}$; cost for firing a transition
- $D: T \rightarrow \emptyset(G)$; power set of players, identifying all possible subsets of players charged for fired transitions.

In this study, CAPEC reports are used as the formal documentation of the attacker’s process and mitigation techniques that are represented within the PNPSC design. The focus areas of the CAPEC report include, but are not limited to, the summary of the attack which describes a scenario and how the attack works, the attack steps which can consist of three phases (explore, experiment, and exploit), attack prerequisites, solutions and mitigations, the motivation and consequences behind the attack. In a previous study [4], the steps that are taken to design a basic Petri net and a PNPSC are discussed in detail using CAPEC report 66, SQL Injection as the example.

### 2.1 Composition
Hamadi and Benatallah described a BNF grammar with eight ways that Petri nets can be composed. They defined Petri nets composition operations based on Web service models containing a single start state and one final state. Their definitions allowed the Petri nets to be composed by using the following operations, (1) sequential, (2) alternatively, (3) arbitrary sequence, (4) iteratively, (5) in parallel with communication taking place, (6) discriminatorily, (7) through dynamic selection, or (8) refinement [3]. Such operations can be briefly described as:
- sequential: where a Petri net follows another
- alternatively: when two Petri nets can be applied and one of them is activated
- arbitrary sequence: similar to sequential except that there is no definition of which Petri net is the first one to be executed
- iteratively: representing the repetition of a Web service multiple times
- in parallel with communication: concurrent execution of two Petri nets with possible information exchange
- discriminatorily: execution of the same task in different Web services providers (what should be equivalent to the concurrent execution of the same Petri net with different parameters)
- dynamic selection: multiple Petri nets are source to a selection process, which is similar to the alternatively composition with the added selection process
- refinement: which allows a Petri net operation be replaced by a more in-depth representation of its behavior

The algebra and composition operations proposed by Hamadi et. al. were used as the conceptual basis for the cyberattacks composition described in section 3.
3 Cyberattack Composition

This study focuses on two composition methods, sequential and parallel or concurrent, which are initially assumed as the most common ways of combining different cyberattacks. Sequential would imply a possible failure of an attack technique (as a consequence of the attackers actions and not a defense mechanism) followed by a different approach. Parallel would represent several attacks occurring concurrently.

In order to define the composition operations, Petri nets are expanded to include two subsets of the final states: successful attacks and unblocked failures. Blocked failures imply a defensive action that would eliminate the possibility of the attacker being successful, becoming an end point in the Petri net.

**Definition 1.** An expanded standard Petri net is formally defined as a 8-tuple EPN = (P, T, W, L, M₀, B, ES, EF), where

- P = \{p₁, p₂, ...\}; finite, non-empty set of places
- T = \{t₁, t₂, ...\}; finite, non-empty set of transitions
- W = W₁ ∪ W₀ set of arcs from places to transitions and transitions to places, where W₁ ⊆ (P × T); set of arcs from places to transitions and W₀ ⊆ (T × P); set of arcs from transitions to places.
- L: \text{transmit tokens, } L: W₁ → Z⁺ \{ -1 \}; the count (or weight) for each arc representing the number of tokens that can be transmitted, the value -1 implies a blocked arc from a place to transition
- M₀: \text{initial marking of tokens in places, } 0 \leq M₀(p) \leq B(p) \text{ for every } p \in P
- B: \text{upper bound on tokens per place}
- ES = \{s₁, s₂, ...\}; finite set of successful attack end places
- EF = \{f₁, f₂, ...\}; finite set of unblocked failed attack end places

With sequential composition, the model represents an attack being attempted until the attacker succeeds or fails to fulfill the attack. In this scenario, once the attacker fails, it is assumed that he would attempt a different type of attack. While with the web services composition the design was based on one service being dependent on a previous service, in cyberattacks the sequential composition is based on the success or failure of the attacker’s actions. Therefore the composition for the new PNPSC is defined as:

**Definition 2.** Given two PNPSC N₁ and N₂ defined by

\[
N₁ = (EPN₁, G₁, Θ₁, O₁, F₁, Ω₁, Γ₁, X₁)
\]

and EPN₁ = (P₁, T₁, W₁, L₁, M₀₁, B₁, ES₁, EF₁)

\[
N₂ = (EPN₂, G₂, Θ₂, O₂, F₂, Ω₂, Γ₂, X₂)
\]

and EPN₂ = (P₂, T₂, W₂, L₂, M₀₂, B₂, ES₂, EF₂)

The resulting PNPSC N₁ and N₂ is defined by

\[
N = N₁ \rightarrow N₂ = (EPN₁ \cup G₂, Θ₁ \cup Θ₂, O₁ \cup O₂, F, Ω₁ \cup Ω₂, Γ, X)
\]

With EPN defined as:

EPN = (P, T, W, L, M₀, B, ES, EF) such that

- P = P₁ ∪ P₂
- T = T₁ ∪ T₂ ∪ \{t₁₂\}
- W = W₁ ∪ W₂ ∪ \{(u,t₁₂) | u \in EF₁ \} ∪ \{(t₁₂,v) | v \in M₀₂\}
- L = L₁ ∪ L₂ ∪ \{L(u,t₁₂) = 1 | u \in EF₁\}
- M₀ = M₀₁
- B = B₁ ∪ B₂
- ES = ES₁ ∪ ES₂
- EF = EF₂

and F, Γ, X defined as:

F: F₁ ∪ F₂
Γ: \text{defender controllable transitions, } \Gamma(M₀₁ ∪ O₂) \rightarrow \mathbb{R}
X: \text{attacker controllable transitions, } X(M₀₂ ∪ ES₂) \rightarrow \mathbb{R}

Figures 1 and 2 represent two different generic cyberattack PNPSC. Transitions that are outlined with a double line are defender controllable transitions; transitions created with a dashed border are attacker controllable transitions. For ease of reading, places that are outlined with a dashed line are unblocked failure places, those that are outlined with a double line are blocked failure places, and those with a thicker black outline are successful end places.

![Figure 1 Diagram of PNPSC₁](image-url)
For a sequential composition of these two PNPSC, the attacker will first go through PNPSC$_1$ and if the attacker succeeds then there is no continuation to the second PNPSC. However, if the attacker fails because he or she is blocked by a defender then they are no longer capable of attempting any attack. In other words, action had been taken against the attacker that prevents them of continuing their offensive. Figure 3 displays the design of PNPSC$_1$ and PNPSC$_2$ being composed based on a sequential operation. If the attacker fails for any other reason, then he/she will have the opportunity to continue to the second attack, PNPSC$_2$. Therefore when composing an attack PNPSC, there will be an outgoing arc added to the unblocked fail places of PNPSC$_1$ that will lead to a transition that allows the attacker to continue with the sequential attack. The transition will fire and a token will be found in a place that is part of the pre-requisite requirements for the next attack to be able to begin, this transition is identified by the arrow that points to it from the left side in Figure 3. Within the second attack, there is still the opportunity that the attacker will either succeed, be blocked, or fail.

With parallel composition, the composed model represents both attacks occurring at the same time resulting in either success, attacker failing, or the attacker being blocked. With the attacks occurring at the same time, the pre-requisite start places will be combined. The end places for this net will also be unionized. Therefore, the new PNPSC resulting from a parallel composition is presented in definition 3.

**Definition 3.** Given two PNPSC $N_1$ and $N_2$ defined by:

$N_1 = (EPN_1, G_1, \Theta_1, O_1, F_1, \Omega_1, \Gamma_1, X_1)$

and $EPN_1 = (P_1, T_1, W_1, L_1, M_0_1, B_1, ES_1, EF_1)$

$N_2 = (EPN_2, G_2, \Theta_2, O_2, F_2, \Omega_2, \Gamma_2, X_2)$

and $EPN_2 = (P_2, T_2, W_2, L_2, M_0_2, B_2, ES_2, EF_2)$

The resulting PNPSC $N$ of a parallel composition of $N_1$ and $N_2$ is defined by

$N = N_1 \parallel N_2 = (EPN, G_1 \cup G_2, \Theta_1 \cup \Theta_2, O_1 \cup O_2, F, \Omega_1 \cup \Omega_2, \Gamma, X)$

With $EPN$ defined as:

$EPN = (P, T, W, L, M_0, B, ES, EF)$ such that

$P = P_1 \cup P_2$

$T = T_1 \cup T_2$

$W = W_1 \cup W_2$

$L = L_1 \cup L_2$

$M_0 = M_0_1 \cup M_0_2$

$B = B_1 \cup B_2$

$ES = ES_1 \cup ES_2$

$EF = EF_1 \cup EF_2$

and $F, \Gamma, X$ defined as:

$F: F_1 \cup F_2$

$\Gamma: M(O_1 \cup O_2) \rightarrow \mathbb{R}$

$X: M(ES_1 \cup ES_2) \rightarrow \mathbb{R}$

Figure 4 represents a PNPSC where two individual PNPSC’s are running in parallel. The start places for the composed PNPSC will be the start points of PNPSC$_1$ and PNPSC$_2$. This assumption allows for the union of the start places. Also, the final end states are unionized so that they are not redundant.
The operations and examples illustrated in figures 3 and 4 have been defined based on the composition involving only two attacks. To conduct a precise analysis one would want to make sure that the PNPSC created would include any and all possible attacks that target known vulnerabilities of their system. These compositions can then be recursively applied to as many attacks as necessary.

4. EXAMPLE

In a simple example of the application of PNPSC composition, a cyber security consultant would be evaluating the vulnerability of a system, which had been already analyzed and assumed susceptible to cyberattacks targeting SQL. The consultant also knows that CAPEC 63 (Cross Site Scripting) and CAPEC 66 (SQL Injection) present two different cyberattacks that have been modeled using PNPSC, and have similar attacker end goals.

The consultant could start its study by looking at two possibilities. One assuming the attacker would try a SQL injection technique and only try a different approach after failing the first one. The other case could be the attacker trying both approaches concurrently (similar also to two attackers working independently with different techniques). In order to conduct such study, the consultant would retrieve the two modeled PNPSCs from a library and compose them according with the possibility being investigated.

Case 1: attacker will try SQL injection first.

The SQL Injection was chosen by the consultant to be the first attack because it has one additional ultimate attacker goal that Cross Site Scripting does not have. However, if the attacker is willing to perform sequential attacks it can be assumed the goal of the attacker is one that is found in common between both attacks. Therefore the consultant can come to a hypothesis that the attacker is attempting to achieve one of three objectives, (1) execute unauthorized code or commands, (2) modify application data, or (3) read application data. A fourth goal, not common between both attacks, is gaining privileges or assuming an identity. Since this is uncommon between the attacks, one could hypothesize that this is not the attacker’s ultimate goal. This case is equivalent to compose the two PNPSC using the sequence operator.

Composed = PNPSC\(_1\) \(\Rightarrow\) PNPSC\(_2\)

In Figure 5, PNPSC for both attacks are partially represented with \(p_i\) representing places of the first PNPSC and \(q_i\) from the second. The place \(p_{20}\), in particular, is an “attacker is unsuccessful” place, while \(p_{21}\) is an ultimate fail place. If the attacker reaches place \(p_{20}\) he/she will have the opportunity to try the same attack over again, however the attacker must “ultimately fail” to be able to continue with the sequential attack.

Case 2: attacker will attempt both attacks at the same time

If the attacker has the appropriate resources or is not working alone. He may be able to run both attacks at the same time in parallel. SQL Injection and Cross Site Scripting do not share a common attack pre-requisite, therefore the parallel composition of these two attacks are independent of one another with no direct connection between the two PNPSC models, resulting

Composed = PNPSC\(_1\) \(\parallel\) PNPSC\(_2\)

Figure 6 shows the resulting PNPSC, partially displayed, which results from the expression above.
5 Summary
Cyber security has become a priority to governments and private organizations. The simulation of cyberattacks allows a security professional to evaluate the defense mechanisms and the evolution of the attackers’ strategy. Petri nets are being used to model cyberattacks described in CAPEC reports. This study reviewed some basic concepts on Petri nets and a brief description of the CAPEC reports. The possibility of multiple attack requires that such models be combined. In this study, a previously proposed Petri net based algebra was modified to represent the composition of a model consistent of multiple cyberattacks. Two composition operations were introduced: sequential, implying the continuation of an attack using a different technique and parallel, combining models of attacks that run concurrently. The two operations were formally defined. With the use of these formal definitions, any number of cyberattack models can be composed to represent a multiple attack. This study focused fully on the composition of the PNPSCs. Two examples, showing the composition of a cross script and a SQL injection attack were also discussed.

6 Future Works
Continued research is being conducted to identify additional composition methods and to compare the established formalism with other modeling methods. Additional research is also being done to display the PNPSC model of individual cyberattacks and composing them into a larger attack model.

7 References


