An Approach to Incorporating Uncertainty in Network Security Analysis

Hoang Hai Nguyen
hnguye11@illinois.edu

Kartik Palani
palani2@illinois.edu

David M. Nicol
dmnicol@illinois.edu

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

Attack graphs used in network security analysis are analyzed to determine sequences of exploits that lead to successful acquisition of privileges or data at critical assets. An attack graph edge corresponds to a vulnerability, tacitly assuming a connection exists and tacitly assuming the vulnerability is known to exist. In this paper we explore use of *uncertain graphs* to extend the paradigm to include lack of certainty in connection and/or existence of a vulnerability. We extend the standard notion of uncertain graph (where the existence of each edge is probabilistically independent) however, as significant correlations on edge existence probabilities exist in practice, owing to common underlying causes for dis-connectivity and/or presence of vulnerabilities. Our extension describes each edge probability as a Boolean expression of independent indicator random variables. This paper (i) shows that this formalism is maximally descriptive in the sense that it can describe any joint probability distribution function of edge existence, (ii) shows that when these Boolean expressions are monotone then we can easily perform uncertainty analysis of edge probabilities, and (iii) uses these results to model a partial attack graph of the Stuxnet worm and a small enterprise network and to answer important security-related questions in a probabilistic manner.

1. INTRODUCTION

As computers become more ubiquitous in critical infrastructures, evaluating the effect of vulnerabilities becomes increasingly important. In order to make decisions about defense measures, it is vital to study the paths that an attacker might take to intrude into a target network and disrupt services. The attack graph formalism [16] is a representation of the possible ways in which an attacker can get to the desired target host by exploiting vulnerabilities on network hosts while gaining the required privileges at each step. The first step in attack graph generation is analyzing the connectivity of the network components and is termed as reachability analysis [11]. This information is used to determine if a target host is reachable by an attacker from his current host. Ideally, information about the network topology of the target network, applications running on network hosts, access control rules for the network, and the trust relationships between hosts is known to the modeler. Accuracy and exhaustiveness of network configuration information directly affects accuracy of the generated attack graph [14].

Despite being a useful and well-developed tool, attack graphs have deterministic semantics and hence are not capable of expressing uncertainty [18], which is inherent to any model. To our interest, uncertainty arises mainly from three sources: the uncertainty about the attacker (e.g. his skill set, goal, and knowledge), about the system being modeled (e.g. the versions and configuration details of network services and their associated vulnerabilities), and about the environment in which the system is operated (including the legitimate users and administrators). In each category, uncertainty may also come in different shapes, either due to the lack, inadequacy, inaccuracy, or sometimes inconsistency of information. Ideally we should be able to both use an attack graph to identify possible pathways of attack, but also quantify uncertainty about those pathways.

This paper aims to integrate uncertainty into security modeling and analysis of computer systems. As a first step, we choose to focus only on studying uncertainty about the system. Hence, uncertainties about the attacker and the environment (and their implications) will not be considered. Under this assumption, we propose to use uncertain graphs, graphs where potential edges are labeled with an existence probability. Uncertain graphs have been successfully applied to various problems in different domains [2] [26] [8] [9]. We use it to analyze uncertainty of the existence of stepping stone attacks encoded in data structures like attack graphs. However, the usual definition of uncertain graph assumes edges exist independently of each other [10] [12] [13], a major issue when applying to security modeling, e.g., as one vulnerability may simultaneously enable attacks from multiple hosts. Furthermore, existing uncertain graph research does not consider the precision of connectivity subjected to changes (or uncertainty) in edge existence; in other contexts, uncertainty analysis tell us in what cases, a robust conclusion can be made in the face of model input uncertainty.

A major portion of this paper aims to address those two issues. For the first issue, we extend the uncertain graph formalism and model the correlation between edge existence due to common underlying causes. We describe common causes using independent indicator random variables and use Boolean expressions of these to express the edge existence probabilities. For the second issue, we show how uncertainty analysis of uncertain graphs can be easily done when the Boolean expressions are monotone [4], i.e. they do not use negation of random variables. In summary, our contributions are fourfold:

1. To the best of our knowledge, we are the first to propose uncertain graphs for security modeling and anal-

ysis of system with uncertainty.

2. We extend the traditional uncertain graph formalism to model the correlation between edge existence and prove theoretical results about the expressiveness of uncertain graphs.

3. We perform uncertainty analysis of uncertain graphs by leveraging the monotonicity of reachability.

4. We show how to use uncertain graphs to model systems with uncertainty and how the graphs help answering different security-related questions about the modeled systems in a probabilistic manner.

The rest of the paper is organized as follows: Section 2 discusses background, Section 3 extends the uncertain graph formalism and prove some theoretical results, Section 4 performs uncertainty analysis of uncertain graphs, Section 5 and 6 show two modeling examples, Section 7 discusses related works, and Section 8 concludes the paper.

2. BACKGROUND

Symbols	Definitions
V	set of vertices
s, t	vertices in V, start and end point of attack
n	size of V
E	set of edges
m	size of E
G	deterministic graph
$E(G)$	set of edges in G
Γ_V	set of all det. graphs with vertex set V
N	size of Γ_V which is $2^{n(n-1)}$
p	probability assignment vector
X	set of random variables
r	size of X
$\wedge, \vee, \neg$	logic operator AND, OR, NOT
q	function that assigns boolean exp. to edges
$\mathcal{G}$	uncertain graph (basic and extended)
$w_{G,\mathcal{G}}$	the probability of G in $\mathcal{G}$
f	stochastic mapping
$R_{s,t}$	reachability of deterministic graph
$\mathcal{R}_{s,t}$	reachability of uncertain graph
$[0,1]^m$	unit hypercube of dimension m
$\mathcal{H}_{p,\epsilon}$	hyperrectangle containing p

Table 1: Summary of notations

2.1 Attack graph and scenario graph

The operation of systems can be modeled to be in different states at different instants of time. While most states might be benign, there exist critical states that can lead the system to failure. A failure scenario is described as a sequence of events that violate a correctness property defined for the system. A scenario graph [24] is an exhaustive and succinct representation of all failure scenarios. A special case of the scenario graph is an attack graph.

An attack graph models the possible ways an attacker might get access to a critical asset by exploiting a set of vulnerabilities on the services running on the hosts. The vertices of the graph represent the privilege level of the attacker on the network host and the edges represent the vulnerability that the attacker could exploit [22]. Traditionally, teams of experts have looked at the services running on hosts to determine vulnerability information and have coupled this with network information, such as the connectivity of hosts, to build out these attack graphs. However, this information is not always readily available, which makes it important to account for uncertainty in the models.

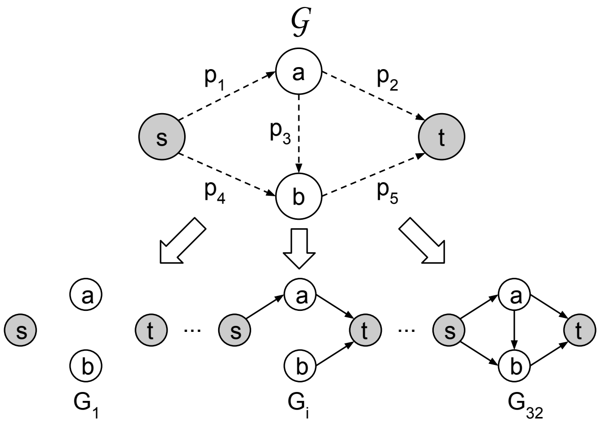

Figure 1: A 4-vertex, 5-edge uncertain graph and three of its 32 possible worlds. In security modeling, s denotes the starting point (e.g. a compromised computer in the network) and t the ending point of the attack (e.g. a critical computer that the attacker wants to gain access to).

2.2 Overview of uncertain graphs

Uncertain graphs extend the definition of deterministic graph by ascribing to each of a deterministic graph's edges an existence probability [19] [10] [13]. Formally, let $G = (V, E)$ denote a deterministic graph[1] where $V = \{V_1, \ldots, V_n\}$ and $E = \{E_1, \ldots, E_m\}$ are the set of vertices and edges. The uncertain graph $\mathcal{G} = (V, E, p)$, where $p = (p_1, \ldots, p_m) \in (0, 1]^m$, allows each edge $E_i \in E$ to exist independently of each other and with probability p_i for $i = 1, \ldots, m$. We call p the probability assignment vector of E. An uncertain graph may contain both certain edges – edges that exist with probability one – and uncertain edges – edges that exist with probability strictly less than one. When all edges are certain edges, the uncertain graph degenerates to a deterministic graph. In the literature, uncertain graphs are sometimes treated as generative models of deterministic graphs [19] [10]. With this view, every deterministic graph $G = (V, E')$ where $E' \subseteq E$ is called a possible world (or possible outcome) of $\mathcal{G}$. Slightly abusing the notation, we denote this as $G \in \mathcal{G}$. $\mathcal{G}$ generates an exponential number of $2^{m'}$ possible worlds, each $G = (V, E')$ with probability:

$$w_{G,\mathcal{G}} = \prod_{E_i \in E'} p_i \prod_{E_i \in E \setminus E'} (1 - p_i)$$

where $m' \leq m$ is the number of uncertain edges in $\mathcal{G}$. For example, the probability of G_i in $\mathcal{G}$ (Figure 1) is $w_{G_i,\mathcal{G}} = p_1 p_2 (1 - p_3)(1 - p_4) p_5$. Obviously $w_{G,\mathcal{G}} \in (0, 1]\ \forall G \in \mathcal{G}$ and the law of total probability dictates that $\sum_{G \in \mathcal{G}} w_{G,\mathcal{G}} = 1$.

[1] we only consider simple directed graphs

An uncertain graph distinguishes itself from a Bayesian network [27] [29], which was designed to model causal effects. While Bayesian networks are acyclic, cyclic relationship arises from many practical situations and is allowed in uncertain graphs. [27] circumvented the problem with cycles, but the technique had to rely on metric-dependent property. Uncertain graphs also do not assume the state transition modeled in transition systems [3] (e.g. Markov decision processes, probabilistic automata). Such transitions have a subtle drawback in security modeling of computer networks since an attacker does not "jump" from one place to the other. Instead, he gains access to more and more places as the attack progresses and is capable of showing up at multiple places at the same time.

2.3 Properties of uncertain graphs

For any given graph property, e.g., reachability from vertex s to vertex t, a deterministic graph has the property or does not have it. Since edges in an uncertain graph are random, we will speak of the probability that an uncertain graph has a given property, as the sum of the probabilities of graphs in its possible worlds that possess that property. We are particularly interested in reachability.

Using mathematical symbols, let function $R_{s,t}(G)$ denote the reachability of the deterministic graph G, which evaluates to 1 if s reaches t in G and to 0 otherwise. In Figure 1, $R_{s,t}(G_i) = R_{s,t}(G_{32}) = 1$ and $R_{s,t}(G_1) = 0$. The reachability of the uncertain graph $\mathcal{G}$ is defined as:

$$\begin{aligned}\mathcal{R}_{s,t}(\mathcal{G}) &= \sum_{G \in \mathcal{G}} w_{G,\mathcal{G}} \; R_{s,t}(G) \\ &= \sum_{G \in \mathcal{G}} \left(\prod_{E_i \in E(G)} p_i \prod_{E_i \in E \setminus E(G)} (1 - p_i) \; R_{s,t}(G) \right)\end{aligned} \tag{1}$$

where $E(G)$ denotes the set of edges in G. Using Equation 1, the reachability of the uncertain graph in Figure 1 can be computed as follows (after simplification):

$$\begin{aligned}\mathcal{R}_{s,t}(\mathcal{G}) = p_1p_2 + p_4p_5 + p_1p_3p_5 - p_1p_2p_3p_5 - \\ p_1p_2p_4p_5 - p_1p_3p_4p_5 + p_1p_2p_3p_4p_5\end{aligned}$$

Although we only focus on reachability in this paper, many other properties of uncertain graphs can be defined in a similar fashion.

2.4 Measuring uncertain graph properties

Most problems in uncertain graphs are #P-complete, including the problem of computing reachability [25]. For that reason, sampling techniques have been proposed as the alternative to direct computation in solving problems of large uncertain graphs [7] [19] [10] [13]. A basic Monte-Carlo method for estimating the reachability of an uncertain graph $\mathcal{G}$ works as follows. First, sample i possible worlds $G_1, \ldots, G_i$ from $\mathcal{G}$. This can be achieved by sampling edges in $\mathcal{G}$ independently according to their existence probabilities. Then, compute the reachability $R_{s,t}(G_j)$ for each G_j, $j = 1, \ldots, i$. The reachability of the uncertain graph is estimated as:

$$\widehat{\mathcal{R}}_{s,t}(\mathcal{G}) = \frac{1}{i} \left(\sum_{j=1}^{i} R_{s,t}(G_j) \right)$$

The estimator $\widehat{\mathcal{R}}_{s,t}(\mathcal{G})$ is a random variable whose mean is $\mathcal{R}_{s,t}(\mathcal{G})$ (for this we say the estimator is unbiased) and variance $\frac{1}{i}\mathcal{R}_{s,t}(\mathcal{G})(1 - \mathcal{R}_{s,t}(\mathcal{G}))$ [7] [10]. Advanced sampling techniques have been proposed to reduce the estimator variance while requiring fewer number samples [10] [13]. Those techniques recursively compute $\mathcal{R}_{s,t}(\mathcal{G})$ by conditioning on the existence of an edge.

3. EXTENDED UNCERTAIN GRAPHS

While a promising tool, the existing uncertain graph formalism however does not support modeling of the correlation between edge existence. Such correlation arises naturally from modeling various systems (Section 5 and 6). Here is an example. Assume in a certain network, host 0 and host 1 can freely communicate with all services running on host 1 and host 2, respectively. Furthermore, both host 1 and host 2 run a similar set of services. If an attacker from host 0 can gain access to host 1 by exploiting some vulnerability of a service running on host 1, then surely he is also able to do so from host 1 to host 2. As we model the possibility of attacks in the network using an uncertain graph, edge $(0, 1)$ existence guarantees that edge $(1, 2)$ also exists. In other words, there is no possible world in which edge $(0, 1)$ exists while edge $(1, 2)$ does not. This behavior cannot be modeled using the described uncertain graphs where edges exist independently of one another (Section 3.2). As the result, an altered and more powerful formalism is indeed required.

The layout of this section is as follows. First, we formally define the correlation between edge existence and extend the basic uncertain graph formalism to model such property (Section 3.1). Next, we show that modeling the correlation expands the expressiveness of basic uncertain graphs, in the sense that there exists an extended uncertain graph that has no equivalent basic uncertain graph (Section 3.2). Lastly, we prove that extended uncertain graphs can model an arbitrary stochastic mapping, making the two of them equivalent in term of expressiveness (Section 3.3).

3.1 Formal definition

Define $\mathcal{G} = (V, E, X, p, q)$ where $V = \{V_1, \ldots, V_n\}$ and $E = \{E_1, \ldots, E_m\}$ are the set of vertices and edges, $X = \{X_1, \ldots, X_r\}$ the set of independent Boolean random variables, $p = (p_1, \ldots, p_r) \in (0, 1]^r$ the probability assignment vector of X, i.e. $p_i = P[X_i]$ is the probability that X_i evaluates to true for $i = 1, \ldots, r$, and q the function that associates each edge $E_i \in E$ with a Boolean expression of the random variables in X for $i = 1, \ldots, m$. The existence probability of edge E_i is the probability that its associated Boolean expression evaluates to true, or $P[E_i \text{ exist}] = P[q(E_i)]$. We refer to this formalism the extended uncertain graph, in contrast with the basic uncertain graph $\mathcal{G} = (V, E, p)$ defined in Section 2.2. An example of an extended uncertain graph is shown in Figure 2. When the context is clear, we use the term uncertain graph to refer to both basic and extended uncertain graph (although their probability assignment vectors have slightly different meanings). Every basic uncertain graph $\mathcal{G} = (V, E, p)$ has an equivalent extended uncertain graph representation $\mathcal{G} = (V, E, X, p, q)$, which uses m random variables and $q(E_i) = X_i$ for $i = 1, \ldots, m$. The definition of basic uncertain graph properties (Section 2.3) and methods to estimate the graph properties (Section 2.4) apply to extended uncertain graph in a similar fashion.

If we consider uncertain graphs as generative models of

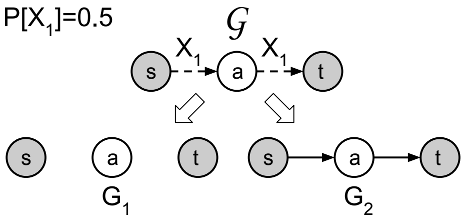

Figure 2: An extended uncertain graph and its only two possible worlds. This graph has no equivalent basic uncertain graph.

deterministic graphs, then each uncertain graph defines a mapping from the set of possible worlds to the unit interval $(0, 1]$. Let Γ_V denote the set of all deterministic graphs with vertex set V and $N = |\Gamma_V| = 2^{n(n-1)}$ the size of Γ_V (i.e. we consider all possible directed edges except loops). Define a mapping $f : \Gamma_V \to [0, 1]$ that associates with each deterministic graph $G \in \Gamma_V$ a real number $w_{G,\mathcal{G}}$ between 0 and 1. If the mapping f satisfies the condition $\sum_{G \in \Gamma_V} f(G) = 1$, then we call it a stochastic mapping [2]. A stochastic mapping is then a joint probability distribution function over the space of deterministic graphs whose edges are a subset of E. We express

$$f(G) = \begin{cases} w_{G,\mathcal{G}} & \text{if } G \in \mathcal{G} \\ 0 & \text{if } G \in \Gamma_V \backslash \mathcal{G} \end{cases}$$

and call f the equivalent stochastic mapping of $\mathcal{G}$ and denote that as $\mathcal{G} \equiv f$. Every uncertain graph has an equivalent stochastic mapping and two uncertain graphs are equivalent if they have the same stochastic mapping.

3.2 Expressiveness of basic uncertain graphs

In this subsection, we prove the following theorem:

THEOREM 3.1. *Extended uncertain graphs strictly expand the expressiveness of basic uncertain graphs, i.e. there exists an extended uncertain graph that has no equivalent basic uncertain graph.*

PROOF. We prove this theorem by giving an example. Consider the extended uncertain graph $\mathcal{G}$ in Figure 2. It has only two possible worlds G_1 and G_2 with $w_{G_1,\mathcal{G}} = 1 - P[X_1] = 0.5$ and $w_{G_2,\mathcal{G}} = P[X_1] = 0.5$. We will show that this extended uncertain graph has no equivalent basic uncertain graph representation. Define the basic uncertain graph $\mathcal{G}' = (V, E, p)$ where $V = (s, a, t)$, $E = ((s, a), (a, s), (a, t), (t, a), (t, s), (s, t))$ and p some probability assignment vector. The definition of $\mathcal{G}'$ captures all possible basic uncertain graphs that can be constructed from three vertices s, a, t. The probabilities of G_1 and G_2 in $\mathcal{G}'$ are:

$$\begin{aligned} w_{G_1,\mathcal{G}'} &= (1 - p_1)(1 - p_2)(1 - p_3)(1 - p_4)(1 - p_5)(1 - p_6) \\ &= (1 - p_1)(1 - p_3)Q \\ w_{G_2,\mathcal{G}'} &= p_1(1 - p_2)p_3(1 - p_4)(1 - p_5)(1 - p_6) \\ &= p_1 p_3 Q \end{aligned}$$

where $Q = (1 - p_2)(1 - p_4)(1 - p_5)(1 - p_6)$ and $0 < Q \leq 1$. Assume by contradiction that $\mathcal{G}'$ produces the same stochastic mapping as $\mathcal{G}$, or equivalently $w_{G_1,\mathcal{G}'} = w_{G_2,\mathcal{G}'} = 0.5$, we have: $0 = w_{G_1,\mathcal{G}'} - w_{G_2,\mathcal{G}'} = (1 - p_1)(1 - p_3)Q - p_1 p_3 Q = (1 - p_1 - p_3)Q$. Since $Q > 0$, $1 - p_1 - p_3 = 0$ or $p_1 + p_3 = 1$. Moreover, $1 = w_{G_1,\mathcal{G}'} + w_{G_2,\mathcal{G}'} = (1 - p_1)(1 - p_3)Q + p_1 p_3 Q = (1 - p_1 - p_3 + 2p_1 p_3)Q = 2p_1 p_3 Q$. Since $Q \leq 1$, $2p_1 p_3 = 1/Q \geq 1$. Combine this with $p_1 + p_3 = 1$ we have $(p_1 + p_3)^2 - 2p_1 p_3 \leq 1^2 - 1 = 0$ or $p_1^2 + p_3^2 \leq 0$, therefore $p_1 = p_3 = 0$. This solution does not satisfy $p_1 + p_3 = 1$, hence no basic uncertain graph equivalent to $\mathcal{G}$ exists. □

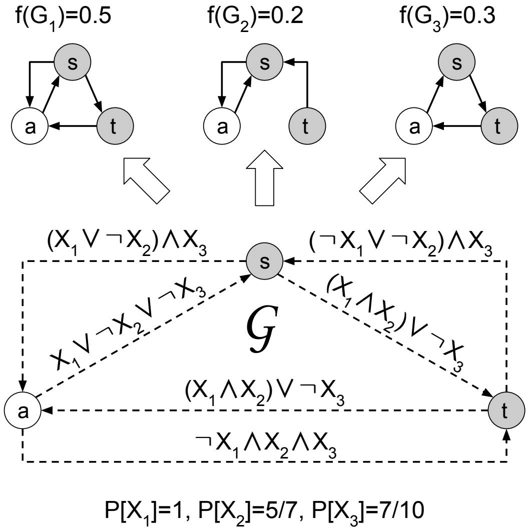

Figure 3: Top: a stochastic mapping f and three deterministic graphs with non-zero probabilities $f(G_1) = 0.5$, $f(G_2) = 0.2$, $f(G_3) = 0.3$. Bottom: an equivalent extended uncertain graph $\mathcal{G}$ of f generated using the construction described in Section 3.3.

Although extended uncertain graphs strictly expand the expressiveness of basic uncertain graphs, there are cases in which the extended uncertain graph model of the studied system can be reduced to an equivalent basic uncertain graph using simple graph transformation tricks (Section 6.3).

3.3 Expressiveness of extended uncertain graphs

In this subsection, we show that our definition of extended uncertain graph is maximally expressive, in the sense that for any stochastic mapping of Γ_V, we can construct an extended uncertain graph whose joint edge existence probability distribution is identically that of Γ_V's stochastic mapping.

THEOREM 3.2. *Every stochastic mapping has an equivalent extended uncertain graph.*

PROOF. Fix the set of vertices V. Let f be a stochastic mapping defined over $\Gamma_V = \{G_1, \ldots, G_N\}$. Define $f^{(i)}$ for $i = 1, \ldots, N$ the following mapping:

$$f^{(i)}(G_j) = \begin{cases} \frac{f(G_j)}{\sum_{k=1}^{i} f(G_k)} & \text{if } 1 \leq j \leq i \\ 0 & \text{if } i < j \leq N \end{cases}$$

Without loss of generality, assume $f(G_1) > 0$ so that every $f^{(i)}$ is well-defined and moreover, it is a valid stochastic mapping since $\sum_{j=1}^{N} f^{(i)}(G_j) = 1$. Especially, $f^{(N)} \equiv f$.

[2] as in stochastic vector

We will show how to iteratively construct an equivalent extended uncertain graph $\mathcal{G}^{(i)}$ of every $f^{(i)}$.

The first step is to show an equivalent extended uncertain graph $\mathcal{G}^{(1)}$ of $f^{(1)}$, a stochastic mapping that maps G_1 to 1 and the rest in Γ_V to 0. Define the extended uncertain graph $\mathcal{G}^{(1)} = (V, E, X^{(1)}, p^{(1)}, q^{(1)})$ as follows:

- V the set of vertices and E the set of all $n(n-1)$ edges, i.e. $G = (V, E)$ is a complete directed graph
- $X^{(1)} = \{X_1\}$
- $p^{(1)} = (p_1)$ where $p_1 = 1$ (i.e. $P[X_1] = p_1 = 1$)
- $q^{(1)}$ works as follows: $\forall E_j \in E$, if $E_j \in E(G_1)$ then $q^{(1)}(E_j) = X_1$, else $q^{(1)}(E_j) = \neg X_1$

It can be easily seen that $\mathcal{G}^{(1)} \equiv f^{(1)}$.

Assume we have constructed $\mathcal{G}^{(i)} = (V, E, X^{(i)}, p^{(i)}, q^{(i)})$ where $X^{(i)} = \{X_1, \ldots, X_i\}$ and $p^{(i)} = (p_1, \ldots, p_i)$ such that $\mathcal{G}^{(i)} \equiv f^{(i)}$ for some $1 \leq i < N$. If $f(G_{i+1}) = 0$ then $f^{(i+1)} \equiv f^{(i)}$. Hence $\mathcal{G}^{(i+1)} = \mathcal{G}^{(i)}$ is the equivalent extended uncertain graph of $f^{(i+1)}$. When $f(G_{i+1}) > 0$, the equivalent extended uncertain graph

$$\mathcal{G}^{(i+1)} = (V, E, X^{(i+1)}, p^{(i+1)}, q^{(i+1)})$$

of $f^{(i+1)}$ can be constructed as follows:

- V the set of vertices and E the set of all $n(n-1)$ edges
- $X^{(i+1)} = \{X_1, \ldots, X_i, X_{i+1}\}$ where X_{i+1} is the newly introduced random variable
- $p^{(i+1)} = (p_1, \ldots, p_i, p_{i+1})$ where $p_{i+1} = \frac{\sum_{j=1}^{i} f(G_j)}{\sum_{j=1}^{i+1} f(G_j)}$
- $q^{(i+1)}$ works as follows: $\forall E_j \in E$, if $E_j \in E(G_{i+1})$ then $q^{(i+1)}(E_j) = q^{(i)}(E_j) \vee \neg X_{i+1}$, else $q^{(i+1)}(E_j) = q^{(i)}(E_j) \wedge X_{i+1}$

The full proof of correctness of this construction is not included in this paper. The construction of $\mathcal{G}^{(i+1)}$ works by scaling down the edge existence probabilities in $\mathcal{G}^{(i)}$ by a factor of p_{i+1} before adding the new graph G_{i+1} with probability $1 - p_{i+1} = 1 - \frac{\sum_{j=1}^{i} f(G_j)}{\sum_{j=1}^{i+1} f(G_j)} = \frac{f(G_{i+1})}{\sum_{j=1}^{i+1} f(G_j)} = f^{(i+1)}(G_{i+1})$. The last step of the construction achieves this by first performing a logic AND operation ($\wedge$) between the Boolean expression associated with every edge of $\mathcal{G}^{(i)}$ and the new random variable X_{i+1}, or formally $q^{(i+1)}(E_j) = q^{(i)}(E_j) \wedge X_{i+1}$. Then, for every edge of $\mathcal{G}^{(i)}$ that appears in G_{i+1}, we additionally perform a logic OR operation ($\vee$) between its associated Boolean expression and $\neg X_{i+1}$. The purpose of doing so is to force $\mathcal{G}^{(i+1)}$ to generate G_{i+1} with probability $1 - p_{i+1}$. Combining these two operations, the Boolean expression associated with every edge of $\mathcal{G}^{(i+1)}$ that appears in G_{i+1} is:

$$\begin{aligned} q^{(i+1)}(E_j) &= (q^{(i)}(E_j) \wedge X_{i+1}) \vee \neg X_{i+1} \\ &= (q^{(i)}(E_j) \vee \neg X_{i+1}) \wedge (X_{i+1} \vee \neg X_{i+1}) \\ &= q^{(i)}(E_j) \vee \neg X_{i+1} \end{aligned}$$

This process allows us to construct an equivalent extended uncertain graph $\mathcal{G}^{(i)}$ of $f^{(i)}$ for $i = 1, \ldots, N$. As the result, $\mathcal{G} = \mathcal{G}^{(N)}$ will be the equivalent extended uncertain graph of f since $f \equiv f^{(N)} \equiv \mathcal{G}^{(N)}$. $\square$

The construction outlined here requires a new random variable for every deterministic graph that has a non-zero probability in f. Therefore, the total number of random variables used by the final extended uncertain graph is $r = |\{G_i | f(G_i) > 0 \text{ for } i = 1, \ldots, N\}|$. For example, the extended uncertain graph in Figure 3 only uses three random variables to model an equivalent stochastic mapping in which only three deterministic graphs have non-zero probabilities G_1, G_2, and G_3. After the first, second, and last iteration of the construction, the Boolean expressions associated with edge (s, a) in $\mathcal{G}^{(1)}$, $\mathcal{G}^{(2)}$, and $\mathcal{G}^{(3)}$ are X_1, $X_1 \vee \neg X_2$, and $X_1 \vee \neg X_2 \vee \neg X_3$, respectively. We notice that both edge (s, t) and (t, a) in $\mathcal{G}$ are associated with the same Boolean expression $(X_1 \wedge X_2) \vee \neg X_3$. This is because (s, t) and (t, a) coexist in all deterministic graphs that have a non-zero probability in f. In general, basic uncertain graphs are not capable of modeling such behavior.

The main importance of this result is that our particular method for extending uncertain graphs, motivated by a particular need to describe correlation among edges in an attack graph, is capable of describing *any* joint distribution of edge existence probabilities. This is an important foundational result in the theory of uncertain graphs.

4. UNCERTAINTY ANALYSIS

Uncertainty analysis plays an important role in understanding how uncertainty in model inputs affects its output. While a selection of vector p gives an expression of uncertainty, that expression itself is likely inexact. This is partly because in many cases, p cannot be directly computed or measured and hence some form of estimation is required. When estimation is used, the resulting estimate usually comes with the form of a mean, which is p, and its upper and lower bounds. Analyses of the uncertain graph therefore must be applied to p as well as its credible neighborhood so that robust conclusions can be made [21]. Among the neighborhood of p, we are interested in two probability assignment vectors under which the model output, i.e. a property of the uncertain graph, acquires its maximum and minimum value. Those extrema tell us how precisely we can arrive at the value of the property in the face of model input uncertainty.

In this paper, we focus on the reachability property of uncertain graphs (first introduced in Section 2.3). Reachability has an intuitive interpretation in the context of security and forms the basis to the answering of numerous security-related questions (Section 5.2). Henceforth, when we talk about uncertainty analysis we will implicitly refer to the reachability property of uncertain graphs. In the remaining part of this section, we first formally define uncertainty analysis as the problem of finding the extrema of the model output (Section 4.1). Then, we show how to quickly identify the extrema using the monotonicity of reachability of the class of monotone uncertain graphs (Section 4.2).

Remark 1. One may argue that although we have supplied the edge existence probability value with its bounds, the bounds can be inexact and so another layer of uncertainty should be considered; this argument can go on forever. Indeed, we never truly know the underlying probability (if one exists) and do not consider such a value in our model. Instead, we take the Bayesian view of probability and treat the edge existence probability as the numerical representation of our belief (and the bounds our confidence in the number), given the information we have collected and subjected to the assumptions we have made. This saves us from the impossible task of defending whether a probability assignment vector is representative of reality or a method to estimate one is the right method – that is to say, unless the method is based on logically flawed technique. Hopefully the justification will become clearer when we attempt to estimate the probability assignment vector in section 6.3.

4.1 Formal definition

Let $\mathcal{G} = (V, E, p)$ denote a basic uncertain graph and $\mathcal{R}_{s,t}(\mathcal{G})$ the probability that s reaches t in $\mathcal{G}$. Define $\epsilon = (\epsilon_1, \ldots, \epsilon_m) \in [0, 1]^m$ the perturbation vector and $\mathcal{H}_{p,\epsilon}$ the hyperrectangle[3] obtained by perturbing each entry p_i in p by an amount of at most ϵ_i, or formally:

$$\mathcal{H}_{p,\epsilon} = \{p' \in [0, 1]^m \mid |p'_i - p_i| \leq \epsilon_i \; \forall i = 1, \ldots, m\}$$

The mean and confidence interval of estimates described earlier can be modeled using the probability assignment and perturbation vector. Uncertainty analysis of uncertain graphs (with respect to the reachability property) aims to find two probability assignment vectors p^{min}, p^{max} in the hyperrectangle $\mathcal{H}_{p,\epsilon}$ such that the reachability of the uncertain graph $\mathcal{G}$ reaches its extrema, i.e:

$$p^{min} = \underset{p' \in \mathcal{H}_{p,\epsilon}}{\operatorname{argmin}} \; \mathcal{R}_{s,t}(V, E, p') \tag{2}$$

$$p^{max} = \underset{p' \in \mathcal{H}_{p,\epsilon}}{\operatorname{argmax}} \; \mathcal{R}_{s,t}(V, E, p') \tag{3}$$

Here we use the notation $\mathcal{R}_{s,t}(V, E, p')$ to denote $\mathcal{R}_{s,t}(\mathcal{G})$ where $\mathcal{G} = (V, E, p')$. Uncertainty analysis of extended uncertain graphs is defined in a similar fashion.

Searching for p^{min} and p^{max} in the hyperrectangle $\mathcal{H}_{p,\epsilon}$ proves to be a nontrivial task. Part of it comes from the difficulty of estimating the reachability of large uncertain graphs. Fortunately, the monotonicity property of reachability allows us to find p^{min} and p^{max} immediately without having to formulate Equations 2 and 3 as optimization problems. The monotonicity of reachability in the context of deterministic graphs means (i) adding one or more edges to a deterministic graph does not change its reachability status (with respect to some source and destination vertex) from 1 to 0 and vice verca, (ii) removing one or more from the graph does not change its reachability status from 0 to 1. The next subsection extends this property to the class of monotone uncertain graphs – uncertain graphs whose edges are associated with monotone Boolean expressions – and the implication regarding how to find p^{min} and p^{max}.

4.2 Uncertainty analysis of monotone uncertain graphs

An extended uncertain graph $\mathcal{G} = (V, E, X, p, q)$ where each uncertain edge is associated with only one random variable, i.e. $q(E_i) \in X$ for every uncertain edge $E_i \in E$, is called a single uncertain graph. We first start with an observation about monotone and single uncertain graphs.

Lemma 4.1. *Every monotone uncertain graph has an equivalent single uncertain graph representation.*

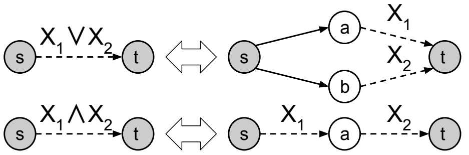

Figure 4: Two monotone uncertain graphs and their equivalent single uncertain graphs.

Details of the proof are omitted to conserve space. An example of monotone uncertain graphs and their equivalent simple uncertain graph representations is given in Figure 4. Lemma 4.1 allows us to prove the following theorem about the monotonicity of reachability with respect to monotone uncertain graphs.

Theorem 4.2. *Let $\mathcal{G} = (V, E, X, p, q)$ and $\mathcal{G}' = (V, E, X, p', q)$ be two monotone uncertain graphs. Furthermore, let $p_i \geq p'_i$ for $i = 1, \ldots, r$. For all $s, t \in V$, the following inequality holds: $\mathcal{R}_{s,t}(\mathcal{G}) \geq \mathcal{R}_{s,t}(\mathcal{G}')$.*

Proof. Without loss of generality, assume $\mathcal{G}$ and $\mathcal{G}'$ are simple uncertain graphs, i.e. $q(E_i) \in X$ for every uncertain edge $E_i \in E$ (otherwise, we can use lemma 4.1 to transform them into simple uncertain graphs).

We first prove a special case of theorem 4.2 in which $\mathcal{G}' = (V, E, X, p', q)$ where $p' = (p'_1, p_2, \ldots, p_r)$. Define $E^1 \subseteq E$ the set of all edges associated with the random variable X_1 and assume $E^1 \neq \varnothing$ (otherwise, redefine $\mathcal{G}$ and $\mathcal{G}'$ without X_1). Furthermore, define two following uncertain graphs:

$$\mathcal{G}^0 = (V, E \backslash E^1, X, (p_2, p_3 \ldots, p_r), q)$$
$$\mathcal{G}^1 = (V, E, X, (1, p_2, \ldots, p_r), q)$$

In plain text, all possible worlds in $\mathcal{G}^1$ contain all edges in E^1 while none in $\mathcal{G}^0$ contains any. The reachability of $\mathcal{G}$ and $\mathcal{G}'$ with respect to any $s, t \in V$ can be computed by conditioning on the random variable X_1 as follows:

$$\mathcal{R}_{s,t}(\mathcal{G}) = p_1 \mathcal{R}_{s,t}(\mathcal{G}^1) + (1 - p_1)\mathcal{R}_{s,t}(\mathcal{G}^0)$$
$$\mathcal{R}_{s,t}(\mathcal{G}') = p'_1 \mathcal{R}_{s,t}(\mathcal{G}^1) + (1 - p'_1)\mathcal{R}_{s,t}(\mathcal{G}^0)$$

Hence:

$$\mathcal{R}_{s,t}(\mathcal{G}) - \mathcal{R}_{s,t}(\mathcal{G}') = (p_1 - p'_1)\left(\mathcal{R}_{s,t}(\mathcal{G}^1) - \mathcal{R}_{s,t}(\mathcal{G}^0)\right)$$

Since $p_1 \geq p'_1$, we only need to prove that $\mathcal{R}_{s,t}(\mathcal{G}^1) \geq \mathcal{R}_{s,t}(\mathcal{G}^0)$. For every possible world $G^1 \in \mathcal{G}^1$, the four following properties hold: (i) G^1 contains all edges in E^1, (ii) G^0, as the result of removing all edges in E^1 from G^1, is a possible world in $\mathcal{G}^0$, (iii) moreover $w_{G^1,\mathcal{G}^1} = w_{G^0,\mathcal{G}^0}$, and lastly (iv) $R_{s,t}(G^1) \geq R_{s,t}(G^0)$ according to the monotonicity of reachability of deterministic graph. Consequently:

$$w_{G^1,\mathcal{G}^1} R_{s,t}(G^1) \geq w_{G^0,\mathcal{G}^0} R_{s,t}(G^0)$$
$$\sum_{G^1 \in \mathcal{G}^1} w_{G^1,\mathcal{G}^1} R_{s,t}(G^1) \geq \sum_{G^0 \in \mathcal{G}^0} w_{G^0,\mathcal{G}^0} R_{s,t}(G^0)$$
$$\mathcal{R}_{s,t}(\mathcal{G}^1) \geq \mathcal{R}_{s,t}(\mathcal{G}^0)$$

Therefore, $\mathcal{R}_{s,t}(\mathcal{G}) \geq \mathcal{R}_{s,t}(\mathcal{G}')$ for a specific case in which $\mathcal{G}' = (V, E, X, p', q)$ where $p' = (p'_1, p_2, \ldots, p_r)$.

Define $\mathcal{G}^{(i)} = (V, E, X, p^{(i)}, q)$ where $p^{(i)} = (p'_1, \ldots, p'_i, p_{i+1}, \ldots, p_r)$ for $i = 1, \ldots, r$. Note that $\mathcal{G}^{(0)} = \mathcal{G}$ and $\mathcal{G}^{(r)} = \mathcal{G}'$. By chaining the inequalities in the following fashion where each

[3] rectangle generalized for higher dimensions

holds as a specific case, $\mathcal{R}_{s,t}(\mathcal{G}) = \mathcal{R}_{s,t}(\mathcal{G}^{(0)}) \geq \mathcal{R}_{s,t}(\mathcal{G}^{(1)}) \geq \ldots \geq \mathcal{R}_{s,t}(\mathcal{G}^{(r-1)}) \geq \mathcal{R}_{s,t}(\mathcal{G}^{(r)}) = \mathcal{R}_{s,t}(\mathcal{G}')$, the theorem is proven. □

The next result immediately follows theorem 4.2:

Corollary 4.2.1. *Let $\mathcal{G} = (V, E, X, p, q)$ be a monotone uncertain graph, $\epsilon \in [0,1]^r$ a perturbation vector such that $p_i - \epsilon_i \geq 0$ and $p_i + \epsilon_i \leq 1$ for $i = 1, \ldots, r$. We have: $p^{min} = p - \epsilon$ and $p^{max} = p + \epsilon$.*

As the main result of this section, corollary 4.2.1 shows us how to perform uncertainty analysis of monotone uncertain graphs. The set of all monotone uncertain graphs contains all basic uncertain graphs but strictly subsumes the set of all extended uncertain graphs, as one might expect. If we take the extended uncertain graph in Figure 2 and change the boolean expression associated with edge (a, t) from X_1 to $\neg X_1$, then we obtain a graph that does not have an equivalent monotone uncertain graph representation. We believe that uncertainty analysis for extended uncertain graphs in the general case can be reduced to the Boolean satisfiability problem, so it is NP-hard with respect to the number of random variables X_i such that both X_i and its negation $\neg X_i$ appear in q. Not surprisingly, this is usually the price we have to pay for extending the expressiveness of a modeling formalism. However, since the NOT logic operator is not required in the modeling examples in Section 5 and 6, uncertainty analysis can be performed efficiently in both cases.

Remark 2. Incorporating uncertainty into the model input is one right step toward producing more trustworthy analyses. However, a large amount of uncertainty in the model input will likely produce a large amount of uncertainty in the model output. Although uncertainty analysis helps us quantify this relation, it does not tell exactly what part of the input's uncertainty attributes the most to the output's. This information is crucial to a modeler who desires to draw a more robust conclusion about the system and who wants to know the best places to spend on reducing uncertainty (by collecting more information, adding more details into the model, etc.) When this is the case, a different but closely related form of analysis called sensitivity analysis should be considered.

5. CASE 1: STUXNET PARTIAL ATTACK GRAPH

In the first modeling example, we show how to use an uncertain graph to model a partial attack graph of the Stuxnet worm (Figure 5), the cyberweapon that sabotaged the Iranian nuclear program in 2009.

5.1 Modeling approach

Converting the Stuxnet partial attack graph (denoted as G_{Stux}) to an uncertain graph (denoted as $\mathcal{G}_{Stux}$) is relatively straightforward. $\mathcal{G}_{Stux}$ uses the same set of vertices of G_{Stux}. Each random variable of $\mathcal{G}_{Stux}$ represents a unique edge label of G_{Stux}. Multiple edges of G_{Stux} that share the same starting and ending vertex translate into a single edge of $\mathcal{G}_{Stux}$. Each edge of $\mathcal{G}_{Stux}$ (e.g. (Contractor, Laptop)) is associated with a disjunction of random variables (e.g. $X_{S7} \vee X_{USB}$) where each variable represents an edge label of G_{Stux} (i.e. X_{S7} denotes the risk associated with "S7 Project Files" and X_{USB} "Infected USB Drive").

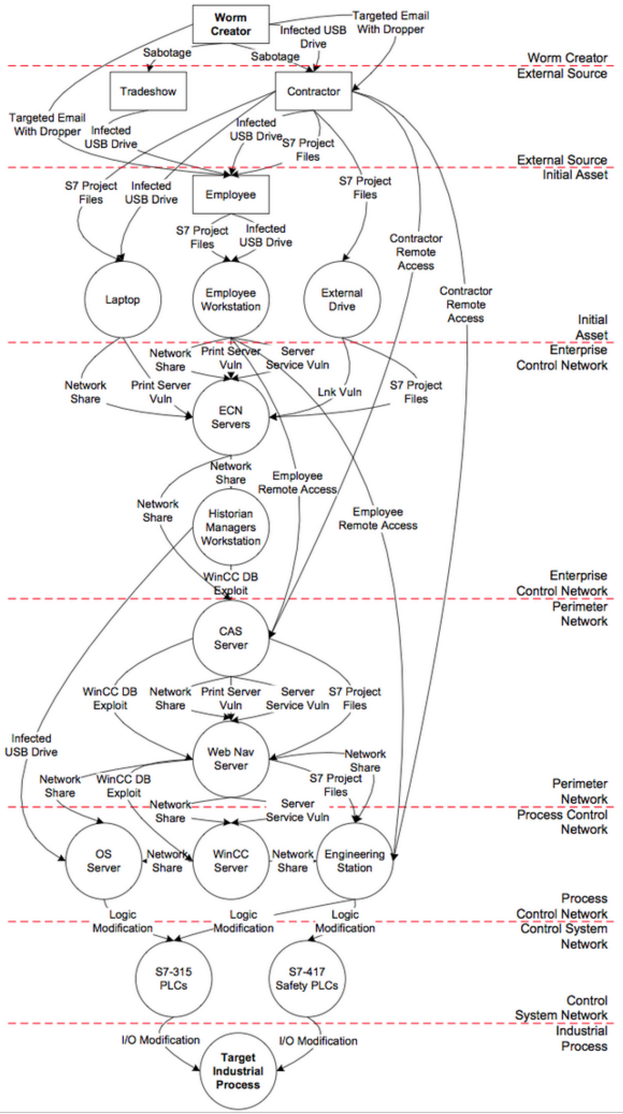

Figure 5: Stuxnet partial attack graph (figure adopted from [5])

The remaining task is to come up with numerical values for the probability assignment vector of $\mathcal{G}_{Stux}$. Those numbers, which may include both the means and their bounds, can be obtained after performing a full security auditing of the system.

5.2 Security analysis

The resulting uncertain graph $\mathcal{G}_{Stux}$ and the analyses in previous sections allow an analyst to answer the following security-related questions:

1. What is the probability $\mathcal{R}_{s,t}(\mathcal{G}_{Stux})$ that there exists a path from the outside of the system to a targeted industrial process?

2. To what extend should I trust the computed probability $\mathcal{R}_{s,t}(\mathcal{G}_{Stux})$, or in other words how precise it is subjected to perturbation of model input?

3. If some form of network hardening is applied to the sys-

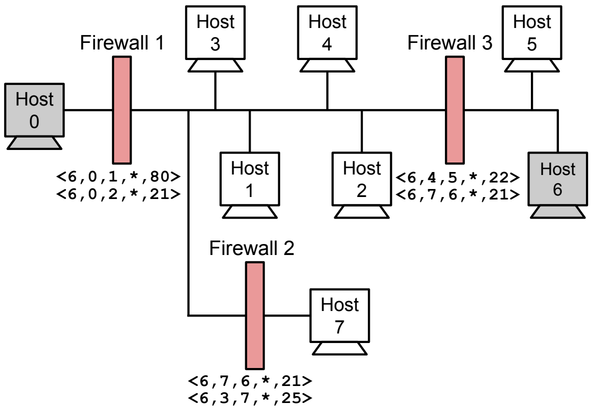

Figure 6: An enterprise network with 3 firewalls and 8 hosts (example adopted from [28], slightly modified for illustration purposes.)

tem and the probability assignment vector re-estimated, will $\mathcal{R}_{s,t}(\mathcal{G}_{Stux})$ be reduced and if so, by how much?

4. Instead of performing network hardening, I want to deploy an intrusion detection system (IDS) to detect ongoing attacks. Assume I choose to monitor a specific set of hosts, what is the chance that I miss an attack?
5. What should I do if the outcome of the analysis is not precise enough to draw a conclusion?

Questions 1 and 3 ask about the reachability of the uncertain graph which is estimated by mean of sampling as shown in Section 2.4. If the size of the graph is relatively small, then reachability can be directly computed using Equation 1. Uncertainty analysis in Section 4 answers Question 2 since $\mathcal{G}_{Stux}$ is monotone. Question 4 can be rephrased into the problem of estimating reachability of uncertain graphs: if I remove the set of vertices that correspond to the set of monitored hosts (together with all edges that connect to and from those vertices), what is the probability that t remains reachable from s? Question 5 is likely to arise in practice and usually indicates that the given amount of information is not sufficient to reason about the security posture of the system (refer to Remark 2 at the end of Section 4).

6. CASE 2: NETWORK SECURITY WITH SERVICE UNCERTAINTY

In the second modeling example, we show how to use uncertain graphs to model a computer network with incomplete information about the network services, or service uncertainty. We first introduce the studied network and some basic networking concepts (Section 6.1). Then we define the threat model (Section 6.2) and propose an approach to model service uncertainty using uncertain graph (Section 6.3). We conclude the section with a note on how the probability assignment vector can be estimated using available information obtained from the common vulnerability databases.

6.1 Network model

Figure 6 shows a simple enterprise network consisting of 3 firewalls and 8 hosts. The firewall rules regulate the communication traffic in the network and define which hosts can directly talk to the other. For example, the 5-tuple rule <6,0,1,*,80> of firewall 1 allows all TCP traffic (protocol type 6) from any port on host 0 to port 80 on host 1. The deny-by-default policy is applied to all firewalls. As a result, firewall 1 blocks all TCP traffic from any port on host 0 to port 25 on host 1.The given enterprise network and the firewall rulesets effectively define a flow graph of logically connected hosts (Figure 7). The flow graph is a directed graph where each vertex represents a host in the enterprise network and each directed edge a flow, i.e. a logical connection. For example, the directed edge from vertex 0 to vertex 1 with the label <80> in Figure 7 represents a 3-tuple flow <6:0-65535:80-80> (i.e. the protocol, the source and destination port). There can be more than one flow from one host to another and in that case, the flow graph is a directed multigraph.

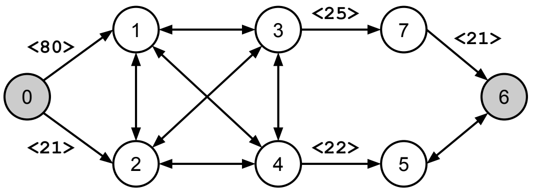

Figure 7: Flow graph representation of the enterprise network in Figure 6. Label <80> on flow from vertex 0 to vertex 1 is short for <6:0-65535:80-80>. Flows without label allow any traffic.

The flow graph is a general description of the types of traffic allowed between hosts in the network. Knowing that flow <6:0-65535:80-80> from host 0 to host 1 exists, we can make an educated guess that host 1 runs some form of an http service. For the purpose of security modeling and analysis, we are also interested in knowing the version and configuration details of the service. Without such information, the existence of a flow does not necessarily imply that an attacker can utilize it as a link in his stepping-stone attack sequence (in fact, the flow might exist while its corresponding service is not running at all). Security modeling and analysis under unquantified input uncertainty will not produce any significant result since any outcome is equally likely. However, if we are allowed to make further assumptions, which are valid ones, then the service uncertainty in flow graphs can be greatly reduced and reasonably estimated using augmented information from the public domain.

6.2 Threat model

We assume the attacker has already gained access to host 0. His ultimate goal is to gain access to host 6, which is a critical asset in the system. To simplify the discussion, we make some further assumptions:

- The attacker only exploits vulnerability of network services running on the receiving host of flows. As a result, if no flow from host 0 to host 1 is allowed or host 1 does not run any vulnerable service, then the attacker cannot launch a direct attack from host 0 to host 1.
- The flow graph remains unchanged throughout the attack period, meaning the attacker does not attempt to attack the firewalls and modify the rulesets to enable new flows.

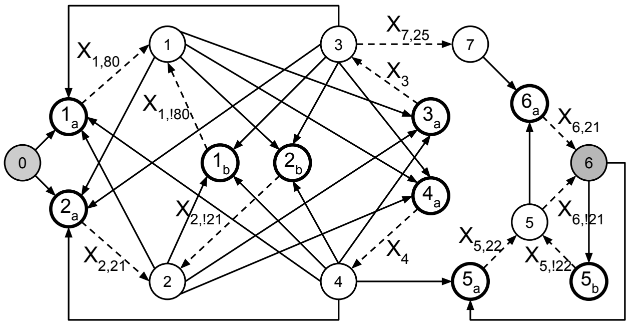

Figure 8: Basic uncertain graph representation of the flow graph in Figure 7.

- Local attacks like privilege escalation are not modeled; we assume the attacker acquires the highest access level after compromising a machine.

6.3 Modeling approach

Define $X_{1,80}$ and $X_{1,!80}$ the random variables that denote if host 1 runs a vulnerable service on port 80 and on some other port that is not 80. The flow graph in Figure 7 indicates the correlation between exploitability of flows in the following sense. If host 1 runs a vulnerable http service on port 80, or $X_{1,80} = true$, then an attacker on either host 0 or 2 can use the existing flows to attack host 1. In contrast, if host 1 does not run any vulnerable http service on port 80, or $X_{1,80} = false$, the attacker cannot attack host 1 from host 0. However, he might be able to do so from host 2, given that host 1 runs a vulnerable service on some other port, i.e. $X_{1,!80} = true$. If we convert the flow graph to an extended uncertain graph with the same set of vertices and edges, then such property can be modeled by associating edge $(0, 1)$ with $X_{1,80}$ and edge $(2, 1)$ with $X_{1,80} \vee X_{1,!80}$. Repeating this process to other edges and vertices, we can build an extended uncertain graph that faithfully models the service uncertainty and the correlation between edge existence of the flow graph in Figure 7. In this modeling example, such an extended uncertain graph can be further reduced to an equivalent basic uncertain graph (Figure 8) using simple graph transformation tricks, which contains transformation artifacts like certain edges, or edges that exist with probability one (solid arrows), and extra vertices (bold circles).

In the last part the section, we briefly discuss how to estimate the probability assignment vector for the constructed basic uncertain graph. The security analyst may assume (or he may learn so from the system administrator) that with no exception, all network services run on standard network ports, i.e. http services on port 80, ftp services on port 21, smtp services on port 25, and so on. The problem of service uncertainty still persists but the uncertainty is now greatly reduced. That is because the analyst knows that, e.g, an attacker can go directly from host 0 to host 1 only if host 1 runs a vulnerable http service. For each http implementation h, the analyst searches in all common vulnerability databases (e.g. the National Vulnerability Database[4]) for some vulnerability of h that allows an attacker to compromise the hosting machine. Denote $v(h) = 1$ if the analyst finds some vulnerability of h and $v(h) = 0$ otherwise. The probability assigned to $X_{1,80}$ is:

$$P[X_{1,80}] = \left(\sum_h w_h v(h)\right) / \sum_h w_h \tag{4}$$

where w_h is the relative weight assigned to implementation h (if no further information is given, all implementations carry the same weight). The analyst might assume all hosts share the default probability p^{def} of running some vulnerable network service (again, if no further information is given). The probability assigned to $X_{1,80}$ and $X_{1,!80}$ and p^{def} are related according to the following equation:

$$p^{def} = 1 - (1 - P[X_{1,80}])(1 - P[X_{1,!80}])$$

Therefore $P[X_{1,!80}]$ can also be estimated. This process applies to the remaining random variables in a similar fashion. Numerical results of the analyses are not reported in this paper and will be a significant topic in follow-up work, in which we study larger and more realistic systems.

7. RELATED WORK

7.1 Uncertain graphs

Uncertain graphs, also known as probabilistic graphs, have been applied to modeling of problems from various domains like interaction between proteins using noisy and error-prone experimental data [2], entity resolution for inexact machine learned models [26], optimal reachability in intermittently connected network with known routing algorithm [8], path queries on road networks with unexpected traffic jams [9], and many others. The power of uncertain graphs comes from its capability of modeling systems with uncertainty, whether due to lack of knowledge about certain part of the systems [8] [9] or to noisy model data [2], [26].

Reasoning with uncertain graphs is challenging since most problems in uncertain graphs are computationally hard. For example, counting the number of possible worlds of an uncertain graph in which vertex s reaches vertex t is #P-complete (ST-CONNECTEDNESS [25]). [19] derived sampling-based approximation algorithms for the k-nearest neighbor problem of uncertain graphs. [10] formulated the distance-constraint reachability (DCR) problem and introduced efficient recursive sampling schemes to estimate DCR of large uncertain graphs. [12] studied reliability search problems of uncertain graphs, i.e. finding all vertices reachable from some query vertices with probability no less than a given threshold, using RQ-tree. Recently, [13] proposed recursive stratified sampling-based estimators to reduce the variance of standard Monte-Carlo approach in estimating uncertain graph properties.

7.2 Attack graph

Traditionally, red teams have constructed attack graphs to represent paths that an attacker may use to compromise the security of a system [22]. Due to the manual nature of the construction of such attack graphs, they are prone to error and often not exhaustive. Automated attack graph generation using model checking was introduced by Ritchey and Ammann [20]. The model check, however, provided just a single attack scenario. Sheyner et al [23] use model checking on heterogeneous networks to provide an exhaustive list

[4] https://nvd.nist.gov/

of attack scenarios. A more scalable solution for larger networks has been proposed in [17]. Another optimization using the monotonicity property has been proposed by Ammann et al [1].

Another related aspect is the process of reachablity analysis. Reachability analysis of a network investigates the conditions under which a target host can be reached by an attacking host. Network scanners [15] and vulnerability discovery tools [6] can be leveraged to derive the configuration of the target network.

Work in [27] and [29] use Bayesian networks to capture the uncertainty of information in attack graphs. However we believe that the acyclic nature of Bayesian networks limits its ability to model the possible cyclic relationships that arise in many practical situations.

8. CONCLUSION

In this paper, we show how to use uncertain graphs for the security modeling and analysis of computer systems with uncertainty. In doing so, we have extended the traditional uncertain graph formalism to model the correlation between edge existence and prove theoretical results about the expressiveness of basic and extended uncertain graphs. We also show how to perform uncertainty analysis of monotone uncertain graphs. Modeling-wise, the developed examples serve as a starting point for taking on larger and more complex systems. In such systems, uncertainty arises from modeling at different layers of abstraction and from the presence of human-in-the-loop. Regarding the later one, uncertain graphs can use existing human-related models to plug holes in the overall attack graph and model the probability that a phishing campaign succeeds or the probability that a power grid operator plugs in the USB stick he received at the conference. Analysis-wise, we are also interested in formulating and solving optimization problems to find the best defense actions, which minimizes the probability of a successful attack, given a limited budget. Those aspects will be explored in subsequent studies.

Acknowledgement

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

9. REFERENCES

[1] Ammann, P., Wijesekera, D., and Kaushik, S. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (2002), ACM, pp. 217–224.

[2] Asthana, S., King, O. D., Gibbons, F. D., and Roth, F. P. Predicting protein complex membership using probabilistic network reliability. *Genome Res.* (2004).

[3] Baier, C., and Katoen, J.-P. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

[4] Blum, A., Burcht, C., and Langford, J. On learning monotone Boolean functions. In *Proceedings 39th Annual Symposium on Foundations of Computer Science* (Nov 1998), pp. 408–415.

[5] Byres, E. Stuxnet Report V: Security Culture Needs Work. http://www.isssource.com/stuxnet-report-v-security-culture-needs-work/, 2011.

[6] Developers, O. The Open Vulnerability Assessment System (OpenVAS), 2012.

[7] Fishman, G. S. A Comparison of Four Monte Carlo Methods for Estimating the Probability of s-t Connectedness. *IEEE Transactions on Reliability 35*, 2 (June 1986), 145–155.

[8] Ghosh, J., Ngo, H. Q., Yoon, S., and Qiao, C. On a Routing Problem Within Probabilistic Graphs and its Application to Intermittently Connected Networks. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications* (May 2007), pp. 1721–1729.

[9] Hua, M., and Pei, J. Probabilistic Path Queries in Road Networks: Traffic Uncertainty Aware Path Selection. In *Proceedings of the 13th International Conference on Extending Database Technology* (New York, NY, USA, 2010), EDBT '10, ACM, pp. 347–358.

[10] Jin, R., Liu, L., Ding, B., and Wang, H. Distance-constraint Reachability Computation in Uncertain Graphs. *Proc. VLDB Endow. 4*, 9 (June 2011), 551–562.

[11] Kaynar, K. A taxonomy for attack graph generation and usage in network security. *Journal of Information Security and Applications* (2016).

[12] Khan, A., Bonchi, F., Gionis, A., and Gullo, F. Fast Reliability Search in Uncertain Graphs. In *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014.* (2014), pp. 535–546.

[13] Li, R.-H., Yu, J. X., Mao, R., and Jin, T. Recursive Stratified Sampling: A New Framework for Query Evaluation on Uncertain Graphs. *IEEE Trans. on Knowl. and Data Eng. 28*, 2 (Feb. 2016), 468–482.

[14] Lippmann, R. P., and Ingols, K. W. An annotated review of past papers on attack graphs. Tech. rep., DTIC Document, 2005.

[15] Lyon, G. F. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning.* Insecure, 2009.

[16] McDermott, J. P. Attack net penetration testing. In *Proceedings of the 2000 workshop on New security paradigms* (2001), ACM, pp. 15–21.

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

[18] Ou, X., and Singhal, A. *Quantitative Security Risk Assessment of Enterprise Networks.* SpringerBriefs in Computer Science. Springer-Verlag New York, 2012.

[19] Potamias, M., Bonchi, F., Gionis, A., and Kollios, G. K-nearest Neighbors in Uncertain Graphs. *Proc. VLDB Endow. 3*, 1-2 (Sept. 2010).

[20] Ritchey, R. W., and Ammann, P. Using Model Checking to Analyze Network Vulnerabilities. In *Proceedings of the 2000 IEEE Symposium on Security*

and Privacy (2000), SP '00, IEEE Computer Society.

[21] Saltelli, A., Ratto, M., Tarantola, S., and Campolongo, F. Sensitivity analysis practices: Strategies for model-based inference. *Reliability Engineering & System Safety 91*, 10-11 (2006), 1109–1125.

[22] Schneier, B. Attack trees. *Dr. Dobb's journal 24*, 12 (1999), 21–29.

[23] Sheyner, O., Haines, J., Jha, S., Lippmann, R., and Wing, J. M. Automated generation and analysis of attack graphs. In *Security and privacy, 2002. Proceedings. 2002 IEEE Symposium on* (2002).

[24] Sheyner, O. M. *Scenario graphs and attack graphs.* PhD thesis, US Air Force Research Laboratory, 2004.

[25] Valiant, L. G. The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing 8*, 3 (1979), 410–421.

[26] Vesdapunt, N., Bellare, K., and Dalvi, N. Crowdsourcing Algorithms for Entity Resolution. *Proc. VLDB Endow. 7*, 12 (Aug. 2014), 1071–1082.

[27] Wang, L., Islam, T., Long, T., Singhal, A., and Jajodia, S. An Attack Graph-Based Probabilistic Security Metric. In *Proceeedings of the 22Nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 283–296.

[28] Wang, L., Jajodia, S., Singhal, A., Cheng, P., and Noel, S. k-Zero Day Safety: A Network Security Metric for Measuring the Risk of Unknown Vulnerabilities. *IEEE Transactions on Dependable and Secure Computing 11*, 1 (Jan 2014), 30–44.

[29] Xie, P., Li, J. H., Ou, X., Liu, P., and Levy, R. Using Bayesian networks for cyber security analysis. In *2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN)* (June 2010), pp. 211–220.