12

Analysis of Stepping-Stone Attacks in Internet of Things Using Dynamic Vulnerability Graphs

Marco Gamarra[1], Sachin Shetty[2], Oscar Gonzalez[1], David M. Nicol[3], Charles A. Kamhoua[4], and Laurent L. Njilla[5]

[1] College of Engineering, Old Dominion University, Norfolk, VA, USA
[2] Virginia Modeling Analysis and Simulation Center, Old Dominion University, Norfolk, VA, USA
[3] Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA
[4] US Army Research Laboratory, Adelphi, MD, USA
[5] Cyber Assurance Branch, US Air Force Research Laboratory, Rome, NY, USA

12.1 Introduction

The ubiquitous adoption of the Internet of Things (IoT) has resulted in an exponential growth in the number of IoT devices. Though IoT has made great impacts to systems in commercial and military domains, there is a growing concern about the security risks introduced by IoT devices. The pervasive networked computing capabilities provided by IoT increases its security risks as attack enablers rather than attack targets. IoT devices can be unwitting participants in a botnet that could lead to secondary attacks. In a 2008 attack on a Turkish oil refinery, security analysts found out that vulnerabilities in the surveillance camera's communication software enabled attackers to gain entry and penetrate deeper into the internal network. Thus, cameras were used as stepping stones to gain access to the network that housed the critical assets.

The growth in the number of IoT devices increases the potential of using these devices as stepping stones to assist attacks by providing additional attack platforms, more opportunities for attack indirection, and eventually increasing the attribution complexity. Security risk assessment in IoT hinges on the ability to quantify the probability of lateral propagation by attackers through a network

Modeling and Design of Secure Internet of Things, First Edition. Edited by Charles A. Kamhoua, Laurent L. Njilla, Alexander Kott and Sachin Shetty.

Published 2020 by John Wiley & Sons, Inc.

of stepping stones. The choice of stepping stones is influenced by the attacker's goal, increased difficulty of attribution, and exertion of minimum effort. The identification of attacker stepping stones is important for security administrators to aid the mitigation process. Vulnerability graphs have been used to model and analyze stepping stones in networked systems [1]. The construction of vulnerability graphs hinges on the availability of network connectivity; identification of software, operating systems, and network protocols; network and system access control rules; and vulnerability information. National Vulnerability Database (NVD) and Common Vulnerability Scoring System (CVSS) scores are typically used to quantify the exploitability of vulnerabilities. In Nicol and Mallapura [1], the authors propose an approach to determine the most vulnerable paths corresponding to stepping-stone attacks by calculating the shortest path in a vulnerability graph with fixed topology. An heuristic analysis using Monte Carlo simulations has been developed in the case of switching topology, assuming that when the attacker is at any host $\boldsymbol{h}$ and the graph topology changes, the path from the host $\boldsymbol{h}$ to the target is invariant. However, there is no guarantee that the edge weights will always remain the same during the attacker's lateral propagation due to defensive mechanisms that can result in modification to the firewall rules, patching of vulnerabilities, and application of security controls. In this chapter, we develop a mathematical model to explore whether it is possible for attackers and/or network administrators to identify stepping stones when the edge weights in the vulnerability graph change.

The shortest path problem in a directed graph was formulated as a linear equation in a min-plus algebra, which can be solved using the Bellman–Ford algorithm [2, 3]. A variant of the Bellman–Ford algorithm for single-source shortest paths in graphs that optimize the algorithm, compared with the previously best variant by Yen [4], has been reported in Bannister and Eppstein [5]. In all of these works, the graph topology is fixed. Average consensus in networks with switching topologies was investigated in Olfati-Saber and Murray [6]. In Nejad et al. [7], max-consensus in graphs with switching topology was investigated using max-plus algebra.

This chapter is complementary to Nicol and Mallapura [1] and attempts to calculate the minimum cost path between source and target nodes when the graph topology changes. Assuming that there is a path from every source to any target in an attack graph at every time instant, the challenge is to calculate the minimum cost path from a source node to a target node when the vulnerability graph changes according to a switching signal that is triggered by the network system's defenses and then analyze scenarios when the problem is not NP-hard (nondeterministic polynomial-time hardness). The main contribution of this chapter is twofold:

1) A mathematical model that describes the stepping-stone attack cost as a dynamical system in a vulnerability graph with switching topology is provided,

where an interplay between min-consensus and min-plus algebra is used for modeling and analysis.

2) The necessary and sufficient conditions are provided for finite-time convergence in the fixed topology case, and a necessary condition for the time interval between two consecutive switching signals that ensure the convergence of the minimum path in finite time is also provided.

The rest of the chapter is organized as follows: Section 12.2 discusses the background, Section 12.3 develops the model of the stepping-stone cost attack in a vulnerability dynamic graph with fixed and switching topology, and Section 12.4 introduces the min-plus algebra and its interplay with the biased min-consensus to analyze the shortest path cost convergence. Finally, Section 12.5 discusses the chapter's results, and Section 12.6 provides the chapter's conclusions and future research.

12.2 Background

12.2.1 Graphs

Given a finite set $V = \{v_1, \ldots, v_n\}$.

Definition 12.1 A **directed Graph** over V is an ordered pair $G = (V, E)$, where E is a subset of the Cartesian product $V \times V$. In this context, V is called Vertex set and E is called Edge set. Every $v_i \in V$ is called **vertex** or **node** and every ordered pair (v_i, v_j) of E is called **directed edge**, where v_i is called the tail and v_j is called the head of the edge.

Definition 12.2 A **directed weighted Graph** over V is an ordered triple $G = (V, E, w)$, where (V, E) is a directed graph and $w : E \rightarrow \mathbb{R}$ is a function that associates a value to each edge.

Definition 12.3 Given a directed weighted graph $G = (V, E, w)$.

1) A vertex v_j is said **adjacent** to v_i if, and only if, $(v_i, v_j) \in E$.
2) For every vertex, v_i is defined as the set of all its **neighbors** as $N_i = \{v_j \in V/(v_i, v_j) \in E\}$.
3) A **path** C of length m in G is a sequence of $m + 1$ vertices $v_{i_1}, v_{i_2}, \cdots, v_{i_{m+1}}$, such that $(v_{i_k}, v_{i_{k+1}}) \in E$ for all $k = 1,\ldots, m$. If $v_{i_1} = v_{i_{m+1}}$, then C is called a **cycle** of length m. A **cycle** of length 1 is called a self-loop.
4) The weighted **adjacency** matrix associated to the weighted graph $G = (V, E, w)$ is defined as the square $n \times n$ matrix A, such that $[A]_{ij} = w_{ij} > 0$ if $(v_i, v_j) \in E$ and $[A]_{ij} = 0$ in all other cases.

Definition 12.4 An undirected Graph over V is a directed graph $G = (V, E)$, such that for every directed edge (v_i, v_j) in V, there is a directed edge (v_j, v_i) in E. These two edges are denoted in a compact way as the unordered pair $\{v_i, v_j\}$ and are called an undirected edge.

Definition 12.5 A directed graph is said to be strongly connected if there is a path connecting every two nodes, and it is called weakly connected if the graph obtained by adding an edge (v_j, v_i) for every existing edge (v_i, v_j) in the original graph is strongly connected. An undirected graph is connected if, and only if, it is strongly connected.

Definition 12.6 A directed multigraph is a directed graph that is permitted to have multiple directed edges between any two vertices, that is, there are multiple edges that have the same head and tail. If the edges are undirected, the graph is called undirected multigraph. If the edges are weighted, the graph is called weighted multigraph.

12.2.2 Vulnerability Graphs and Stepping Stones

Stepping-stone attacks are modeled and analyzed in Nicol and Mallapura [1], which describes a stepping-stone attack as a path through a multigraph where nodes represent hosts, and each edge represents one way in which an attacker, who is resident on the source host, can gain a foothold through an exploit on the destination host. According to the CVSS, a scoring function for every edge in the stepping-stone path has been constructed, called the *exploit complexity score*, which ranges between 0 and 10 defined as $\epsilon = 10 - \varepsilon / A_v$, where ε is the *exploitability score* and A_v is the *access vector*. The *exploit complexity score* quantifies the difficulty of compromise for each node. The smaller the score, the easier it is to exploit the vulnerability. See Nicol and Mallapura [1] for more details in this construction.

A **vulnerability multigraph** has been constructed such that given a set of n node hosts $\{h_1, \cdots, h_n\}$, a weighted edge exists from h_i to h_j if, and only if, there is a vulnerability that allows an attacker on h_i to compromise h_j. This weight ε_{ij} is the *exploit complexity score*, so $0 < \varepsilon_{ij} < 10$. A *stepping-stone path* is a path through a vulnerability multigraph, where the edges denote the vulnerabilities exploited by the attacker to reach the last host in the path from the first.

The **cost** of a stepping-stone path is defined as "the sum of the costs of the edges of the path." The problem of finding a min-cost stepping-stone path between any two hosts comes when the vulnerability graph topology change is NP-hard [1]. As a feasible solution to this problem, Monte Carlo simulations have been used in which stochastic sample paths are generated and the low-cost ones are saved. The challenge posed by varying the edge weights is **solved** under the assumption that the as-yet unseen edge costs are invariant; the standard shortest path algorithm can be used to **estimate** the minimum remaining cost. For more information on this approach, see Nicol and Mallapura [1].

Example 12.1 To illustrate the idea of vulnerability graphs and stepping-stone attacks, we are considering a network that is presented in [8]. Including two virtualization servers and one physical server, the network topology is presented in Figure 12.1. The configuration and function information of the servers is shown in Table 12.1. In this example, it is assumed that an attacker takes the root permissions in the target database PM1 as the final goal to obtain business data. To achieve this goal, an attacker can follow many ways and means. As a first path, the attackers can find the SQL injection vulnerability CVE-2011-2688 on the web server VM21. Through this vulnerability, the attacker gets the user rights of the VM21 and establishes a connection with the database server PM1 on the VM21 with a legitimate identity. Then, through the CVE-2012-2122 and CVE-2010-2693 vulnerabilities on the server PM1, the access mechanism is bypassed and the local authority is carried out. Finally, the attacker gets the root permissions

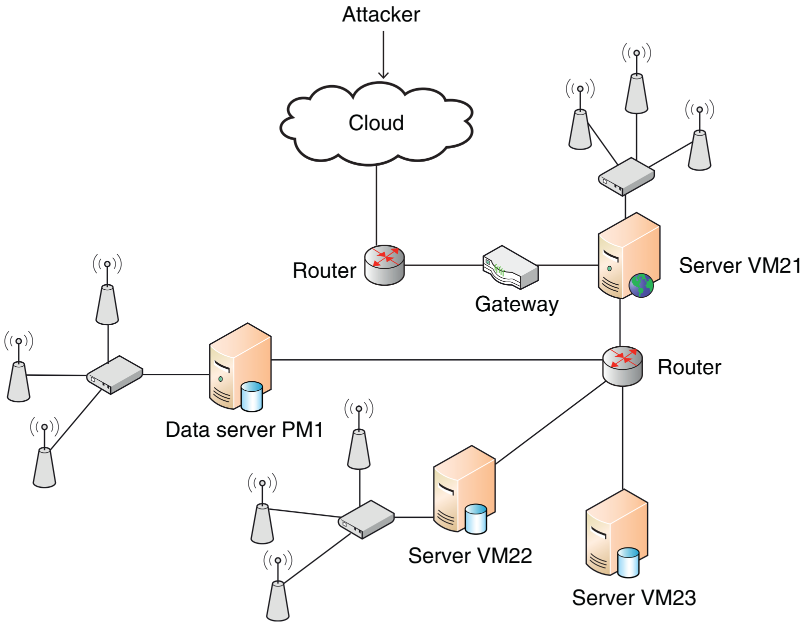


Figure 12.1 Network topology.

Table 12.1 Host configuration, function information, and vulnerability scores information.

Host	OS	Function	Server	Vulnerability CVE-ID	Exploitability subscore ε	A_v	Exploit complexity score $10 - \varepsilon / A_v$	NVD last modified
VM21	Redhat 5.4	Web server	HTTP, SSH	CVE-2011-2688	10.0	1	0	28 August 2017
PM1	Redhat 5.4	Database server	SSH	CVE-2012-2122, CVE-2010-2693	4.9 3.9	1 0.395	5.1 0.1	20 February2014 14 July 2010
VM22	Redhat 5.4	File server	FTP, SSH	CVE-2011-4130, CVE-2012-6067	8.0 10.0	1 1	2 0	12 August 2011 12 May 2012
VM23	Redhat 5.4	Host	SSH	CVE-2008-3234	8.0	1	2	28 September 2017

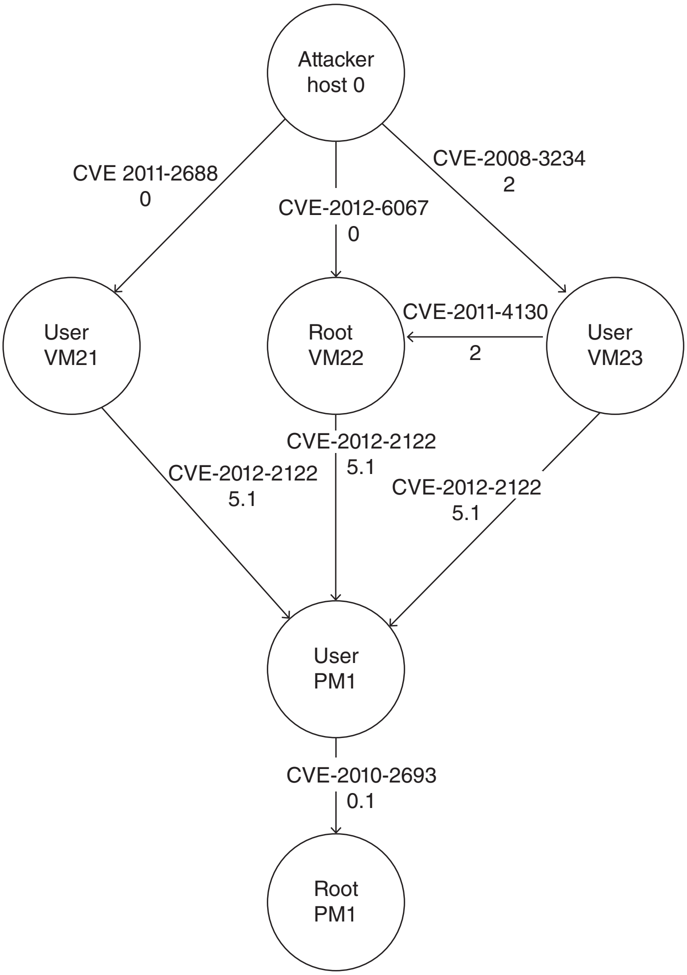


Figure 12.2 A vulnerability graph derived from the network topology presented in Figure 12.1. The weight in the edges is the exploit complexity score associated with its vulnerability. For example, there is a vulnerability in VM23 (CVE-2008-3234) that an attacker in Host 0 can exploit and gain access to VM23, then there is an edge from Host 0 to VM23 with weight 2.

of the database server PM1. On the second path, an attacker can also improve VM22 permissions through a vulnerability, CVE-2012-6067 and CVE-2011-4130 on the file server VM22, then infiltrate to the database server PM1, and finally get the root permissions. In a third path, attackers reach the final goal by directly attacking the database server PM1, see [8] for more details. The associated vulnerability graph is presented in Figure 12.2.

In this context, the most probable path that an attacker will follow is the one with minimum cost. For example, the stepping-stone attack Host 0, VM21, PM1 (User), PM1(Root) that has a cost $0 + 5.1 + 0.1 = 5.2$, or Host 0, VM22, PM1(User), PM1(Root) that has a cost $0 + 5.1 + 0.1 = 7.2$.

Example 12.2 Figure 12.3 presents an abstraction of a physical network as an undirected graph G, and two vulnerability graphs derived from G for two different times, t_1 and t_2, are presented in Figures 12.4 and 12.5, respectively.

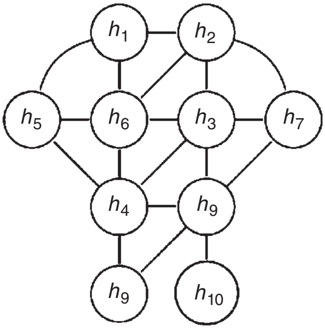


Figure 12.3 Physical network represented with a 10-node undirected graph G. The undirected edge between two nodes means that the communication flow is bidirectional.

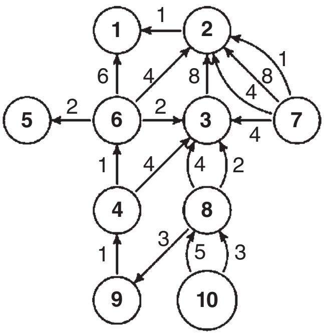


Figure 12.4 A vulnerability multigraph derived from G at any time t_1. For example, there are three directed weighted edges from node 7 to node 2, which means that there are three vulnerabilities with respective exploit complexity scores that allow an attacker on host h_7 to compromise h_2.

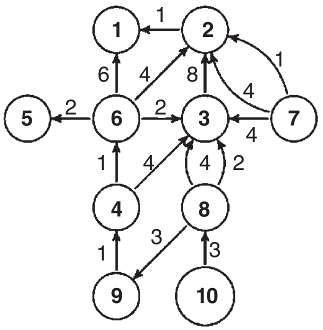


Figure 12.5 Vulnerability multigraph derived from G at any time t_2. In this case, there are two directed weighted edges from node 7 to 2.

12.2.3 Consensus Protocol in a Dynamic Graph

Let $G = (V, E)$ be a directed graph with a vertex set $V = \{v_1, ..., v_n\}$ and edge set E. Let $x \in \mathbb{R}$ denote the value of vertex v_i. Definitions 12.7 and 12.8 are introduced according to Olfati-Saber and Murray [6].

Definition 12.7 The pair (G, x) with $x = [x_1, ..., x_n]^T$ is called an **algebraic graph** with value $x \in \mathbb{R}^n$ and topology (or information flow) G. The value x_i of a vertex v_i represents any physical quantities or any other attribute of the network.

Assume that every vertex-value x_i in the algebraic graph (G, x) is a dynamic agent with dynamics $\frac{dx_i}{dt} = \dot{x}_i(t) = f_i\,(x(t), u_i), \quad i{\in}I = \{1, ..., n\}$.

Definition 12.8 A **dynamic graph** is a dynamical system with a state (G, x) in which the value $x = [x_1, ..., x_n]^T$ evolves according to the network dynamics

$$\dot{x}_i = f_i(x, u_i), \quad \forall i = 1, 2, ..., n, \tag{12.1}$$

where

$$u_i = g_i(x) \tag{12.2}$$

is called a **distributed protocol** with topology G. In matrix form

$$\dot{x} = f(x, u), \tag{12.3}$$

where

$$\dot{x} = [\dot{x}_1, ..., \dot{x}_n]^T, \quad f(x, u) = [f_1(x, u_1), ..., f_n(x, u_n)]^T,$$

and

$$u = [u_1, ..., u_n]^T = [g_1(x), ..., g_n(x)]^T.$$

In the literature, an algebraic graph is usually called an algebraic network and a dynamical graph is called a dynamic network. In the rest of the chapter, we illustrate a simple case of the dynamical network (12.1) when $f_i(x, u_i) = u_i$ is considered, that is,

$$\dot{x}_i(t) = u_i(t), \quad \forall i = 1, 2, ..., n. \tag{12.4}$$

The next two examples are an illustration of a distributed calculation with the linear and nonlinear functions:

$\chi_1(x) = \text{Ave}\{x_1(t), ..., x_n(t)\} = \text{Ave}\{x(t)\}$, $\chi_2(x) = \min\{x_1(t), ..., x_n(t)\} = \min\{x(t)\}$, respectively, using a dynamic graph (G, x), where G is assumed strongly connected and $N_i = \{x_j/(x_i, x_j)$ is an edge in $G\}$ is the set of all the neighbors of the vertex x_i.

Example 12.3 (Average consensus). The following distributed linear protocol,

$$u_i = \sum_{j \in N_i} (x_j - x_i), \tag{12.5}$$

asymptotically solves the average consensus problem and is called the average consensus protocol, that is, replacing (12.5) in (12.4) yields

$$\dot{x}(t) = \sum_{j \in N_i} (x_j - x_i), \tag{12.6}$$

as is proved in [6],

$$\begin{aligned} \lim_{t \to \infty} x_i(t) &= \frac{1}{n} \sum_{i=1}^{n} x_i(0), \quad i = 1, ..., n \\ &= \text{Ave}\{x(0)\}. \end{aligned}$$

Example 12.4 (Min-consensus). The following distributed nonlinear protocol,

$$u_i = \min_{j \in N_i} (x_j - x_i), \tag{12.7}$$

asymptotically solves the min-consensus problem. Replacing (12.7) in (12.4) yields

$$\dot{x}_i = \min_{j \in N_i} (x_j - x_i) = -x_i + \min_{j \in N_i} (x_j) \tag{12.8}$$

as is proved in [6],

$$\begin{aligned} \lim_{t \to \infty} x_i(t) &= \min\{x_1(0), ..., x_n(0)\}, \quad i = 1, ..., n \\ &= \min\{x(0)\}. \end{aligned}$$

12.2.4 Biased Min-Consensus

The min-consensus protocol is perturbed [9], yielding the following distributed nonlinear protocol,

$$u_i = -x_i + \min_{j \in N_i} (x_j + w_{ij}), \tag{12.9}$$

where w_{ij} is the weight of edge (x_i, x_j), which is called *biased min-consensus protocol.* Closing the loop, we have

$$\dot{x}_i = -x_i + \min_{j \in N_i} (x_j + w_{ij}), \tag{12.10}$$

that asymptotically converges to the equilibrium point x_i^* [9], that is,

$$\lim_{t \to \infty} x_i(t) = x_i^*,$$

which satisfies the following equation:

$$x_i^* = \min_{j\in N_i}\left(x_j^* + w_{ij}\right), \quad i = 1, \ldots, n.$$

12.2.5 Leader–Follower Strategy

In many applications of multi-agent systems, a leader–follower strategy is applied. In this approach, a subset of agents is called *leader set* N_l, and the remainder agents are called *follower set* N_f. In this context, the average consensus, min-consensus, and biased min-consensus are

$$\begin{cases} \dot{x}_i = v_i & \text{if } i \in N_l \\ \dot{x}_i = \sum\limits_{j\in N_i}\left(x_j - x_i\right) & \text{if } i \in N_f \end{cases}, \tag{12.11}$$

$$\begin{cases} \dot{x}_i = v_i & \text{if } i \in N_l \\ \dot{x}_i = -x_i + \min\limits_{j\in N_i}\left(x_j\right) & \text{if } i \in N_f \end{cases}, \tag{12.12}$$

$$\begin{cases} \dot{x}_i = v_i & \text{if } i \in N_l \\ \dot{x}_i = -x_i + \min\limits_{j\in N_i}\left(x_j + w_{ij}\right) & \text{if } i \in N_f \end{cases}, \tag{12.13}$$

respectively, where v_i is an exogenous input. If $v_i = 0$, the systems (Eqs. 12.11–12.13) are called *static leaders system*. The following theorem has been stated and proven in Zhang and Li [9].

Theorem 12.1 Let G be an undirected connected graph, and suppose that the dynamical network (G, x) evolves according to the protocol (Eq. 12.13) with static leaders. Then the system asymptotically converges to the equilibrium point x^* of (Eq. 12.13), which satisfies the following equation [9]:

$$\begin{cases} x_i^* = x_i(0) & \text{if } i \in N_l \\ x_i^* = \min\limits_{j\in N_i}\left(x_j^* + w_{ij}\right) & \text{if } i \in N_f \end{cases}. \tag{12.14}$$

12.2.6 Biased Min-Consensus and Shortest Path

The relationship between the biased min-consensus protocol in a dynamical network (G, x) and the shortest path in G has been developed in Zhang and Li [9] as follows: The "leader" agents are static, that is, $\dot{x}_i\ (t) = 0$, $\forall \dot{x}_i \in N_l$, and are called *destination nodes*. The "follower" agents are called *source nodes*.

- If there is an edge between x_i and x_j, the weight w_{ij} of this edge is the *length* between these agents.

- The system evolves according to the protocol (Eq. 12.13) with static leaders. Note that according to the optimality principle of Bellman's dynamic programming [10], the solution of the considered shortest path problem satisfies the following nonlinear equations:

$$\begin{cases} x_i^* = 0 & \text{if } i \in N_l \\ x_i^* = \min\limits_{j \in N_i} \left(x_j^* + w_{ij} \right) & \text{if } i \in N_f \end{cases}, \tag{12.15}$$

which are the equilibrium points of (Eq. 12.13) with static leaders and $x_i(0) = 0, \ \forall x_i \in N_l$.

The following theorem has been stated and proven in Zhang and Li [9].

Theorem 12.2 If $x_i(0) = 0, \ \forall x_i \in N_l$, then the equilibrium of the system (Eq. 12.13) with statics leaders is given by (Eq. 12.15), which forms a solution to the corresponding shortest path problem [9].

Remark 12.1

1) When the states have reached the equilibrium point (Eq. 12.15), the shortest path can be found by recursively finding the parent nodes [9].
2) According to Theorems 12.1 and 12.2, all the state values of the system globally converge to the lengths of the corresponding shortest path independently of the initial state values [9].

12.3 Stepping-Stone Dynamics

12.3.1 Fixed Topology Case

Given a vulnerability graph G with n hosts $\{h_1, \ldots, h_n\}$, a dynamic graph (G, x) is assigned such that for every h_i, there is a state $x_i \in \mathbb{R}$ and the weight of the edge (x_i, x_j) is its exploit complexity score ϵ_{ij}. A leader–follower strategy is used where the state x_i evolves according to biased min-consensus dynamics (Eq. 12.13) with static leaders; in this model, the leaders are called ***valuable targets*** and the followers are called ***source nodes***. If x_l is a valuable target, then the state of the source x_i is defined as the stepping-stone cost from x_i to x_l, that is, the length of the path from x_i to x_l. Mathematically, the stepping-stone dynamics can be written as

$$\begin{cases} \dot{x}_i(t) = 0, \quad x_i(0) = 0 & \text{if } i \in N_l \\ \dot{x}_i(t) = -x_i(t) + \min\limits_{j \in N_i} \left(x_j(t) + \epsilon_{ij} \right) & \text{if } i \in N_f \end{cases}, \tag{12.16}$$

then, according to Theorem 12.2, in the equilibrium $x_i^* \in N_f$ is the minimum stepping-stone cost from x_i to any valuable target x_l, that is, the most probable stepping-stone attack from host h_i to any valuable target h_l.

As claimed in Nicol and Mallapura [1], according to experience, as an attacker penetrates more deeply into a system, the exploit complexity score should change and, as a consequence, the graph topology will change as well. In the following section, we develop a more realistic model in which the *exploit complexity score* changes as the attack penetrates more deeply in the system.

12.3.2 Switching Topology Case

Let $\{G_1, ..., G_m\}$ be a finite collection of vulnerability multigraphs with the same n hosts $\{h_1, ..., h_m\}$, and $s : \mathbb{R} \to \{1, ..., m\}$ a switching signal. For every $s(t) = l$, a vulnerability multigraph $G_l \in \{G_1, ..., G_m\}$ and its associated dynamic graph (G_l, x) are well defined, and then the stepping-stone dynamics with switching topology is equivalent to the hybrid system

$$\begin{cases} \dot{x}_i(t) = 0, \quad x_i(0) = 0 & \text{if } i \in N_l \\ s(t) = l & \\ \dot{x}_i(t) = -x_i(t) + \min\limits_{j \in N_i} \left(x_j(t) + \epsilon_{ij}(i) \right) & \text{if } i \in N_f \end{cases}, \tag{12.17}$$

where $\epsilon_{ij}(l)$ is the exploit vulnerability score of the edge $\{x_i, x_j\}$ in the dynamic graph (G_l, x). Hence, if the network has a vulnerability graph G_p and at any time $t > 0$:

1) An attack from h_i is detected in h_j, then the network's vulnerability graph switches to $G_q = s(t)$ such that $\epsilon_{ij}(q) > \epsilon_{lm}(p)$ for all $\epsilon_{lm}(p) = \epsilon_{ij}(p)$.
2) An attack from h_i compromises h_j, then the network's vulnerability graph switches to $G_q = s(t)$ such that $\epsilon_{ij}(q) < \epsilon_{lm}(p)$ for all $\epsilon_{lm}(p) = \epsilon_{ij}(p)$.

The progress of the stepping-stone dynamics is monitored at δ time intervals; hence, the stepping-stone dynamics at a discrete time for the fixed topology case is

$$\begin{cases} x_i[k+1] = x_i[k], \quad x_i[0] = 0 & \text{if } i \in N_l \\ x_i[k+1] = \min\limits_{j \in N_i} \left(x_j[k] + \epsilon_{ij}(l) \right) & \text{if } i \in N_f \end{cases}, \tag{12.18}$$

where

$$x_i[k] \overset{\text{def}}{=} x_i(\delta k), \delta > 0 \text{ and } k \in \mathbb{Z}_0{}^{+}, \tag{12.19}$$

and for the switching-topology case, is

$$\begin{cases} x_i[k+1] = x_i[k], \quad x_i[0] = 0 & \text{if } i \in N_l \\ l = s[k] & \\ x_i[k+1] = \min\limits_{j \in N_i} \left(x_j[k] + \epsilon_{ij}(l) \right) & \text{if } i \in N_f \end{cases}. \tag{12.20}$$

12.4 Min-Plus Algebra

Min-plus algebra consists of two binary operations, $\oplus$ and $\otimes$, on the set $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$, which is defined as follows:

$$a \oplus b = \min\{a, b\}, \tag{12.21}$$

$$a \otimes b = a + b. \tag{12.22}$$

The neutral element with respect to the min-plus addition $\oplus$ is $+\infty$, denoted as θ, and with respect to the min-plus multiplication $\otimes$ is 0, denoted as e. Both operations are associative and commutative, and the multiplication is distributive over the addition. Both operations are extended to matrices as follows:

Given $A, B \in \mathbb{R}_{\min}^{m \times n}$,

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij}, \quad i = 1, \ldots, m \quad j = 1, \ldots, n.$$

Given $A \in \mathbb{R}_{\min}^{m \times q}, B \in \mathbb{R}_{\min}^{q \times n}$,

$$\begin{aligned}[A \otimes B]_{ij} &= \bigoplus_{k=1}^{q} \left(a_{ik} \otimes b_{kj}\right), \quad i = 1, \ldots, m \quad j = 1, \ldots, n \\ &= \min_{k=1}^{q} \left\{a_{ik} + b_{kj}\right\}.\end{aligned}$$

The identity matrix of size n is a square matrix denoted by I_n and given by

$$[I_n]_{ij} = \begin{cases} e & \text{for } i = j \\ \theta & \text{for } i \neq j \end{cases}.$$

If $A \in \mathbb{R}_{\min}^{n \times n}$ for any integer $k \geq 1$, $A^k = \underbrace{A \otimes A \otimes \cdots \otimes A}_{k-1 \quad \text{multiplications}}$ and $A^0 = I_n$. For more properties and application of min-plus algebra, see Cohen et al. [3] and the references therein. If G is a vulnerability graph with n nodes {1, ..., n}, then a modified weighted adjacency matrix, $A \in \mathbb{R}_{\min}^{n \times n}$, is associated to G and defined as

$$A = \begin{cases} [A]_{ii} = e, & \text{if } i \text{ is a target node} \\ [A]_{ij} = \epsilon_{ij}, & \text{if } (i,j) \text{ is an edge} \\ [A]_{ij} = \theta, & \text{in other cases} \end{cases}. \tag{12.23}$$

Notice that in this matrix, $[A]_{ij} \neq 0$ means that there is one path of length **1** from i to j in G. In $A^2 = A \otimes A$, if $[A \otimes A]_{ij} \neq \theta$, then there is a path of length **2** in G with a minimum cost from node i to node j, that is, there is a node l in G such that (i, l) and (l, j) are edges in G, such that $\min_{k=1}^{n}\left\{a_{ik} \otimes a_{kj}\right\} = a_{il} \otimes a_{lj} = a_{il} + a_{lj}$, but also $a_{il} \otimes$

$a_{lj} \neq \theta$ implies that $a_{il} \neq \theta$ and $a_{lj} \neq \theta$ simultaneously, and $a_{il} \otimes a_{lj} = \theta$ implies that $a_{il} = \theta$ or $a_{lj} = \theta$. Using the min-plus formalism, the stepping-stone dynamics with fixed topology (Eq. 12.18) can be rewritten as

$$\begin{cases} x_i[k+1] = x_i[k], \quad x_i[0] = 0 & \text{if } i \in N_l \\ x_i[k+1] = \underset{j \in N_i}{\oplus} \left(x_j[k] \otimes \epsilon_{ij}\right) & \text{if } i \in N_f \end{cases},$$

or in matrix form as

$$x[k+1] = A \otimes x[k] = A^{k+1} \otimes x[0], \tag{12.24}$$

and the stepping-stone dynamics with switched topology (Eq. 12.20) can be written as

$$\begin{cases} x_i[k+1] = x_i[k], \quad x_i[0] = 0 & \text{if } i \in N_l \\ l = s[k] & \\ x_i[k+1] = \underset{j \in N_i}{\oplus} \left(x_j[k] \otimes \epsilon_{ij}(l)\right) & \text{if } i \in N_f \end{cases},$$

or in matrix form as

$$x[k+1] = A_l \otimes x[k], \quad l = s[k], \tag{12.25}$$

where A_l is the matrix associate with the vulnerability graph, $G_l \in \{G_1, ..., G_m\}$.

Theorem 12.3 A necessary and sufficient condition for which the stepping-stone dynamics (Eq. 12.18) converge to equilibrium (Eq. 12.14) is that $A^{k+1} = A^k$ for any integer $k \geq 1$.

Proof: **Sufficiency**

$$x[k+1] = A \otimes x[k] = A^{k+1} \otimes x[0] = A^k \otimes x[0] = x[k]$$

hence, $x[k+1] - x[k] = 0$, which implies (Eq. 12.14).

Necessity: If the system is in equilibrium for any integer $k \geq 1$, then this implies that $x[k+1] - x[k] = 0$,

or equivalently

$$\begin{cases} x_i[k+1] = x_i[k] = x_i[0] = 0 & \text{if } i \in N_l \\ x_i[k+1] = x_i[k] = \min\limits_{j \in N_i} \left(x_j[k] + \epsilon_{ij}\right) & \text{if } i \in N_f \end{cases},$$

hence $[A^{k+1}]_{ii} = [A^k]_{ii} = 0$ if $i \in N_l$, because in the vulnerability graph there are no self-loops in the source nodes. If $i \in N_f$, there is a path of length k from node i to any target node j with minimum cost $x_i[k] = [A^k]_{ij} \otimes x_j[0] = [A^k]_{ij}$, and there is a path of length $k+1$ from node i to the same target node j with minimum cost

$x_i[k+1] = [A^{k+1}]_{ij} \otimes x_j[0] = [A^{k+1}]_{ij} = x_i[k] = [A^k]_{ij}$, because $x_j[0] = 0$ is the cost for a target node, and therefore, $A^{k+1} = A^k$. □

Corollary 12.1 If $A \in \mathbb{R}_{\min}^{n \times n}$ is the matrix associated with the vulnerability graph G and there is an integer $k \geq 1$ such that $A^{k+1} = A^k$, then $k \leq n-1$.

Proof: $x_i[k+1] = [A^{k+1}]_{ij} = x_i[k] = [A^k]_{ij}$ implies that there is a path of length k from node i to node j with minimum cost, and equivalently there is a simple path from i to j with length k. Because the maximum length of a simple path in a graph with n nodes is $n-1$, then $k \leq n-1$. □

Remark 12.2 Corollary 12.1 implies that the stepping-stone dynamics (Eq. 12.18) converge to equilibrium in finite time $\tau = k\delta$ with $k \leq n-1$ instant communications.

Theorem 12.4 The stepping-stone dynamics with switching topology (Eq. 12.20) converges to equilibrium if the time interval between two consecutive switching signals is large enough, as $k = n-1$ instant communications.

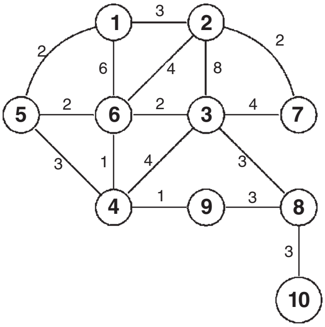
Proof: By Corollary 12.1, $n-1$ instant communications are the finite time period that guarantees the equilibrium for any fixed graph topology, and if the time interval between two switching signals is larger than $n-1$, then the system reaches equilibrium and the shortest path is calculated, which proves the theorem. □

Remark 12.3 In Theorem 12.4, notice that a convergence to equilibrium is possible for any time interval between switching signals in less than $n-1$ communication instants, as shown in Example 12.6. We will discuss this point in Section 12.5.

The model was developed for a directed graph with the condition that there is a path from every source node to any target. In the following examples, the vulnerability graph is considered to be strongly connected, assuming that an attacker has the advantage of compromising any node in any layer of the vulnerability graph.

Example 12.5 (Fixed topology). Consider the 10-node vulnerability graph G presented in Figure 12.6. The exploit complexity scores are encoded in its weighted adjacency matrix presented in Eq. (12.26), and the modified adjacency matrix, as defined in Eq. (12.23), can be derived immediately, which is presented in Eq. (12.27).

Figure 12.6 A 10-node vulnerability graph $G = G_p$ with node 1 as a target node.



$$
\epsilon = \begin{bmatrix}
0 & 3 & 0 & 0 & 2 & 6 & 0 & 0 & 0 & 0 \\
3 & 0 & 8 & 0 & 0 & 4 & 2 & 0 & 0 & 0 \\
0 & 8 & 0 & 4 & 0 & 2 & 4 & 3 & 0 & 0 \\
0 & 0 & 4 & 0 & 3 & 1 & 0 & 0 & 1 & 0 \\
2 & 0 & 0 & 3 & 0 & 2 & 0 & 0 & 0 & 0 \\
6 & 4 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0
\end{bmatrix} \tag{12.26}
$$

$$
A = \begin{bmatrix}
e & 3 & \theta & \theta & 2 & 6 & \theta & \theta & \theta & \theta \\
3 & \theta & 8 & \theta & \theta & 4 & 2 & \theta & \theta & \theta \\
\theta & 8 & \theta & 4 & \theta & 2 & 4 & 3 & \theta & \theta \\
\theta & \theta & 4 & \theta & 3 & 1 & \theta & \theta & 1 & \theta \\
2 & \theta & \theta & 3 & \theta & 2 & \theta & \theta & \theta & \theta \\
6 & 4 & 2 & 1 & 2 & \theta & \theta & \theta & \theta & \theta \\
\theta & 2 & 4 & \theta & \theta & \theta & \theta & \theta & \theta & \theta \\
\theta & \theta & 3 & \theta & \theta & \theta & \theta & \theta & 3 & 3 \\
\theta & \theta & \theta & 1 & \theta & \theta & \theta & 3 & \theta & \theta \\
\theta & \theta & \theta & \theta & \theta & \theta & \theta & 3 & \theta & \theta
\end{bmatrix} \tag{12.27}
$$

Table 12.2 Stepping-stone cost evolution from every source node to the target, where every numerical column shows the stepping-stone cost from the respective source; for example, the row labeled with the agent x_8 shows that in the third iteration, its stepping-stone cost is 6.

		Stepping-stone cost/iteration						
Source	$x2$	3	3	3	3	3	3	3
	$x3$	3	4	5	6	6	6	6
	$x4$	2	3	4	5	5	5	5
	$x5$	2	2	2	2	2	2	2
	$x6$	2	3	4	4	4	4	4
	$x7$	3	5	5	5	5	5	5
	$x8$	4	5	6	7	8	9	9
	$x9$	2	3	4	5	6	6	6
	$x10$	4	7	8	9	10	11	12

If x_1 is a valuable target, then the stepping-stone dynamics are

$$\begin{cases} x_1[k+1] = x_1[k] = x_1[0] = 0 \\ x_i[k+1] = x_i[k] = \min\limits_{j \in N_i} \left(x_j[k] + \epsilon_{ij} \right) \qquad \text{if } i \in \{2, \dots, 10\} \end{cases} \cdot \tag{12.28}$$

The simulation is presented in Table 12.2, where the initial states are $x_i(0) = 1, \quad \forall i = 2, \dots, 10$, and the most vulnerable stopping-stone path for all the sources (shortest path) has been reached after seven iterations. For example, the most vulnerable stepping-stone path from node 10 to node target 1 have cost 12, and there are two: $10 \rightarrow 8 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 1$ and $10 \rightarrow 8 \rightarrow 9 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 1$. The most vulnerable stepping-stone path from node 6 to node target 1 has cost 4 and is unique: $6 \rightarrow 5 \rightarrow 1$.

Example 12.6 (Switching topology). Assume that at any time $t_1 > 0$, a 10-node vulnerability graph has topology $G_p = G$ (presented in Figure 12.6). The exploit complexity scores are encoded in its weighted incidence matrix $\epsilon(p) = \epsilon$ (presented in Eq. (12.26)). Assume that at any time $t_2 > t_1$, and an attack from host h_6 is detected in host h_5, then the network defense is activated with a switching signal $s(t_2) = q$ yielding a new network topology G_q (presented in Figure 12.7). The new exploit complexity scores are encoded in its weighted incidence matrix $\epsilon(q)$ as presented in Eq. (12.29), where $\epsilon_{ij}(q) = 9$ if $\epsilon_{ij}(p) = 2$ and $\epsilon_{ij}(q) = \epsilon_{ij}(p)$ if $\epsilon_{ij}(p) \neq 2$. The new modify incidence matrix A_q, as defined in Eq. (12.23), is presented in Eq. (12.30).

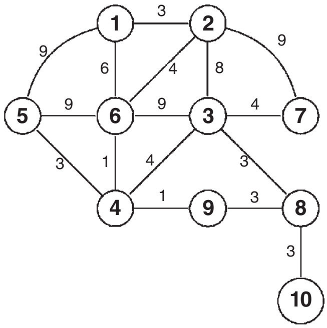


Figure 12.7 A 10-node vulnerability graph G_q derived from graph G_p where the edges with weight 2 have been switched with edges with weight 9.

$$\epsilon(q) = \begin{bmatrix} 0 & 3 & 0 & 0 & 9 & 6 & 0 & 0 & 0 & 0 \\ 3 & 0 & 8 & 0 & 0 & 4 & 9 & 0 & 0 & 0 \\ 0 & 8 & 0 & 4 & 0 & 9 & 4 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 & 3 & 1 & 0 & 0 & 1 & 0 \\ 9 & 0 & 0 & 3 & 0 & 9 & 0 & 0 & 0 & 0 \\ 6 & 4 & 9 & 1 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \end{bmatrix} \tag{12.29}$$

$$A_q = \begin{bmatrix} e & 3 & \theta & \theta & 9 & 6 & \theta & \theta & \theta & \theta \\ 3 & \theta & 8 & \theta & \theta & 4 & 9 & \theta & \theta & \theta \\ \theta & 8 & \theta & 4 & \theta & 9 & 4 & 3 & \theta & \theta \\ \theta & \theta & 4 & \theta & 3 & 1 & \theta & \theta & 1 & \theta \\ 9 & \theta & \theta & 3 & \theta & 9 & \theta & \theta & \theta & \theta \\ 6 & 4 & 9 & 1 & 9 & \theta & \theta & \theta & \theta & \theta \\ \theta & 9 & 4 & \theta & \theta & \theta & \theta & \theta & \theta & \theta \\ \theta & \theta & 3 & \theta & \theta & \theta & \theta & \theta & 3 & 3 \\ \theta & \theta & \theta & 1 & \theta & \theta & \theta & 3 & \theta & \theta \\ \theta & \theta & \theta & \theta & \theta & \theta & \theta & 3 & \theta & \theta \end{bmatrix} \tag{12.30}$$

In the simulation presented in Table 12.3 where the initial states are $x(0) = [0\ 4\ 7\ 4\ 9\ 8\ 7\ 8\ 3\ 9]^T$, the graph topology has changed from G_p to G_q in the fifth iteration, that is, for $k = 5$, $t_2 = \delta 5$, then $s[5] = q$. The most vulnerable stepping-stone path for all the sources (shortest path) has been reached after nine iterations; for example, the most vulnerable stepping-stone path from node 10 to

Table 12.3 Stepping-stone cost evolution from every source; notice that after the double vertical line, when $k = 5$, the graph topology has changed, and the stepping-stone cost has been recalculated with the new information.

		Stepping-stone cost/iteration								
Source	$x2$	3	3	3	3	3	3	3	3	3
	$x3$	44	8	6	6	9	9	11	11	11
	$x4$	31	5	5	5	5	7	7	7	7
	$x5$	2	2	2	2	8	8	9	9	9
	$x6$	6	4	4	4	6	6	6	6	6
	$x7$	42	5	5	5	10	12	12	12	12
	$x8$	33	44	11	9	9	9	9	11	11
	$x9$	41	32	6	6	6	6	8	8	8
	$x10$	83	36	47	14	12	12	12	12	14

node target 1 has cost 14 and is unique: $10 \rightarrow 8 \rightarrow 9 \rightarrow 4 \rightarrow 6 \rightarrow 1$. The most vulnerable stepping-stone path from node 6 to node target 1 has cost 6 and is unique: $6 \rightarrow 1$.

12.5 Discussion

If r is the minimum cost from node i to node j, and s is the minimum cost from node i to node l with $r < s$, then equations (Eq. (12.18)) that are used in the calculation of the shortest path provide r and the paths themselves are calculated according to Remark 12.1, which means that the most probable stepping-stone attack from the source node i is toward the target node j.

In the model, a path between every source and any target has been assumed, but also, if there is a source without a path to any target, then the equations (Eq. (12.18)) converge to the minimum path for all the other sources.

Theorem 12.4 provides a time interval $\tau = n - 1$ of instant communication between two consecutive switching signals that ensures the convergence of the system to the minimum path. Notice that the convergence to the minimum path is by Corollary 12.1 $k \leq n - 1$ instant communications, so the convergence to the minimum path could be observed and detected with the equilibrium condition for $k \leq n - 1$. As previously reported in Watanabe and Watanabe [2] and Bannister and Eppstein [5], the Bellman–Ford algorithm can be optimized by reducing the iteration in more than $n/2$. However, this analysis is out of the scope of this chapter and will be studied in future research.

12.6 Conclusions

In this chapter, the stepping-stone cost is modeled as a multi-agent dynamical system in a vulnerability graph with fixed and switching topology. A biased min-consensus protocol is used for distributed calculation of the shortest path, and because the network is monitored in discrete time, the model is discretized using min-plus algebra for modeling and analysis. Theorem 12.3 and Corollary 12.1 prove that stepping-stone dynamics in a vulnerability graph with fixed topology converge to the shortest path in finite time, given $k = n - 1$ instant communications. Theorem 12.4 provides a metric for the time interval between switching signals that guarantees convergence to the minimum path for switching topology. In future work, we will develop a metric for computing the minimum time interval between two switching signals that ensures convergence to the shortest path for the vulnerability graphs with switching topology. The biased min-consensus protocol used as a model for the stepping-stone cost dynamics is deterministic. We will develop a stochastic model to characterize the dynamics in vulnerability graphs.

Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780 and Office of the Assistant Secretary of Defense for Research and Engineering agreement FA8750-15-2-0120.

References

1 D. M. Nicol and V. Mallapura. “Modeling and analysis of stepping stone attacks.” In: *Proceedings of the Winter Simulation Conference 2014*, pages 3036–3047, Dec 2014.

2 S. Watanabe and Y. Watanabe. “Min-plus algebra and networks.” *Novel Development of Nonlinear Discrete Integrable Systems. RIMS Ko^kyu^roku Bessatsu B*, 47, 2014.

3 G. Cohen, F. Baccelli, G. J. Olsder, and J. P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley. http://www-rocq.inria.fr/metalau/cohen/DES/book-online.html, 2001.

4 J. Y. Yen. “An algorithm for finding shortest routes from all source nodes to a given destination in general networks.” *Quarterly of Applied Mathematics*, **27**:526–530, 1970.

5 M. J. Bannister and D. Eppstein. "Randomized speedup of the Bellman-Ford algorithm." In: *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics, ANALCO '12*, pages 41–47. Society for Industrial and Applied Mathematics, 2012.

6 R. Olfati-Saber and R. M. Murray. "Consensus problems in networks of agents with switching topology and time-delays." *IEEE Transactions on Automatic Control*, **49** (9):1520–1533, Sept 2004.

7 B. M. Nejad, S. A. Attia, and J. Raisch. "Max-consensus in a max-plus algebraic setting: the case of switching communication topologies." *IFAC Proceedings Volumes*, **43**(12):173–180, 2010. 10th IFAC Workshop on Discrete Event Systems.

8 H. Wang, Z. Chen, J. Zhao, X. Di, and D. Liu. "A Vulnerability Assessment Method in Industrial Internet of Things Based on Attack Graph and Maximum Flow," *IEEE Access*, vol. **6**, pp. 8599–8609, 2018.

9 Y. Zhang and S. Li. "Distributed biased min-consensus with applications to shortest path planning." *IEEE Transactions on Automatic Control*, **62**(10):5429–5436, Oct 2017.

10 R. Bellman. "On a routing problem." *Quarterly of Applied Mathematics*, **16**(1):87–90, 1958.