Computer Standards & Interfaces 56 (2018) 74–82

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Cyber-attack path discovery in a dynamic supply chain maritime risk management system

Nikolaos Polatidis*, Michalis Pavlidis, Haralambos Mouratidis

School of Computing, Engineering and Mathematics, University of Brighton, BN2 4GJ, Brighton, United Kingdom

ARTICLE INFO

Keywords:
Cyber-security
Attack path discovery
Risk management system
Maritime supply chain
ISO standards
NIST SP 800-30

ABSTRACT

Maritime port infrastructures rely on the use of information systems for collaboration, while a vital part of collaborating is to provide protection to these systems. Attack graph analysis and risk assessment provide information that can be used to protect the assets of a network from cyber-attacks. Furthermore, attack graphs provide functionality that can be used to identify vulnerabilities in a network and how these can be exploited by potential attackers. Existing attack graph generation methods are inadequate in satisfying certain requirements necessary in a dynamic supply chain risk management environment, since they do not consider variables that assist in exploring specific network parts that satisfy certain criteria, such as the entry and target points, the propagation length and the location and capability of the potential attacker. In this paper, we present a cyber-attack path discovery method that is used as a component of a maritime risk management system. The method uses constraints and Depth-first search to effectively generate attack graphs that the administrator is interested in. To support our method and to show its effectiveness we have evaluated it using real data from a maritime supply chain.

1. Introduction

Modern port infrastructures, tend to be highly dependent on the operation of complex, dynamic IT-based maritime supply chains. Maritime supply chains comprise globally distributed, interconnected set of organizations that involve numerous entities, including other Critical Information Infrastructures (CIIs); such as transport, energy, telecommunication and cyber networks. This emerging landscape of IT-empowered CIIs-based critical infrastructures requires a paradigm shift in the way it assesses risks and vulnerabilities, as most existing risk management methodologies are overly focused on physical-security aspects and pay limited attention to CIIs, remaining oblivious to the complex nature of the IT systems and assets used in the maritime sector, along with their interrelationships and do not adequately take into account security processes associated with international supply chains, which are nowadays IT enabled and therefore severely dependent on intentional and unintentional compromise of CIIs. Hence, in a dynamic environment where constant hardware and software changes take place in different parts of the supply chain IT infrastructure, there is a need for rethinking risk management in the maritime sector by addressing the role of port CIIs and their impact on maritime supply chains, since in a dynamic environment CIIs that include both hardware and software assets constantly change and existing risk management systems fail to address specific network aspects such as entry and target points, propagation length and the location and capability of a potential attacker. In this direction, we developed, a risk management system, called MITIGATE[1], for the dynamic nature of supply maritime chain IT infrastructure. To perform rigorous risk assessments in MITIGATE, it is necessary to identify potential cyber-attacks by constructing the attack graph and performing analysis to identify attack paths [1,2]. In the context of risk management, attack path discovery is important to perform risk assessments and mitigations [2,3]. Attack path discovery is important to identify the attack paths that potential attackers might follow to exploit a network. By identifying the necessary paths, the mitigation of potential threats become more effective.

1.1. Problem definition and contributions

In the maritime supply chain management, it is necessary to perform risk assessments at regular intervals to identify the possibility of cyber-attacks that might occur in the future. Attack path discovery methods,

[1] The acronym MITIGATE stands for Multidimensional, integrated, risk assessment framework and dynamic, collaborative Risk Management tools for critical information infrastructures and is a collaborative research project co-funded by the European Commissions under its biggest Research and Innovation program Horizon 2020.

* Corresponding author.
E-mail addresses: N.Polatidis@Brighton.ac.uk (N. Polatidis), M.Pavlidis@Brighton.ac.uk (M. Pavlidis), H.Mouratidis@Brighton.ac.uk (H. Mouratidis).

http://dx.doi.org/10.1016/j.csi.2017.09.006
Received 21 June 2017; Received in revised form 26 July 2017; Accepted 12 September 2017
Available online 14 September 2017
0920-5489/© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

such as MulVAL, TVA or NuSMV [4–6], can be used to identify attack paths within a network and then, these paths can be used within a risk management system to perform risk assessments and offer potential mitigation solutions. In the literature, several approaches [2–14] can be found that offer attack graph generation and analysis solutions. However, in most, if not all, assessment scenarios, it is not necessary to identify all possible paths within a network, since only certain parts of it typically change in a given period, and usually a risk assessment is performed in specific network parts. In a dynamic environment, it is necessary for managers to be able to identify possible attack paths that satisfy certain constraints, such as the potential location and capability of an attacker, the entry and target points and the propagation length. Existing approaches do not output the most probable paths but rather all network paths resulting in slower analysis times and duplication of analysis. Moreover, the literature fails to provide evaluations of the existing work based on maritime supply chains. Thus, to fill the gap, we deliver:

- A highly parameterized cyber-attack path discovery method that works within a dynamic risk management system to detect the vulnerabilities of the IT infrastructure and to deliver attack paths that satisfy certain criteria. The proposed parameterized method discovers paths in certain network parts, thus making risk assessment specific and faster.
- Extensive evaluation based on synthetic and real data from the maritime supply chain management sector, show that the proposed method performs as well under different scenarios.

1.2. Paper structure

The rest of the paper is structured as follows: Section 2 contains the relevant background. Section 3 presents the proposed method. Section 4 explains the experimental evaluation and Section 5 contains the discussion and Section 6 the conclusions and future work parts.

2. Background

2.1. MITIGATE

MITIGATE is an EU funded project that has a consortium of 12 partners with scientific and industrial background in the maritime port domain. The main goal of MITIGATE is to realize a radical shift in risk management methodologies for the maritime sector toward a collaborative evidence-driven maritime supply chain risk assessment approach that alleviates the limitations of state-of-the-art risk management frameworks. The MITIGATE risk assessment methodology is directly applicable to maritime supply chains, but is not limited to this business sector. The methodology provides a holistic view of the IT infrastructure required for spanning across business partners and organizational boundaries, to identify and evaluate all cyber threats and risks within the supply chain. MITIGATE promotes collaboration between business partners and considers the involvement and importance of the business partners to identify the vulnerabilities, develop an attack path discovery method, perform risk assessments and provide potential mitigation solutions. The attack path discovery method is a link between the vulnerabilities and their mitigation, since it identifies all paths that satisfy the requested criteria. The identified paths are then used for risk assessment and mitigation. In addition, MITIGATE is a standards-based risk management system. In the context of MITIGATE standardization guidance has been followed throughout the development of the platform. Furthermore, the use of standards in risk management has been identified and followed in previous related works [15,16]. In MITAGE guidance from the ISO and NIST standards was applied to deliver a complete platform that identifies relevant attack paths, performs risk assessments and provides migration solutions. Although, discovery of attack paths can be made without standardization procedures, standards assist in platform development by providing guidance that ranges from requirements to implementation. The methodology by design is compliant with international standards (from the ISO27K and ISO28K families) and capitalizes on them and other well-known and proved guidelines and good practices (NIST SP800-30), following standardized notations. In addition, it is implementable, adopting a sequential step-by-step process with clear inputs and outcomes. Standard compliance ranges from management system specification guidelines (ISO 28000:2007), guidance for implementation (28001:2007), risk management process and activities (27005:2011), establishment, implementation, monitoring and review, maintenance and improvement of an Information Security Management System (27001:2013) and guidelines and general principles for initiating, implementing, maintaining, and improving information security management in an organization (27002:2013). Finally, the attacker profiling, as suggested by NIST SP800-30, has been used as a basis for the characterization of an attacker in the risk management system. The qualitative scale of which is ranging from “Low” to “Medium” to “High”.

2.2. Related work in attack graph analysis

In attack graph generation and analysis several approaches can be found in the literature. Typically graph construction takes place within a network to identify all possible attacks paths that can be exploited by attackers to gain unauthorized access to the system [17]. For instance, MulVal is a well-established enterprise network security analyzer that is based on logic [4]. It models software bug interactions along with network configuration, with data supplied by an open source reporting community. Another tool for generating attack graphs can be found in [5] with the name TVA. This is a tool for topological network analysis based on graph dependency exploitation. In [7] the authors propose the use of a general graph model that is based on a specification language proposed by them. Then they create sample attack scenarios using different methods such as substitution, looping and distribution techniques. In [8] the authors implemented an intrusion detection system that produces a graph as output. Another approach is the one found in [9] that uses model checking to analyze a network and find vulnerabilities. NuSMV [6] is another model checking tool that finds vulnerabilities and generates an attack graph. The work proposed by Xinming Ou and-Boyer [10] is yet another logic based approach that uses deduction to form the attack graph.

Solutions that are closer to our method exist, with one found in [6] that uses a Breadth-first search method to identify the vulnerabilities and build the attack graph. Another similar method is [11] that introduces the concept of group reachability to reduce graph complexity based on Breadth-first search. Furthermore, more recent approaches exist and offer different solutions to generate attack graphs. In [12] the authors propose a distributed approach to attack graph generation. This method is based on a multi-agent system, a virtual shared memory abstraction and hyper graph portioning to improve the performance. This method uses a Depth-first search method and the performance is improved with the use of multiple agents after a specific network size. It is also shown that in small network sizes a single threaded approach is faster. In [13] the authors propose the use of a dynamic algorithm that generates an attack graph consisting of the top K paths that there is a probability of being exploited. Also in [3] an approach used exclusively in dynamic risk management is found. This approach uses a Bayesian-based attack graph generation method. Another approach for attack graph generation for risk assessment is the one proposed by Lee et al. [2]. This method provides scalability and is based on a cut and divide method and a series of division rounds and uses Depth-first search to search the smaller graphs. Yet another approach is described in [14], where the authors exploit risk flow within an attack graph for performing security risk assessment.

In the literature, there are specifically tailored methods for risk management and most of the network analysis methods can be used for that after specific configuration. Table 1 provides a list of the relevant related

Table 1
Method features.

Method	Brief description
Ref. [2]	• Risk management • Cut and divide based on division rounds • Depth-first search
Ref. [3]	• Risk management • Bayesian based attack graph generation
Ref. [4]	• Network analysis • Logic based
Ref. [5]	• Network analysis • Graph dependency
Ref. [6]	• Model checking • Find vulnerabilities and generate graph
Ref. [7]	• Network analysis • Sample attacks
Ref. [8]	• Intrusion detection • Graph construction
Ref. [9]	• Model checking • Find vulnerabilities and generate graph
Ref. [10]	• Network analysis • Logic based
Ref. [11]	• Breadth-first search • Group reachability
Ref. [12]	• Depth-first search • Distributed computation
Ref. [13]	• Dynamic • Top K paths
Ref. [14]	• Risk management • Depth-first search • Risk flow within graph

works and gives a brief description of each. In the first column, the reference of each relevant work is listed, whereas in the second column the main characteristics of each method are listed.

3. Attack path discovery in MITIGATE

An attack path is the identification of one or more vulnerabilities that can be exploited by attackers to gain access to specific assets and move between them in a network, thus, forming an exploitable path between the assets. The main goal of the attack path discovery method is to identify the attack paths in specified network fragments of the maritime supply chain infrastructure and use them in MITIGATE for risk management. Furthermore, the attack path discovery method is comprised of the following main components, that the related works mentioned in Section 2 fail to address as a whole:

1 Capability and location of the attacker.
2 Max length.
3 Propagation length.
4 Entry and target points.

The capability of the attacker could be either low, medium or high, while the location of the attacker could be local, adjacent or network based. The max length and the propagation length specify the depth of the graph that will be searched. The entry and target points specify which are the entry points that we want to assess and which are the target points. MITIGATE considers a supply chain being the linked set of resources and processes that begins with the sourcing of raw material and extends through the manufacturing, processing, handling and delivery of goods, products and/or services to the consumer through different transport means and that a supply chain service is a service provided and/or supported by a supply chain. Thus, supply chain service cyber threats indicate how a potential security incident might occur, affecting a specific cyber asset. Furthermore, we define a vulnerability being a weakness or a flaw in an asset, raised either from implementation, design, or other processes that can be exploited or triggered by a threat, an attacker being a person or independently executing program that intends to compromise the confidentiality, integrity, or availability of an asset and an attack being a set of actions that an attacker performs to exploit a vulnerability. Furthermore, the following general guidelines have been followed in the design, development and testing, after careful consideration and requirement analysis. The following guidelines are a general model resulted after the requirement analysis. The guidelines are important since they give a general overview of what the method considers and implements.

1 We consider only cyber threats that can occur from malicious attempts to gain unauthorized access to a network or system.
2 We model the supply chain service as a one-way directed graph and consider only independent attacks and not cyclic attacks.
3 We map the main threat categories to specific vulnerability categories.
4 We use the open NIST national vulnerability repository (although one may use any other open repository).

MITIGATE, as a risk management system needs to take as input information such as the attacker's capability and location, the propagation and max length and the entry and target assets. This is done by the administrator of the system when running various tests to find risks in a network. MITIGATE, consists of a web interface that the system administrator uses to perform different tasks, including attack path identification. In a dynamic environment, where changes are being made constantly an administrator might want to perform risk assessments for a specific network part, thus using the web interface she will enter manually the location and capability of the attacker, the propagation and max length and the entry and target assets to perform different types of risk assessments under different settings to have a broad view of the risks.

3.1. Proposed method

According to the definitions, the attacker profile, the attacker location and the association rules the algorithm executes a series of steps to construct the attack graph and identify the paths.

The steps are:

Step 1. Load the data from the graph database and store it in a graph data structure.

- In both cases that the tool is used as standalone or as a component of MITIGATE the database details, such as the username, password and connection details need to be present in the source code of the system.

Step 2. Load the capability and location.

- If the method is used as standalone these data need to be entered manually in the source code. In the case that the tool is used within the MITIGATE platform then these are entered through the web interface.

Step 3. Delete all the assets from the graph that the attacker does not have the capability or location level required.

- If the attacker does not have the capability or the required location status then assets that are do not fall into this category are removed from the final graph.

Step 4. Specify the max and propagation lengths and delete all the assets that are not within the lengths.

- If the method is used as standalone these data need to be entered manually in the source code. In the case that the tool is used within the MITIGATE platform then these are entered through the web interface.

Algorithm 1
Attack path discovery.

Input: Asset graph
Output: Affected assets, attack paths

```
#Load the graph and store it in memory
Load Asset graph from database
    Create Graph data structure G and store the asset graph
#Load the attacker location and capability
Load attacker location and capability
  for e in parameters entry points
   If attacker location < required level of attacker location OR attacker capability < required attacker capability
   return empty graph
    else
    get single source shortest path length
#Set the maximum and propagation lengths
set max and propagation length
    for entry point e
        for target point t
#Create a list with all non-circular paths from entry e to target t
#Search the graph up to the specified length
get all paths in the graph G from entry e to target t that are up to the pre-specified path length
    for the size of paths found
       add paths to attackpaths [] list
       add affected assets to affectedassets [] list
return attack paths, affected assets
```

Table 2
Attacker capability.

Qualitative values	Description
High	The attacker is an expert and has the sufficient resources to perform an attack.
Medium	The expertise and the resources of the attacker are of a moderate level.
Low	The attacker has limited resources and expertise to perform a successful attack.

Step 5. Specify which the entry and target assets are.

- If the method is used as standalone these data need to be entered manually in the source code. In the case that the tool is used within the MITIGATE platform then these are entered through the web interface.

Step 6. Create a new graph containing all the remaining assets according to the above steps.

- The final graph is loaded into memory using a graph data structure.

Step 7. Use Depth-first search to identify all non-circular attack paths in the newly formed graph.

- Depth-first can be efficiently used to identify the attack paths.

The pseudocode of the method is shown in Algorithm 1.

3.2. Attacker profile and location

This section describes the profile and location of an attacker. The main attacker profile is identified based on their capability, knowledge and expertise in coordinating, executing and succeeding an attack. The characteristics are based on the NIST SP800-30 guide but only utilized three levels of expertise. Table 2 provides the characteristics of the attacker's capability. Furthermore, the attacker could be an "inside" or "outside" attacker. The location will influence which assets will be potential entry points. Table 3 provides the characteristics of the attacker's location.

In attack path discovery is important to be able to use the profile and location of the attacker. The identified vulnerabilities found in cyber systems can be exploited by attackers possessing a certain level of expertise and by their location in the system. All the information about the expertise and the location of an attacker should be available for risk identification and mitigation, since certain types of vulnerabilities need to be prioritized according to the current defence strategy.

3.3. Rules

Attack path identification follows certain rules that belong to two different phases. The first phase is the knowledge base of the system and remains constant unless changes in the assets or in their connections are made. Moreover, the second phase is the path construction phase which follows a set of rules as well.

Phase 1: Knowledge base

Rule 1. Assets connected using a symmetric relationship.
$\forall \text{asset1,asset2}\quad \text{Connected(aseet1,aseet2)} \Leftrightarrow \text{Connected(asset2,asset1)}$.

Rule 2. Assets connected using a common network.
$\forall \text{asset1,asset2,net ConnectsTo(asset1,net)} \wedge \text{ConnectsTo(asset2,net)} \Rightarrow \text{Connected(asset1,asset2)} \wedge \text{Connected(asset2,asset1)}$.

Rule 3. Assets connected to vulnerabilities using a symmetric relationship.
$\forall \text{vuln1,vuln2,asset}\quad \text{Connected(vuln1,asset,vuln2,asset)} \Rightarrow \text{Connected(vuln2,asset,vuln1,asset2)}$

Rule 4. Vulnerabilities connected to assets using a symmetric relationship.
$\forall \text{vuln1,vuln2,asset1,asset2}\quad \text{Connected(vuln1,asset1,vuln2,asset2)} \Rightarrow \text{Connected(vuln2,asset2,vuln1,asset1)}$.

Rule 5. Exploiting different vulnerabilities affecting the same asset.
$\forall \text{vuln1,vuln2,asset}\quad \text{Vulnerability(vuln1,asset)} \wedge \text{Vulnerability (vuln2,asset)} \Rightarrow \text{Connected(vuln1,asset,vuln2,asset)} \wedge \text{Connected (vuln2,asset,vuln1,asset)}$.

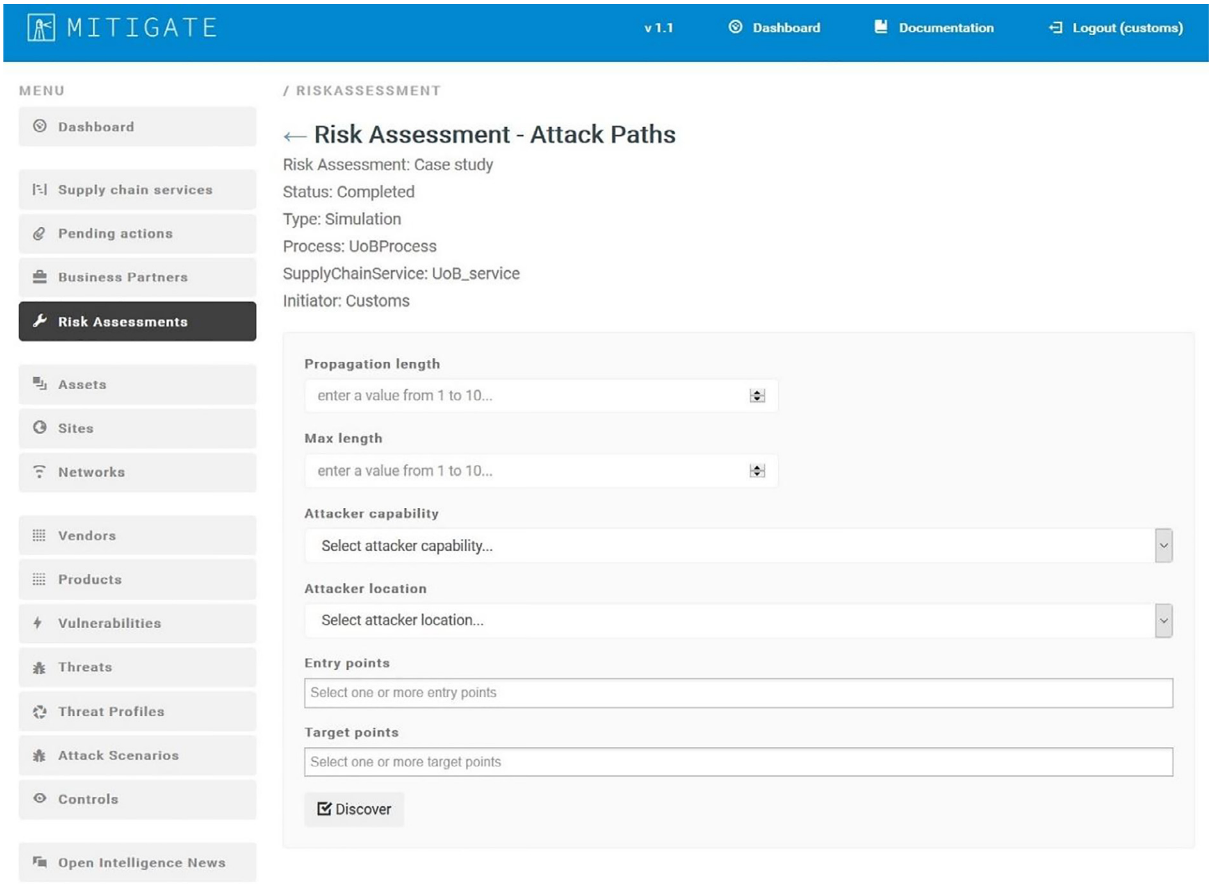
Rule 6. Relationship between vulnerabilities of connected assets.
$\forall \text{vuln1,vuln2,asset1,asset2}\quad \text{Vulnerability(vuln1,asset1)} \wedge \text{Vulnerability(vuln2,asset2)} \wedge \text{Connected(asset1,asset2)} \Rightarrow \text{Connected(vuln1,asset1,vuln2,asset2)} \wedge \text{Connected(vuln2,asset2, vuln1,asset1)}$.

Phase 2: Path construction

The second phase is the path construction phase follows rules that belong to the propagation and allow us to determine which vulnerabilities

Table 3
Attacker location.

Qualitative values	Description
Local	The attacker is located within the network.
Adjacent	The attacker is in an adjacent network that currently communicates with the target network.
Network	The attacker is in a wider network such as the Internet.



← Risk Assessment - Attack Paths

Risk Assessment: Case study
Status: Completed
Type: Simulation
Process: UoBProcess
SupplyChainService: UoB_service
Initiator: Customs

Propagation length: enter a value from 1 to 10...
Max length: enter a value from 1 to 10...
Attacker capability: Select attacker capability...
Attacker location: Select attacker location...
Entry points: Select one or more entry points
Target points: Select one or more target points

Fig. 1. MITIGATE - attack path discovery interface.

the attacker can use to further penetrate the supply chain infrastructure to reach a predetermined target or to cause general damage by affecting as many assets as possible from a given entry point/asset. In their current form, the rules don't construct the whole path but merely indicate for every connection if it could be used for the propagation of the attacker. For convenience, the rules are divided into subgroups. The rules to discover which vulnerabilities are accessible by the attacker are:

Rule 1. An existing vulnerability (whether confirmed or a zero-day one) on an asset is accessible by an attacker if its AV is 'Network' (i.e. remotely exploitable) and both the asset and attacker are connected to the same network (e.g. the Internet).
∀vuln,asset,attacker,loc Network(attacker,loc)∧ConnectsTo(asset,loc)∧Vulnerability(vuln,asset)∧Network(vuln)⇒Accessible(vuln,asset,attacker).

Rule 2. An existing vulnerability on an asset is accessible by an attacker if its AV is 'Adjacent Network (i.e. exploitable over local network) and both the asset and attacker are connected to the same local network.
∀vuln,asset,attacker,loc AdjacentNetwork(attacker,loc)∧ConnectsTo(asset,loc)∧Vulnerability(vuln,asset)∧(AdjacentNetwork(vuln)∨Network(vuln))⇒Accessible(vuln,asset,attacker)

Rule 3. An existing vulnerability on an asset is accessible by an attacker if its AV is 'Local' (i.e. requires physical access or local account) and attacker has physical access or a local account.
∀vuln,asset,attacker,localAccount Local(attacker,asset)∨AccessRight(attacker,localAccount)∧Vulnerability(vuln,asset)∧(Local(vuln)∨AdjacentNetwork(vuln)∨Network(vuln))⇒Accessible(vuln,asset,attacker)

Rule 4. An existing vulnerability on an asset is accessible by an attacker if it can be reached via some other vulnerability on the system.
∀vuln1,asset1,vuln2,asset2,attacker Traversable(vuln1,asset1,asset2,attacker)∨Traversable(vuln,asset1,vuln2,asset2,attacker)⇒Accessible(vuln2,asset2,attacker).

4. Experimental evaluation

To perform the experimental evaluation, we used a Pentium i7 with 12GBs of RAM running windows 10. Furthermore, we developed the proposed attack path discovery algorithm using Python and Neo4J. The evaluation has been based in two different case studies. In the first we have used synthetic data to give a step by step description to show the relationships between assets and vulnerabilities and that the algorithm can identify the relevant attack paths under the specified settings. In

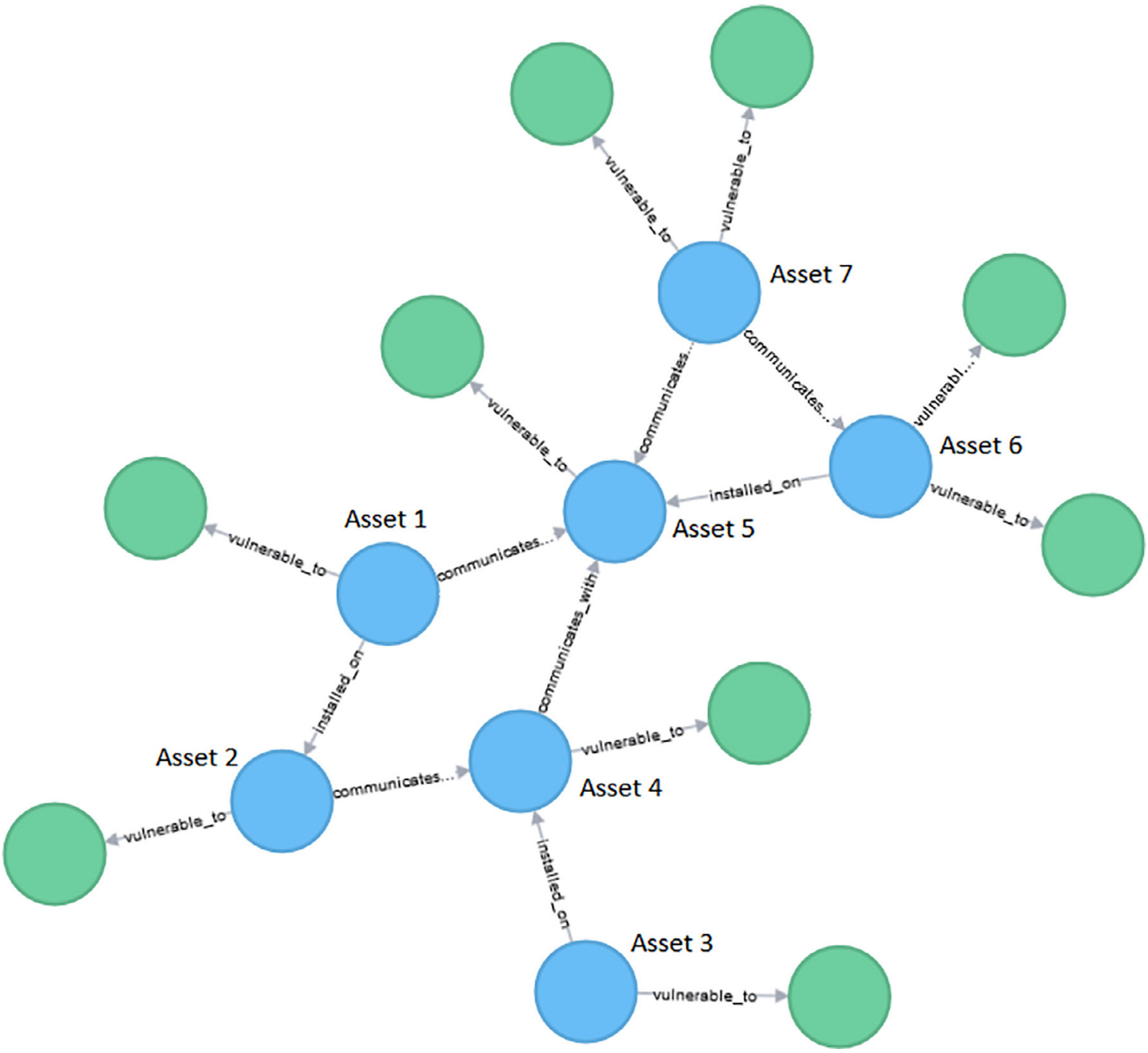


Fig. 2. Case study graph database.

the second case study, we have used real data about assets, vulnerabilities and relationships supplied by the port of Valencia to show how the algorithm performs. Furthermore, the attack path discovery interface component of the MITIGATE system is shown in Fig. 1. In the figure, it is shown that the administrator has the option to enter all the necessary information and by pressing the 'Discover' button the algorithm is executed.

4.1. *Case study: synthetic data*

Initially, we developed a case study that is based on seven assets and nine vulnerabilities. We have used seven assets and nine vulnerabilities to have at least one asset being vulnerable to one or more vulnerabilities. Moreover, each vulnerability can be exploited by an attacker with specific knowledge and location. In the graph in Fig. 2 blue circles represents assets and green represent the vulnerabilities and the data used for this part are synthetic. According to our proposed method each asset is vulnerable to one or more vulnerabilities in this scenario. Furthermore, the types of relationships between assets and vulnerabilities as set from the MITIGATE consortium. These are:

- Communicates with (asset to asset communication).
- Installed on (asset, app or similar, installed on asset).
- Vulnerable to (asset is vulnerable to vulnerability).

In the settings of the proposed method several different settings must be set before the execution of the algorithm. These are:

- Attacker capability (Low, Medium or High).
- Attacker location (Local, Adjacent or Network).
- Propagation length (The length of the propagation).
- Max length (The max length in the network that an attacker could go into).
- Entry points (Which the entry points are).
- Target points (Which the target points are).

For the case study and to validate the quality of the algorithm we have run three different tests:

Test 1. In this test, we have set the attacker capability to 3 (high), the location to 1 (local), max length to 2 and propagation length to 2. Additionally, we have set the entry points as assets 1 and 7 and the target points as assets 2 and 5. According to the information form the assets, the vulnerabilities, the relationships and the settings used two paths should be identified. Subsequently, the following paths have been identified by the algorithm: [1,2,5].

Test 2. In this test, we have set the attacker capability to 1 (low), the location to 2 (adjacent), max length to 2 and propagation length to 2. Additionally, we have set the entry point as asset 3 and the target points as assets 6 and 7. According to the information form the assets, the vulnerabilities, the relationships and the settings used no paths should

Table 4
Synthetic data case study quality results.

No of test	No of present paths	No of paths found	Paths
1	2	2	[1, 2, 1, 5]
2	0	0	[empty list]
3	1	1	[3, 4, 5, 7]

be identified. Subsequently, the algorithms returned the following message: Attacker cannot exploit entry point: 3 [empty list].

Test 3. In this test, we have set the attacker capability to 3 (high), the location to 2 (adjacent), max length to 3 and propagation length to 3. Additionally, we have set the entry point as asset 3 and the target points as assets 6 and 7. According to the information form the assets, the vulnerabilities, the relationships and the settings used one path should be identified. Subsequently, the following path has been identified by the algorithm: [3–5, 7].

The output of the three tests of the case study show that the algorithm identifies the desired attack paths according to the settings specified in the risk management system. At the end, the algorithm returns the paths or an empty list to clearly show the quality of the result. Table 4 shows the results of the case study where for each test it is shown how many paths exist and how many parts the method found. Moreover, in the last column the paths are displayed.

4.2. *Case study: the port of Valencia*

Valencia port is a port community system that is located at the port of Valencia and is managed by the port authority of Valencia. The system allows actors who are involved in the transportation of goods to connect and exchange information. The current information system is comprised of 26 assets including both hardware and software assets. Using the information supplied by the port of Valencia we have executed a series of tests to validate the performance of the attack path discovery method, the results of which are presented in Table 5. In the table, the propagation length and max length refer to the length that the respective steps that an attacker could made according to the risk assessment scenario. The entry points refer to how many entry points have been specified and the target points refer to the number of target points the attacker wants to reach from the entry points. Moreover, it is can also be specified which the entry and target points are. Finally, the number of paths found is the exact number of attack paths found by the algorithm according to the settings uses and the time in seconds is the exact processing time of the algorithm.

Furthermore, based on the data from the port of Valencia we developed a realistic database consisting of 182 assets. The assets include 35 hardware assets, 147 software assets installed evenly on the hardware assets and vulnerabilities associated to various software assets. Then further associations have been made to form a network between the hardware assets. The supplementary performance results based on the 182 assets are presented in Fig. 3. On the left part of the figure from 0 to 5 the seconds are represented, the terms low, medium and high represent the capability of the attacker and in all cases the location of the attacker has been set to local. Additionally, the values 5, 10, 20, and 50 represent both the number of the entry and target points. Lastly, the max and propagation length values have been both set to 10 in all cases.

5. Discussion

The proposed method is tailored to dynamic risk management for graph construction and analysis. This takes place within MITIGATE and although it takes features from methods found in the literature it provides unique characteristics [2,3,12,14]. When comparing the proposed method with other methods we can easily identify the main similarities as shown in the first column of Table 6 and the unique differences in the second column of Table 6. Both the similarities, but more importantly, the differences are a central part of attack path identification in dynamic supply chain environments. In a dynamic environment, there are small but constant changes of hardware and software assets within the network. Thus, it is important for risk management to be able to have options such as the entry and target points to be able to perform assessments in various network parts as these are constantly changed or updated.

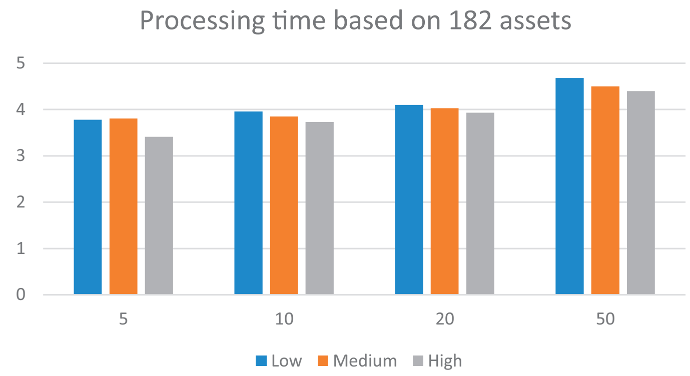


Fig. 3. Performance evaluation based on 182 assets.

Attack path discovery is a major link between components in risk management systems [2,14]. Cyber-attacks can be mitigated using the discovered attack paths for risk assessment and by offering potential mitigation solutions. In MITIGATE the proposed attack path discovery method differs from the ones proposed in the literature since it can be effectively used to find attack paths from specific entry points that aim to attack specific target points, take into consideration characteristics such as the location and the capability of the attacker and the propagation length. Methods for attack paths identification used for risk assessment such as [2,3,14] do not take into consideration all these characteristics and are not suitable for the supply chain management of the maritime domain. Although, other network-based methods could be used such as [4–6], these would not give the optimum outcome since many modifications will be necessary for those to work in MITIGATE. Thus, the proposed method provides specific graph generation that contains certain assets within a specific network part, which leads to risk management in specific network parts. The maritime supply chain management is a dynamic environment constantly, where several parts form the supply chain. In the IT infrastructure of the supply chain only certain software or hardware assets will typically change within a given period. The main difference of the proposed attack path discovery algorithm is the generation of specific attacks graphs that contains only portions of the network, which leads to smaller graphs and smaller search spaces. Additionally, it has been discussed before that pure Depth-first search is more efficient in smaller search spaces when compared to alternatives such as distributed technologies [12]. Our method assists in risk management by providing attack paths for the specified assets and characteristics, although it can still be used to provide all attacks paths. Additionally, risk management in dynamic environments requires the use of a highly-parameterized attack path discovery algorithm due to its dynamic nature. In the maritime supply chain domain MITIGATE offers a solution to risk management due to its suitability for this specific domain. Our attack path discovery method complements MITIGATE by integrating with the system and discovering attack paths between assets that satisfy certain requirements. An administrator of the system could enter or update the data in the database and then check specific parts of the system for vulnerability exploitation, without the need of checking the whole network.

To show that our proposed method is both practical, effective and suitable for its purpose we have conducted two case studies. In the first case study, we have used a small amount of synthetic data to show how

Table 5
Port of Valencia case study performance results.

No. of test	Attacker capability	Attacker location	Propagation length	Max length	No. of entry points	No. of target points	Time in seconds
1	Low	Local	3	3	2	2	1.27
2	Low	Local	3	3	3	3	1.65
3	Low	Adjacent	3	3	2	2	0.95
4	Low	Adjacent	3	3	3	3	0.97
5	Low	Network	3	3	2	2	0.90
6	Low	Network	3	3	3	3	0.82
7	Medium	Local	4	4	2	2	0.94
8	Medium	Local	4	4	3	3	0.98
9	Medium	Adjacent	4	4	2	2	0.90
10	Medium	Adjacent	4	4	3	3	0.81
11	Medium	Network	4	4	2	2	0.90
12	Medium	Network	4	4	3	3	1.07
13	High	Local	5	5	4	4	0.81
14	High	Local	5	5	5	5	0.94
15	High	Adjacent	5	5	4	4	0.92
16	High	Adjacent	5	5	5	5	1.10
17	High	Network	5	5	4	4	1.10
18	High	Network	5	5	5	5	1.10
19	High	Local	3	3	26	26	1.35
20	High	Local	4	4	26	26	1.60
21	High	Adjacent	3	3	26	26	1.31
22	High	Adjacent	4	4	26	26	1.33
23	High	Network	3	3	26	26	1.42
24	High	Network	4	4	26	26	1.36
25	High	Local	8	8	26	26	1.92
26	High	Local	10	10	26	26	1.95
27	High	Adjacent	8	8	26	26	1.65
28	High	Adjacent	10	10	26	26	1.93
29	High	Network	8	8	26	26	1.52
30	High	Network	10	10	26	26	1.64

Table 6
Method similarities and differences.

Similarities between our method and others	Differences between our method and others
1. Developed with risk management in mind. 2. Attack path generation based on attack graph input. 3. Based on depth-first search.	1. Takes as input specific entry points and target points. 2. Takes as input the capability of the attacker. 3. Takes as input the location of the attacker. 4. The maximum and propagation lengths are specified.

the assets and vulnerabilities are related within the graph database, to explain step by step how the algorithm works and to show that the algorithm identifies existing attack paths under the specified settings. In this case study, we conducted three different tests with random choosing of entry points, target points and propagation length. Table 4 presents the results of the case study, where it is shown that the number of present attack paths under the specified settings are identified by the proposed method. For the second case study, we have used real data from the port of Valencia. It is shown that with 26 assets the algorithm performs well and achieves its accomplishment in about 1 s. Although, it starts from 0.7 second it grows larger than 1 s as the network settings change and the algorithm searches through more nodes. Although, it should be noted that depending on the relationships between assets and vulnerabilities the processing time will vary. In our experiments, we have selected random combinations from the data supplied by the port of Valencia regarding which the entry and target points are. Different combinations of entry and target points will result to different outputs and different processing times. However, the point of the algorithm is indeed to search different combinations as these are specified by the administrator of MITIGATE for specific network parts within the supply chain maritime domain. In this domain, parts of hardware and software are constantly added, removed or updated and risk assessments need to take place for specific network domains. Additionally, this should happen within a reasonable amount of time. The results from the second case study are presented in Table 5 and show how the method performs under different settings. However, the processing time could vary depending on how many entry and target points have been specified, which these points are and the propagation and max length.

We have shown that our proposed cyber-attack path discovery method is both practical and effective. It can identify relevant attack paths under specified conditions and contribute to risk management. It is vital in dynamic maritime supply chain maritime environments to provide a method that discovers attack paths for specific part in an effective and efficient way. As part of the MITIGATE risk management system attack path discovery is a link between the identification of possible vulnerability exploitation, attack path identification and mitigation. A potential administrator of the MITIGATE system can successfully identify relevant attack paths under the specified settings, which is important for critical infrastructure protection in a constantly changing dynamic environment.

6. Conclusions and future work

The use of information systems is a vital infrastructure part of various domains including maritime supply chain. However, with supply chain being a dynamic domain in terms of its IT infrastructure the use of a risk management system is necessary to identify and mitigate attacks. In this direction, we have developed MITIGATE, a risk management system that fits well in a constantly changing dynamic environment and that automates the process of risk assessment and mitigation. In this context, we have proposed a highly parameterized cyber-attack path discovery method. The proposed method works as a component of the risk management system to identify attack paths that exist under certain conditions and has been evaluated for its quality and performance using both synthetic data and real data supplied by the port of Valencia. Both the quality and privacy results show that the proposed method is both practical and effective.

As a future work, we aim to investigate the following research directions:

Attack mitigation. This research direction will investigate the possibility of applying recommendation methods to mitigate attacks in real time. In the case the administrator runs a risk assessment and identifies that certain vulnerabilities can be exploited and attack paths exist but no actions have been taken, then the attack mitigation method will assist by providing defence strategies.

Attack prediction. In this step. we will investigate the use of recommender system with the use of previous data from the system to predict future attacks. The attack predictions will be based on a combination of information from vulnerabilities in assets, attack paths between assets, the importance of the target asset and previous attack information.

Acknowledgment

This work has received funding from The European Union's Horizon 2020 research and innovation program under grant agreement No 653212.

References

[1] K.E. Lever, K. Kifayat, Risk assessment and attack graph generation for collaborative infrastructures: a survey, Int. J. Crit. Comput. Syst. 6 (2016) 204–228, doi:10.1504/IJCCBS.2016.079081.

[2] J.L.J. Lee, H.L.H. Lee, H.P. In, Scalable attack graph for risk assessment, 2009 Int. Conf. Inf. Netw, 2009.

[3] N. Poolsappasit, R. Dewri, I. Ray, Dynamic security risk management using bayesian attack graphs, IEEE Trans. Dependable Secur. Comput. 9 (2012) 61–74, doi:10.1109/TDSC.2011.34.

[4] X. Ou, S. Govindavajhala, A.W. Appel, MulVAL: a logic-based network security analyzer, in: Proc. 14th Conf. USENIX Secur. Symp. Vol. 14, 2005: pp. 8–8.

[5] S. Jajodia, S. Noel, B. O'Berry, in: Topological Analysis of Network Attack Vulnerability, Manag. Cyber Threat., 2005, pp. 247–266, doi:10.1145/1229285.1229288.

[6] P. Ammann, D. Wijesekera, S. Kaushik, Scalable, graph-based network vulnerability analysis, Proc. 9th ACM Conf. Comput. Commun. Secur. CCS '02. (2002) 217. doi:10.1145/586110.586140.

[7] S.J. Templeton, K. Levitt, A requires/provides model for computer attacks, in: Proc. 2000 Work. New Secur. Paradig. NSPW '00, 2000, pp. 31–38, doi:10.1145/366173.366187.

[8] P. Ning, D. Xu, Learning attack strategies from intrusion alerts, in: Proc. 10th ACM Conf. Comput. Commun. Secur. CCS '03, 2003, p. 200, doi:10.1145/948134.948137.

[9] R.W. Ritchey, P. Ammann, Using model checking to analyze network vulnerabilities, secur. privacy, 2000. S&P 2000, in: Proceedings. 2000 IEEE Symp., 2000, pp. 156–165, doi:10.1109/SECPRI.2000.848453.

[10] M.A.M. Xinming Ou, WayneF. Boyer, A scalable approach to attack graph generation, in: 13th ACM Conf. Comput. Commun. Secur., 2006, pp. 336–345.

[11] K. Ingols, R. Lippmann, K. Piwowarski, Practical attack graph generation for network defense, in: Proc. - Annu. Comput. Secur. Appl. Conf. ACSAC, 2006, pp. 121–130, doi:10.1109/ACSAC.2006.39.

[12] K. Kaynar, F. Sivrikaya, Distributed attack graph generation, IEEE Trans. Dependable Secure Comput. 13 (2016) 519–532, doi:10.1109/TDSC.2015.2423682.

[13] K. Bi, D. Han, J. Wang, K maximum probability attack paths dynamic generation algorithm, Comput. Sci. Inf. Syst. 13 (2016) 677–689, doi:10.2298/CSIS160227022B.

[14] K. Zheng, B. Wu, F. Dai, Y. Hu, Exploring risk flow attack graph for security risk assessment, IET Inf. Secur. 9 (2015) 344–353 doi:10.1049/iet-ifs.2014.0272.

[15] X. Larrucea, C. Gonzalez-Perez, T. McBride, B. Henderson-Sellers, Standards-based metamodel for the management of goals, risks and evidences in critical systems development, Comput. Stand. Interfaces 48 (2016) 71–79. https://doi.org/10.1016/j.csi.2016.04.004.

[16] B. Barafort, A.L. Mesquida, A. Mas, Integrating risk management in IT settings from ISO standards and management systems perspectives, Comput. Stand. Interfaces. (2016) 1–0, doi:10.1016/j.csi.2016.11.010.

[17] X. Ou, A. Singhal, in: Attack Graph Techniques, Quant. Secur. Risk Assess. Enterp. Networks, 2011, pp. 13–23. doi:10.1007/978-1-4614-1860-3.