Security and Safety Composition Methodology

Seraj Fayyad
Department of Informatics
UNIK, Oslo University
Oslo, Norway
seraj@unik.no

Josef Noll
Department of Informatics
UNIK, Oslo University
Oslo, Norway
josefnoll@gmail.com

***Abstract*—One of the main aspects of modern life is the interaction with sensors and other embedded systems. These systems become increasingly more integrated with daily life activities. They enable the interaction between a variety of components or parties (people, actuator, sensor, software, etc.). This interaction causes the appearance of new challenges in the design of security-related aspects of embedded systems. This paper uses the SHIELD methodology of the JU Artemis to provide Security, Privacy and Dependability (SPD) levels of embedded systems. We propose an extension of the methodology to take into consideration interactions between components, and introduce functions describing the significance of the interconnections. The complete methodology enables the composition of SPD as an add-on or as a built-in feature, and is thus applicable to an already built embedded system or to the development of embedded systems.**

Keywords - security composition; safety composition; component interconnection; sensor security; IoT; security attributes; privacy.

I. INTRODUCTION

The Future Internet is transforming from what it is today (a mere communication highway) into a backend system connecting hybrid networks. These hybrid networks will connect people, services, things (sensors, actuators) and computers all together; in other words, it will create the Internet of People, Things and Services (IoPTS) [1]. IoPTS developed from the Internet of Things (IoT) terminology [2], which has traditionally been a subject of intensive research in the area of computing and networking, taking into consideration the impact of things such as sensors, actuators and devices.

IoPTS is seen as an integrator of processes in the domain of computing, communication, data storing, monitoring and control of entities in the physical world. The integration of heterogeneous processes creates new challenges, especially related to security. Process execution without controlling security features will potentially impact people or services, and may create economic loss, reduce privacy or in the worse case scenario, harm human life.

We consider the IoPTS as a system of interacting embedded systems, with the embedded systems being the central unit in the IoPTS. While specific interest is usually given during the design of embedded systems, operational aspects of security are often less announced. Moreover, life-time security of embedded systems is often neglected during system development, thus leaving many devices vulnerable to attacks. The growing number of embedded systems nowadays (mobile phones, smart TVs, household appliances, home automation products, industrial monitoring, control systems, etc.) make them interesting targets for criminal activities. On the other hand, the implementation of security and safety measures is not easy due to the constraints on resources of this kind of systems.

Addressing embedded systems security and safety challenges is considerably complex, resulting from a variety of factors. One of these factors is the interconnection between system components. From the functionality perspective, the interconnection between system components allows a system to provide its services. From the dependability perspective, this interconnection causes the same system to fail as a result of a defect in one specific component, even though other components are working properly. From the security point of view, component interconnection means that a security problem in one component could lead to a problem within other related components. Let us consider a smart vehicle embedded system, as an example: If a vehicle owner can remotely turn the vehicle on/off using a particular software library, vulnerabilities in this software library might be used by a 3rd party to influence the engine availability.

Novel research by the SHIELD projects in the joint undertaking JU ECSEL (former JU ARTEMIS) suggests a distributed architecture for embedded system safety and security [3]. Semantic descriptions of each aspect and a semantic overlay are the core of the SHIELD methodology enabling measurable and composable security. The descriptions include attack capabilities, security functionalities, system components, and security in the perspective of a system of systems.

The supporting SHIELD ontologies are built on the decomposition of the embedded system into components. For each component, the SPD needs are identified as attributes. For the identified SPD needs, the possible techniques (functionalities) to address these needs are identified. Figure 1 provides the view on how embedded systems can be enriched through security attributes to create systems with security, privacy and dependability functionality.

 ISBN: 978-1-61208-369-8

An embedded system consists of system components, e.g., communication, where each component has certain SPD attributes, e.g., encrypted communication. Through the SHIELD methodology, the attributes are then transferred into

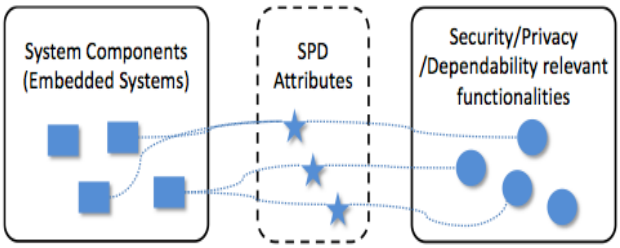

Figure 1. Relation between SPD functionalities and attributes for system component [3].

SPD functionalities, e.g., protection against man-in-the-middle attack. Addressing functionalities through component attributes thus allows matching required security demands of an application.

Possible SPD functionalities addressing SPD attributes could be encryption, redundancy and hash coding (for person ID), satisfying the requirements for confidently, availability, and anonymity of data, for example.

This paper analyzes the interconnection between components, and the impact of the interconnection on SPD attributes. The paper proposes a new composition methodology for SPD functionalities to enhance the SPD level of the embedded system.

The paper is organized as follows: In Section 2, we give an overview of related work. In Section 3, we provide some examples about the impact of interconnection on SPD attributes. In Section 4, we propose our extension of the SHIELD methodology considering components interconnection and SPD attribute interconnection. Section 5 provides a potential use case, and Section 6 presents the composition algorithm for that use case. In Section 7, we present our conclusion.

II. RELATED WORK

Organizations, typically, apply risk management processes for their information technologies in order to mitigate Information Technology (IT) related risks. This way, they assure that their organizational enterprise operates in accordance to their risk appetite [4]-[6].

Recent improvements enhance system security by building security models in their risk assessment. Through security models, they identify attack paths potentially leading to damage. A majority of researchers suggest attack graphs [7]-[9], while others use attack trees to display the attack potential [10].

Attack graphs modeled by administrators could be used to harden a system network through finding critical vulnerabilities whose removal can prevent potential attacks and so improve system SPD level [6][11].

Schneier [5] demonstrated how the attack graphs can help in designing network security metrics. Wang et al. [12][13] propose an attack graph-based probabilistic metric model to quantify the overall security of their network. Xie et al. [14] performed a security risk analysis using Bayesian networks [16] incorporated with Intrusion Detection System (IDS) alerts.

Traditional research concentrates mostly on the discovery of system vulnerabilities and the relation between these vulnerabilities, having the goal to harden the system and to enhance system SPD level. They model these vulnerabilities and their relations to propose security metrics, often based on one semantic description for the specific system.

In this paper we concentrate on the analysis of system components, their interaction and relations to harden the system. Individual ontologies are used to describe the individual embedded systems, their components, SPD attributes and security functionalities. Hardening is achieved through increasing the SPD level of the system by prioritizing system SPD functionalities (e.g., encryption) in the composition process. This prioritization helps system engineers to suit SPD functionalities add-on or built-in based on their SPD cost readiness, so they end up with better SPD level using the same cost.

III. COMPONENTS INTERCONNECTION IMPACT ON SPD ATTRIBUTES

The interconnection between system components makes the SPD attributes of these components dependent on each other. Some examples are provided on how interconnection between system components affect the SPD attributes of these components and their interconnections.

A. *Authentication impact availability*

Let us consider the smart vehicle case, where the vehicle owner could turn the vehicle remote on/off. An attacker might know the authentication and authorization security attributes and their functionalities. Thus, exploiting the vulnerabilities of authentication or authorization functionalities will enable him to take control of the vehicle engine. Thus, the interconnection between remote on/off applications and the engine control makes the availability (SPD attribute) of the engine as a system component dependent on the authentication attributes of the remote application component.

B. *Confidentiality impact privacy*

Let us consider the case of the smart vehicle, where the vehicle owner can monitor the vehicle remotely. With successful attack on the monitoring component confidentiality, the privacy of the vehicle rider is revealed.

C. *Reliability impact privacy*

Let us consider again a smart vehicle, with remote monitoring capabilities that might have engine, position and speed monitoring as part of the functionality. A standard operation would include engine monitoring, while protecting the driver's privacy through hiding speed and position. Improper activation of the monitoring component caused by a functional problem will cause unwanted exposure of the driver's privacy.

 ISBN: 978-1-61208-369-8

D. Reliability impact authorization

Let us consider again the case of the smart vehicle, with the remote engine control. Incorrect configuration of the remote engine control or the communication protocol might allow a 3rd-party to interfere with the vehicle engine, and thus make the vehicle vulnerable. This example demonstrates that dependability attributes, such as reliability, may impact security attributes, such as authorization.

IV. SHIELD ONTOLOGY CONSIDERING COMPONENTS INTERCONNECTION

As indicated in the previous use cases, interconnection between system components may influence SPD attributes of other interconnected components. Although the SHIELD methodology enables measurable and composable SPD of embedded systems (see Figure 1), aspects of component interconnection are not that well announced. We propose an extension of the SHIELD methodology, concentrating on component interconnections as well as their impact on SPD attributes.

A. Extension of the SHIELD component ontology

For the component ontology, we define component interconnection graphs as a representation for system components and their interconnections. Through these graphs, we considered interconnections between system components being both data and control transactions.

Component interconnection graphs are frequently being used in modeling distributed software architectures [15]. In our paper, we define component interconnection graphs as direct graphs, consisting of one type of vertice, which is a *component*, and two type of edges, which are *data* and *control*. The direction in the graph reflects the relation direction. For instance, if the *data* edge point connects from *C1* to *C2*, it means that the component *C1* sends data to/through *C2*. If the *control* edge point from *C2* to *C3*, it means that component *C2* sends control commands to/through C3. Component interconnection graph concepts are illustrated in Figure 2 and formally characterized in Definition 1.

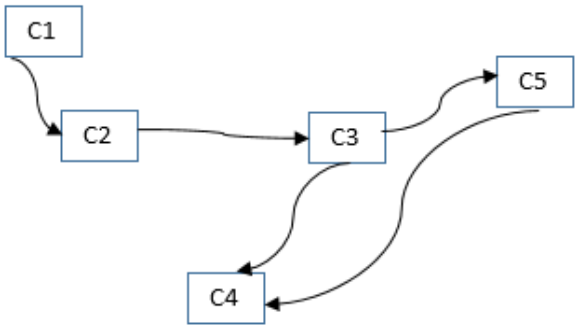

Figure 2. Illustration of a component interconnection graph.

Definition 1. Given a set of Components C, having a set of control relations $R_c \subseteq C \times C$, and a set of data relation $R_d \subseteq C \times C$, then the components interconnection graph G is the directed graph G (C, $R_c \cup R_d$), where C is the vertices set and $R_c \cup R_d$ the edge set.

B. Extension of the SHIELD SPD attribute ontology

In this paper, we define security attributes interconnection graphs as representation for component SPD attributes and their interconnections. Our security attributes interconnection graph is a directed graph having one type of vertices attribute A and one type of edges IMPACT. The direction in the graph reflects the relation direction, e.g. an edge pointing from A to B means that attribute A *IMPACTs* attribute B.

SPD attributes interconnection graph edges are derived from the components interconnection graph edges. If an edge exists between two SPD attributes (vertices) then an edge must exist between these two attribute components in the component interconnection graph.

Security attributes interconnection graphs concepts are illustrated in Figure 3 as SPD attributes, and formally characterized in Definition 2.

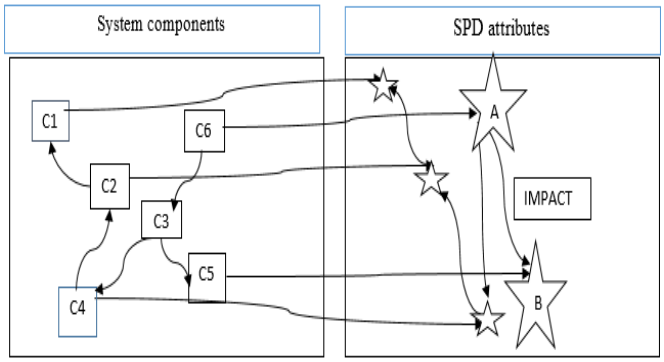

Figure 3. Security attributes interconnection graphs extending the SHIELD methodology.

Definition **2.** Given a set of attributes A, which represent a set of impact relations, $R_{impact} \subseteq A \times A$, then security attributes interconnection graph G is the directed graph G (A, R_{impact}), were A the set of vertices set and R_{impact} the set of edges.

C. SPD attributes interconnections derived from component interconnections

We define an *IMPACT* relation as interconnection relation between SPD attributes of components. This *IMPACT* relation is defined through $DEP_{control}$ and DEP_{data}, connecting the SPD attributes on the control and the data plane. Within these two sets of SPD attributes, the first element of the pair has an impact relation with second element, where:

- DEP_{data}, contains pairs which have an *IMPACT* relation resulted from data transmission.
- $DEP_{control}$ contains pairs which have an *IMPACT* relation resulted from control command transmission.

The formalized description of the relation is seen as follows:

 ISBN: 978-1-61208-369-8

In a given system, let C be a set of system components, A be a set of SPD attributes and *c be a component c ∈ C. The relations are then defined as:*

ATTB(c) = attb – return attb ∈ ATTB, being the SPD attribute set of component c.

CRTL(c) = crtl – return crtl ∈ CRTL, being a set of components, where c has direct or indirect control relation to.

DATA(c) = data – return data ∈ DATA, being a set of components, where c sends data directly or indirectly to. Then $DEP_{control}$ *and* DEP_{data} *are defined as:*

$DEP_{control}$ *={ (a1, a2) | a1, a2 ∈ A ∧ ∃ c1, c2 ∈C (c2 ∈ CRTL(c1) ∧ a1 ∈ ATTB(c1) ∧ a2 ∈ATTB(c2)*
⇒ a1 IMPACT a2*}*

DEP_{data}*= { (a1, a2) | (a1, a2 ∈ A) ∧ (c1, c2 ∈C) ∧ (c2 ∈ DATA(c1) ∧ a1 ∈ ATTB(c1) ∧ a2 ∈ATTB(c2)*
⇒ a2 IMPACT a1 *}*

Note: While the interconnection relation between components exchanging data is in forward direction, *a1 IMPACT a2*, the *IMPACT* relation is the other way around for components exchanging control commands. Let us consider an example, where an attacker exploits the connection component confidentiality (SPD attributes) in the communication between a sensor and a control unit. By exploiting this connection component, the privacy of the sensor component sending data is compromised. Likewise, exploiting the confidentiality of the connection component transmitting commands reveals the privacy of the control unit.

V. CASE STUDY

This case study investigates the impact of authentication attributes of a remote turn on/off component on the availability attribute of other related components in a smart vehicle. The case is based on the extended SHIELD methodology presented in the previous sessions.

Let us consider that the vehicle owner could turn the vehicle on/off using a mechanical component and remotely by a software application. Let us further consider that a remote turn on/off will disable the mechanical turn on/off.

As a first step, the related components of the engine are identified, based on control command transmission and data transmission. The components might be described as follows:

- C1: software component on mobile phone to remotely turn the engine on/off by the owner.
- C2: software connection transmitting the control command from C1 to C3.
- C3: actuator component responsible for turning the vehicle on/off (C4) and also responsible for deactivating of mechanical turn on/off from C5.
- C4: vehicle engine
- C5: mechanical component (key, button) for turning the engine on/off.

An exploitation of the C1 authentication will impact the availability of C4 and C5. An attacker exploiting the C1 authentication will put him into control of the engine and also disable the mechanical turn on/off component C5. But, at the same time, exploiting the C1 authentication will not impact the connection availability (C2) or the actuator availability (C3).

The extended SHIELD methodology for this use case is shown in Figure 4, with the authentication attribute being represented by circle.

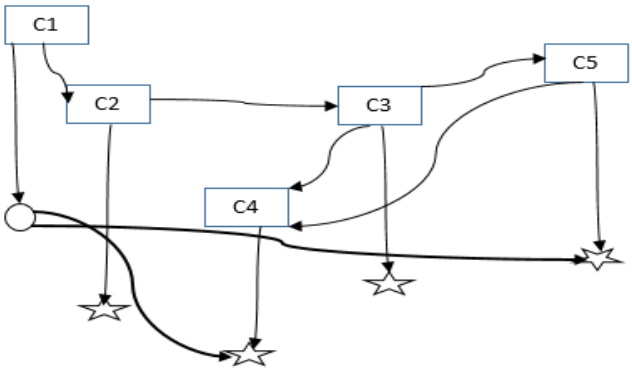

Figure 4. Case representation of the extended SHIELD methodology, exploiting the authentication (circle) on availability of other SPD components.

VI. SPD FUNCTIONALITIES COMPOSING ALGORITHM

The previous section indicated the dependency between components based on their interactions with each other. This section will demonstrate how system component SPD attributes, such as confidentiality, can be composed through SPD functionalities, such as encryption. The composition process using our methodology will also be able to counteract on limited safety and security resources, and will help handle the increasing complexity of systems.

The goal of the risk analyzer is reaching optimal composition of SPD system functionalities. An optimal composition of SPD functionalities has many advantages including:

- Increase system SPD level (through adding of SPD functionalities to most needed places in the system);
- Reduce security cost (through removing of SPD functionalities from less needed places);
- Improve the performance of the system (through removing of SPD functionalities from less needed places).

In our extended methodology, to optimize the composition of SPD functionalities, we weight each component in the system based on its interconnection within the system. Component interconnections will reflect to which degree the failure of component SPD attributes will impact the SPD level of the system. From this perspective, our component weighting algorithm depends on the following factors:

- The number of interconnection relations with surrounding components.
  Here surrounded components increase the weight of a component through direct interconnections. The impact of exploiting the component SPD attributes on the system SPD level increases with the number of direct interconnected components.

 ISBN: 978-1-61208-369-8

- The number of components reachable from weighted components.
  This factor reflects how many components are reachable through data or control relations from/to this component.
- The number of components between a weighted component and a main valuable component, called key component.
  By a key component, we characterize a component, in which exploiting of SPD attributes will cause a significant reduction of the system SPD level. The identification of key components will be a task for the domain specialist. Our assumption is that a close logical distance to a key component makes the system more vulnerable, letting us consider the logical distance between components.
- Type of relation between a component and other components.
  We identified 'data' and 'control' being the two types of interconnections between system components. In our methodology, we consider control being more vulnerable than data relations, as exploiting data is often related to monitoring, while exploiting control is opening for changes in the system. The worst case exploiting of control components could cause serious accidents (in our vehicle scenario).
- Component activation rate:
  Not all components are active during time of operation. For instance, airbags and eCall applications will only be triggered in a crash situation. In normal operations, these crash components have no impact on the SPD level of the system. Based on this status of operation our methodology considers the component activation rate for the weighting of SPD components.

Using the factors mentioned in the previous list, we propose a weighting algorithm, taking into account the impact of an exploitation of SPD attributes.

Let C be the component set of a given system, and
V be the subset of system Valuable components (key component) included in C, $V \subseteq C$.
With 'c' being a component in the system, $c \in C$,
F being the set of *SPD* functionalities, and
A being the set of *SPD* attributes, where 'a' is an *SPD* attribute, $a \in A$, the following functions are defined to introduced weighted relations:

$ATTB(c) = attb$ – *return* $attb \in ATTB$, are *SPD* attributes set of component c.

$CRTL(c) = crtl$ – *return* $crtl \in CRTL$, are sets of components, with c having direct or indirect control relation with the component.

$DATA(c) = data$ – *return* $data \in DATA$ is a set of components, where c sends data directly or indirectly to the component.

$SPD_{Func}(a) = SPD_{func}$ – *return* $SPD_{func} \in F$ is a set of SPD functions, where the SPD_{func} satisfies 'a'.

$DIST(c1, c2) = n$ – *return* is the number of components between $c1$ and $c2$

$NUM(S) = n$ – *return* is the number of elements in S, where S is any set of components (this function just counts the number of set element)

If $c2 \in CRTL(c1) \vee c2 \in DATA(c1) \wedge DIST(c1, c2)$ *is 1* $\Rightarrow c2$ *Surrounded* $c1$

$Surrounded(c) = s$ – *return* $s \in C$ is the set of components surrounding c.

If $c2 \in CRTL(c1) \vee c2 \in DATA(c1) \Rightarrow c1$ *Reach* $c2$

$Reach(c) = R$ – *return* R is the set of components reachable from c, where $R \subseteq C \wedge \forall r \in R$, c *Reach* r.

ActivationRate(c) is the percentage of 'c' activation comparable for longest activated component.

$VAL(c) = v$ - *return* v where $v \in V \wedge v \in Reach(c)$, is a set of valuable component reachable from c. Using above functions component weight is calculated as

$$Weight(c) = NUM(Surrounded(c)) + NUM(Reach(c)) + NUM(CRTL(c))*2 + NUM(DATA(c)) + AtivationRate(c) + \left(\sum_{v=0}^{n} \frac{1}{DIST(c,v)}\right) * NUM\big(VAL(c)\big), where\ v \in VAL(c)$$

Let $c1, c2 \in C$, *if* $weight(c1) > weight(c2) \wedge \exists a1 \in ATTB(c1) \wedge a2 \in ATTB(c2)$ *then* the priority of *composing* $SPD_{Func}(a1)$ is higher than (>) *composing* $SPD_{Func}(a2)$

The above algorithm calculates the weight of components based on the degree of interconnection with other components in the system. Given an example, the exploitation of SPD attributes of one component being in the close neighborhood of a key component will reduce the SPD level of the system considerably, expressed through the *Reach* and *Weight* functions.

Another example addresses the *Reach* condition of a component, given the comparison of A and B being components within a given system, where:

- Within the B component, data are processed without transmission of parameters to other components.
- The component A receives data from components C, D and G.

Exploiting the confidentiality of component B will only impact B, whereas exploiting the confidentiality of component A will impact the confidentiality of components C, D and G. Our extended methodology uses the *Reach* function, which will provide component B with less weight than component A.

Ongoing work applies the weighting algorithms to the calculation of impact of SPD attributes in a system analysis of

 ISBN: 978-1-61208-369-8

the vehicle use case. We expect that the outcome of the study will enable us to draw relational security, privacy and dependability graphs of embedded systems, and thus better tailor the complexity in a system of systems.

VII. CONCLUSION

This paper provided measures for the interaction between a variety of components or parties in the Internet of People, Things and Services (IoPTS). Interactions between components in a system cause security-related challenges for embedded systems. The paper uses the Security, Privacy and Dependability (SPD) approach developed by the SHIELD Methodology of the JU Artemis. We proposed an extension of the methodology to compose security and safety techniques, taking into consideration the interconnection between system components. The overall, system SPD level will thus become more dependent on the impact of interconnection between system components.

For each component, the impact of exploiting its SPD attributes on the system SPD level is represented by a *Weight* function. Our methodology uses both neighborhood and distance to key components as measures for the *Weight* function. The enhanced SHIELD methodology enables system engineers to compose SPD levels, and, by that, increases the SPD level of embedded systems.

Our methodology opens for a graph representation of security, privacy and dependability between components of embedded systems, and thus, visualizes critical paths.

ACKNOWLEDGMENTS

The authors would like to thank their colleagues from the ARTEMIS project, nSHIELD for the basics of the methodology, and the ongoing discussions on applicability.

Authors would like also to thank colleagues from ASSEST project for scientific corporation. The work is financed in part by the JU Artemis and the Research Council of Norway.

REFERENCES

[1] J.H.P. Eloff, M.M. Eloff, M.T. Dlamini, and M.P. Zielinski. “Internet of people, things and services-the convergence of security, trust and privacy”, 2009, Proce. of the third International Workshop IoPTS, Brussels, Dec 2009.

[2] L. Atzori, A. Iera and G. Morabito, “The Internet of Things”, A survey, Comput. Netw. 2010, doi:10.1016/ j.comnet.2010.05.010.

[3] pSHIELD, the pilot SHIELD project, and nSHIELD, the new SHIELD project, are EU Artemis projects in the security domain, Available from http://newshield.eu/ 2014.08.25

[4] B. Schneier, “Attack Trees,” Dr. Dobb’s J., Dec. 1999.

[5] R. Dantu, K. Loper and P. Kolan, “Risk Management Using Behavior Based Attack Graphs,” Proc. Int’l Conf. Information Technology: Coding and Computing, 2004, pp. 445-449.

[6] S. Noel, S. Jajodia, B. O’Berry and M. Jacobs, “Efficient Minimum-Cost Network Hardening via Exploit Interconnection Graphs,” Proc. 19th Ann. Computer Security Applications Conference 2003, pp. 86-95.

[7] P. Ammann, D. Wijesekera and S. Kaushik, “Scalable, Graph- Based Network Vulnerability Analysis,” Proc. Ninth Conf. Computer and Communication Security, 2002, pp. 217-224.

[8] C. Phillips and L.P. Swiler, “A Graph-Based System for Network-Vulnerability Analysis,” Proc. New Security Paradigms Workshop, 1998, pp. 71-79.

[9] S. Jha, O. Sheyner and J.M. Wing, “Two Formal Analysis of Attack Graphs,” Proc. 15th IEEE Computer Security Foundations Workshop, 2002, pp. 49-63.

[10] J. Dawkins, C. Campbell and J. Hale, “Modeling Network Attacks: Extending the Attack Tree Paradigm,” Proc. Workshop Statistical Machine Learning Techniques in Computer Intrusion Detection, 2002.

[11] L. Wang, S. Noel and S. Jajodia, Minimum-cost network hardening using attack graphs, Computer Communications 29(18) 2006, 3812–3824.

[12] L. Wang, A. Singhal and S. Jajodia, “Measuring the Overall Security of Network Configurations Using Attack Graphs,” Proc. 21st Ann. IFIP WG 11.3 Working Conf. Data and Application Security, 2007, pp. 98-112.

[13] L. Wang, T. Islam, T. Long, A. Singhal and S. Jajodia, “An Attack Graph-Based Probabilistic Security Metric,” Proc. 22nd Ann. IFIP WG 11.3 Working Conf. Data and Applications Security, 2008,pp. 283-296.

[14] P. Xie, J.H. Li, X. Ou, P. Liu and R. Levy, “Using Bayesian Networks for Cyber Security Analysis,” Proc. 40th IEEE/IFIP Int. Conf. Dependable Systems and Networks, 2010.

[15] H. Gomaa and D.A. Menascé. “Design and performance modeling of component interconnection patterns for distributed software architectures”. In Proceedings of the 2nd Int. workshop on Software and Performance, Sep 2000, pp. 117-126. ACM.

[16] F.V. Jensen, “An introduction to Bayesian networks”. Vol. 210. London: UCL press, 1996.

 ISBN: 978-1-61208-369-8