Adaptive Risk Management with Ontology Linked Evidential Statistics and SDN

Arto Juhola
VTT Technical Research Centre of Finland
Vuorimiehentie 3
02044 VTT
arto.juhola@vtt.fi

Titta Ahola
VTT Technical Research Centre of Finland
Vuorimiehentie 3
02044 VTT
titta.ahola@vtt.fi

Kimmo Ahola
VTT Technical Research Centre of Finland
Vuorimiehentie 3
02044 VTT
kimmo.ahola@vtt.fi

ABSTRACT
New technologies have increased the dynamism of distributed systems; advances such as Software Defined Networking (SDN) and cloud computing enable unprecedented service flexibility and scalability. By their nature, they are in a constant state of flux, presenting tough challenges for system security. Here an adaptive – in real time - risk management system capable of keeping abreast of these developments is considered. This paper presents an on-going work on combining a hierarchical threat ontology, real-time risk analysis, and SDN to an efficient whole. The main contribution of this paper is on finding the suitable architectures, components, necessary requirements, and favorable modifications on the systems and system modelling (including the models involving the security analysis) to reach this goal.

Categories and Subject Descriptors
C.2.0 [Computer-Communication Networks]: General - Security and protection

K.6.5 [Management of Computing and information Systems] Security and Protection

General Terms
Information Security, Risk Management

Keywords
Neural Network inspired Fuzzy C-means, Dempster-Schafer, Dezert-Smarandache, SDN, Adaptive security, Threat ontology

1. INTRODUCTION
The estimation of the real-time risk levels of a running system poses many challenges. The calculation of the risks is inherently statistical in nature, but traditional aka "frequentist" methods cannot be effectively applied due to the often insufficient or lacking pre-existing data. In addition, even if such data should have been accumulated for given operational circumstances, changes in the system configuration will render this data misleading.

Such changes are ever more common and frequent; up-and-coming technologies like cloud computing and SDN have blurred the borderlines of the system and introduced increased dynamism related to the system configuration.

All this leads to the need to tie the real-time risk analysis not only to the feedback from IDS/IPS but to the automatic system configuration changes, that may or may not be security related.

In this paper, we describe our novel idea on combining a hierarchical threat ontology, real-time risk analysis, and SDN into the adaptive real-time risk management system. It is a part of the on-going work in the CELTIC project SASER/SIEGFRIED Safe and Secure European Routing – Security In EnerGy-efficient Flexible and ResilIEnt Data networks [26]. The work in question is aiming to come up with a proof-of-concept realization of a highly dynamic risk management scheme. This involves monitoring of security aspects of a system and statistical analysis providing probabilities of "atomic" threats, that is, threats that belong to the terminal nodes of a hierarchical threat ontology tree. To arrive at higher level probability values linked with the next level ontology tree nodes, algorithms for calculation of evidential probabilities are (iteratively) used. Such include Bayesian, Dempster-Schafer and/or Dezert-Smarandache theories.

A related functionality concerns the decision upon the actions to be launched given the passing of predetermined quantitative/qualitative risk levels. This belongs in the domain of SDN, and a consequence is that via the action(s) the risk levels will be alleviated. Thus, there needs to be feedback to the risk determination process concerning the enforcement of the remedies.

A side-effect of those actions will be that the expected behavior, and subsequently the statistical measures of the system will be altered. This leads to the need to adjust the analysis activities to accommodate the change. Such dynamic considerations and the requirements placed on the system's risk management will be the subject matter here.

The rest of this paper is organized as follows. Section 2 introduces the background of technologies utilized in our research. Section 3 presents our approach, the testbed and its main functional components. Thereafter, section 4 describes the operations required to refine the detected anomalies to probabilities, and finally to risks. Related work is discussed in section 5. Finally, section 6 concludes the paper and discusses the future work.

2. BACKGROUND
Our approach combines together a hierarchical threat ontology and real-time risk management in the context of SDN. The basics of these concepts are discussed in next sections.

ECSAW, August 25 - 29 2014, Vienna, Austria
Copyright 2014 ACM 978-1-4503-2778-7/14/08…$15.00.
http://dx.doi.org/10.1145/2642803.2642805

2.1 Ontology

The term ontology has its roots in philosophy, and one of its usage areas was to model existence. In [10] Gruber defines ontology as an explicit specification of a conceptualization. Afterwards, Zhou [34] defines ontology as a shared knowledge standard or knowledge model defining primitive concepts, relations, rules and their instances, which comprise topic knowledge. It can be used for capturing, structuring and enlarging explicit and tacit topic knowledge across people, organizations, and computer and software systems. Thus, the defined ontology provides a shared vocabulary to model a problem domain, its objects and their relations. In our case, the ontology is used to describe identified threats and their relations.

2.2 Risk Driven Security Management

The aim of Risk Driven Security Management is straightforward: to focus on the high-impact threats first in the event of limited means.

ISO/IEC 27005:2008 [10] defines risk to the potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization. It is measured in terms of a combination of the probability of occurrence of an event and its consequence.

2.3 SDN

SDN – Software Defined Networking – is a networking paradigm that provides a software abstraction layer on top of the physical network infrastructure, enabling control plane functions distinct from the data plane [21]. The interface that is used in the communication between the control software and the forwarding elements is usually OpenFlow [18] (version 1.0 in our case). The OpenFlow abstracts the low-level forwarding behavior of each forwarding element, and enables the definition of a rule for each flow; if a packet matches a rule, the corresponding actions (e.g. drop, forward, modify) are performed. The high-level view of SDN architecture is depicted in Figure 1.

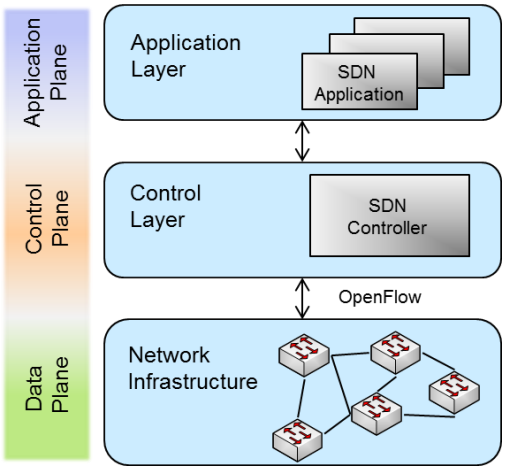

Figure 1. The SDN Architecture.

Control plane offers a global view of the network to the SDN-specific applications on top of the SDN framework. The OpenFlow protocol allows applications to request numerous statistics from the network device, such as counters for each flow, table, port, queue, group, and bucket. These properties provide information about the network performance, and can be collected to draw a better picture concerning network state. In addition to network state, those statistics can be used for anomaly detection, like "normally this time the traffic is only z Mbit/s, but now it is 10 * z Mbit/s".

3. THE APPROACH

The means of risk adaptation will be studied with an experimental set-up to be readied. The research is case driven; the misuse case(s) [30] is yet to be selected. The risk management concept consists of four main functionalities: data fusion, anomaly detection, system management, and SDN. Figure 1 presents these functionalities as the building blocks of the envisioned risk management testbed set-up.

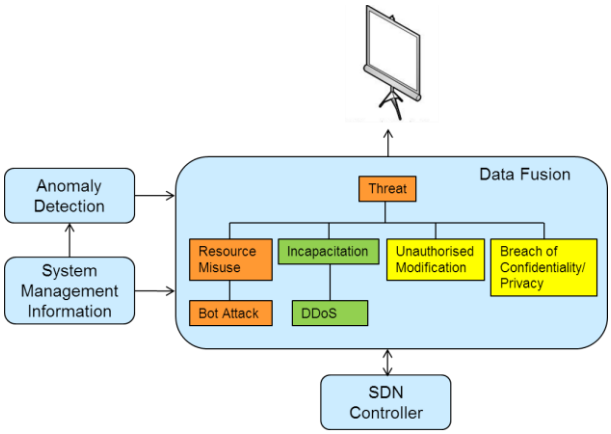

Figure 2. Risk management testbed set-up.

The Data Fusion is responsible for maintaining risk estimations associated with an ontological hierarchy of threats. The "atomic" terminal nodes of the threat ontology are to be readily associable with corresponding occurrence probabilities of anomalies, derivable from prior knowledge, system's anomaly, and state reports.

The calculation of occurrence probabilities of the non-atomic nodes of the threat ontology are performed using methods developed for combination of evidence, particularly Dempster-Schafer and Dezert-Smarandache theories. This reflects the increasing abstraction level of the identified threats when traversing the tree upwards, creating successively condensed views on the system risk state.

Anomaly detection functionality comprises the search of traffic and host related anomalies. The search is performed with existing Network/Host-based Intrusion Detection System technology, using their pre-processing capabilities as far as they permit. Moreover, two prototypes, MitM and LHT, created in the SASER project [26] are also integrated into the testbed. The MitM prototype is a solution for detecting man-in-the-middle attacks using the timestamp of TCP packet header [33]. By calculating the delay of sequential packets, and comparing the mean of the delays in the current session to reference data from old sessions, the unusually long delays can be detected, and thus suspect a MitM-attack. The other prototype, the LHT – Link History Tree records network's behavior, and enables the administrator to use the past behavior to assess nodes' future behavior, and to realize unusual traffic and connections.

In addition to anomalies, the normal system and network management affects system's risk level. System management data including system state and configuration, faults, and network configuration are factors to be taken into account in real-time risk calculations.

When an anomaly is detected or a change in the system is noted or enforced, the information of the change is forwarded to the Data Fusion. As a consequence of the change, the pre-existing risk analysis is no more applicable for the current situation. Therefore, the responsibility of the Data Fusion component is to update the threat ontology tree of a system.

In our case, the risk management system's (real-time) responsibilities do not end there. Active and real-time countermeasures can be initiated for alleviating revealed risks. These are launched using SDN, and they involve operations like redirecting suspect traffic to a special system segment for a closer look. The performing of these actions need to be notified to the risk calculation components.

In practice, SDN Controller pictured in Figure 1 consists of three parts: SDN application, SDN Controller framework, and the actual switching/routing device for data plane. The SDN application provides traffic management services to other applications through Representational State Transfer – REST – interface [7]. The idea is that it offers simple services such as steering or dropping traffic, and the other applications do not need to know anything about underlying network. The first version of the SDN application is ready

Several open source software platforms are available to assist in building an OpenFlow controller. These platforms hide the low-level functions of OpenFlow and the communication with the network devices. Our choice for this testbed was Ryu component-based software defined networking framework [24] implemented with Python programming language. It supports various protocols for managing network devices, such as Netconf [3], OF-Config [21], and OpenFlow (versions 1.0, 1.2, 1.3, 1.4) [19], and offers sample components which the programmer can use as a basis for building more complicated controller functions.

The MikroTik RB2011 [18] multi port device was selected for data plane, because it is modular and low-cost. The MikroTik is powered by RouterOS [19], a specific operating system of MikroTik for RouterBOARD hardware. The OpenFlow support for MikroTik device is available with standalone packet.

With the above arrangement, it will be possible to study the interdependencies and interplay manifested by a system with adaptive and active risk management.

4. DESCRIPTION OF THE OPERATION

4.1 Detection of Anomalies

The traffic/host related anomalies will be found using usual NIDS/HIDS, and SASER specific prototypes. The anomalies pertinent to the selected cases and requiring combination of samples from diverse sources are searched with a Neural Network inspired clustering algorithm variant.

4.2 Clustering algorithm used

There are several clustering algorithms with varying characteristics and preferred use cases. The results of many are tricky to interpret, and since the emphasis here is in the dynamic aspects of risk management, a search was made for an alternative that would require as little in the way of "fine tuning" and interpretation as possible, while yet providing usable information.

An algorithm that has reportedly [13][28] many favorable features was chosen – A "Neural network" style variation of "Fuzzy C-means". This was implemented with some further modifications. The main difference with the original Fuzzy C-means scheme is that instead of supplying the (hopefully exact) number of clusters, obtained by some other means prior calculation, the number of initial cluster centers will be set as "large enough". In fact you could say that this is just another way of using Fuzzy C-means.

The large number of cluster centers causes Fuzzy C-means to resemble a SOM – Self organizing map. Below we give a 3D example of the power of this algorithm (Figure 3 and Figure 4), together with a 2D multidimensional scaling – MDS – representation. The idea of MDS is to retain relative distances between points in space while stripping away dimensions. With it, the "entropy" of the data can be estimated.

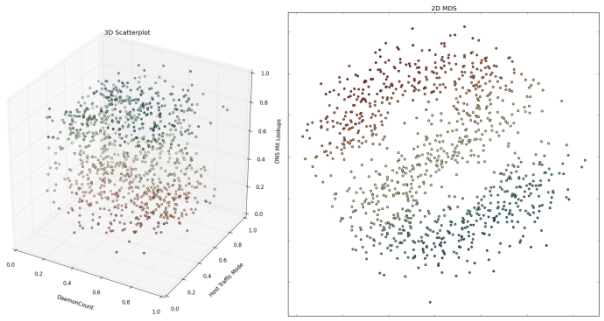

Figure 3. Noisy 3D "S" and 2D MDS figure of it.

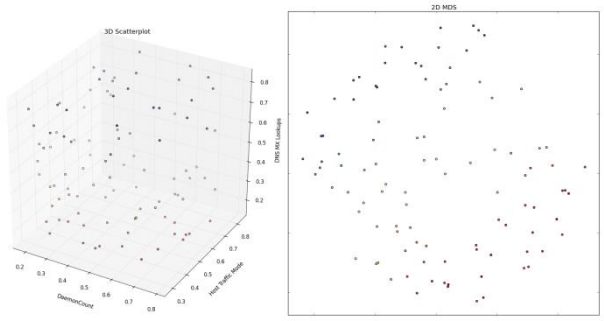

Figure 4. Noisy "S" through Neural NW likeFuzzy C-means and 2D MDS of it.

4.3 From Anomalies to Probabilities

The anomalies need to be converted to probabilities of threats. At first, the human expertise is the only recourse, in the lack of history data. When statistics based on history are available, either from the monitored system or from external sources, the initial values can be corrected. It is conceivable that some automatic means can be used.

We have:

- Some initial idea of the probability of an event (triggering a threat), say 1%. Let us label it as A.
- Another event, say an alert / notice from the system. Let the probability of this be X.

Next, we need to assign some probabilities on the correctness of the alerts, say, some alert is correct (not a false positive). Using Bayes' Theorem:

$$p(A|X) = \frac{p(X|A)p(A)}{p(X|{\sim}A)p({\sim}A)}$$

$\rho(A|X) =$ The probability of event A, given a fulfilment of an additional probability (evidence, like an alarm) X.

In short, this gives a probability of an event given another event (the alarm) X is true/has happened, if we know/supply beliefs:

- $\rho(A), \rho(\sim A) = 1 - \rho(A)$
- The probability that the alarm is sent in case of an event, $\rho(X|A)$
- The probability of the false positives, $\rho(X|\sim A)$.

In Bayesian spirit, these left-hand side probabilities can start their lives as beliefs, i.e. as subjective estimates, to be adjusted whenever more data for their calculation accumulates.

What we have achieved so far is an alarm/notification with a probability of its correctness attached.

4.4 From Probabilities to Risks

The approach taken in SASER/Siegfried is to create an ontological hierarchy of threats, that is, the risks tied to threats higher in the hierarchy are derived from the risks related to one or more threats beneath them. These lower level threats are arrived at by decomposing the higher level threats.

So, for the evaluation of the risks we need the probability of the triggering event, its mapping to the threat ontology (i.e. the probability that a threat will actualise), and the resulting impact.

This impact can be monetary losses or other harm, including side-effects (actualisation of higher level threats).

4.5 Combining Probabilities

The threat ontology tree presents a problem for the calculation of associated risks. There is no straightforward means of combining the lower level risks to the higher level, more abstract risk. The following paragraphs give an account of the known alternatives with an estimation of their applicability.

4.5.1 Bayesian Network

Bayesian Belief Network [1] – BBN – is a popular method for evaluating related probabilities that can be presented as a directed acyclic graph – DAG. There are implementations available, and at the moment the PyMC [7] is a likely candidate for experimentation. PyMC is capable of Bayesian learning tasks, so it can be thought to be overkill for simpler causal Bayesian NWs, but it is easy to set up.

The computation of a Bayes NW with PyMC is principally performed with "Markov Chain-Monte Carlo" – MCMC – algorithm [1], applied on distributions of probabilities rather than deterministic probability values.

That is, the PyMC solves the Bayesian theorem in a form written with *probability distributions* ($\rho(\theta), \rho(X)$), which means using integrals instead of sums [1] :

$$\rho(\theta|x) = \frac{p(\theta)\rho(x|\theta)}{\int \rho(\theta)p(X|\theta)d(\theta)}$$

There is no practical way to solve these integrals analytically (The evidence consists of several components), so computational alternatives like MCMC are a necessity.

Verdict: The match of PyMC with the problem at hand is not perfect; we are dealing with straight probabilities instead of probability densities. The application of PyMC is thinkable but likely to be taxing on computational resources.

4.5.2 Naïve Bayes

Another alternative, simple combining of component probabilities, not as versatile as PyMC, an implementation is available for Python [32]. Naïve Bayes variations are widely used in spam filtering due to good enough judgment of messages, in spite of not fulfilling strictest demands of theoretical purity (like taking in account covariance between pieces of evidence). The calculation involves only basic operations, as seen in the formula below:

$$P(A|x_1 \dots x_n) = \frac{P(A)\prod_{i=1}^{n} P(x_i|A)}{P(A)\prod_{i=1}^{n} P(x_i|A) + P(\sim A)\prod_{i=1}^{n} Px_i|\sim A)}$$

A = Prior probability for an event (here, realization of a threat)

x_i = Evidence, i[th] of n pieces

$\prod_{i=1}^{n}[Px_i|A)$ = Probability of the "true positive" evidence

$\prod_{i=1}^{n}[Px_i|\sim A)$ = Probability of the "false positive" evidence

The denominator can also be $\prod_{i=1}^{n} P(x_i)$ if these are known.

Verdict: Naïve Bayes is simple, has a good match with the problem but is not exact nor comprehensive (does not cater for cases other than combining non-conflicting and exclusive probabilities).

4.5.3 Dempster-Shafer Theory

Dempster-Shafer - DS – theory, together with the Dempster's rule of combination, is a popular Bayes generalization for evaluating evidence [24]. It has been proposed to the risk evaluation of information systems [29].

DS assigns discrete degrees of belief ([0-1], "m-values") to the evidence supporting a hypothesis. That is, "0" signifies that the evidence does not remove uncertainty (surrounding the correctness of the hypothesis), and "1" is for certainty (the evidence proves the hypothesis).

With Bayes, singular evidences were under study, with DS multiple pieces of evidence are arranged into (sub)sets, including "any" and "null". Like in probability mass distributions, they sum to 1.

Also, in contrast to Bayesian thinking, the m-values of singular evidence and its negation need not to add up to 1. If they e.g. are both 0, the significance is that this piece of evidence (and its negation) is either for or against the hypothesis.

From the m-values, belief functions are formed (Bel(x)). They simply sum up the m-value of the hypothesis and all m-values of all subsets where the hypothesis has support.

The "Plausibility" is the last main term in the theory, defined as 1 – bel(~x), where ~x = belief in the hypotheses not supporting the one in question. The belief value is smaller or equal to the Plausibility, so it gives the lower and the Plausibility the upper bound for the truthfulness of the hypothesis.

Finally for the combination of the beliefs, Dempster's rule can be applied.

There is an implementation for Python – pyds - [21].

Verdict: DS has a good match with the problem, but one must be wary of threat ontology trees having a lot of branches from their nodes (=a lot of evidence subsets).

4.5.4 Dezert-Smarandache Theory

This is a recent extension to the Dempster-Schafer theory, with better support for combination of conflicting evidence [2].

There exists at least a matlab implementation that could be a starting point for experimentation [14].

4.5.5 Final Verdict

Neither DS nor DSmT uses integration in solving the combination problem, but with the increasing number of evidence the number of possible set combinations grows rapidly.

However, if the ontology tree can be made to have only few branches per a given node, i.e. tallish rather than short and wide, they are a tempting alternative. This gives better opportunity to employ parallelism as well.

Given that DSmT reduces to DS when conflicting evidence is omitted, DSmT is the one to be tried first. On the other hand, the Naïve Bayes is the likely choice if performance becomes a serious limiting factor.

4.6 The Effect of the System Management on Risks

Due to system management actions, the risk landscape is likely to change, including beliefs and prior estimates. Such actions include e.g. fault tolerance and security countermeasures. Therefore, such changes of state are to be used as input to the real-time risk analysis.

Additionally, there is an impact on the statistical operation parameters and models on which the anomaly detection bases its judgments. To remain usable, these should reflect the "new normal" state of the system.

There are several possibilities in adapting anomaly detection parameters and models to the system state changes. The simplest is ignoring or diminishing of the sensitivity of the detected anomalies for the duration of the short lived and atypical system states arising from known causes.

Second, the effect on parameters and models could be deductible, affording the ability to adjust them accordingly. This could work for well-known cases.

4.7 Enforcement of System Management Actions

The usage of SDN in our risk analysis environment is two-fold: it manages flows according to applications' requests and creates a feedback loop into the system. That is, when the SDN application receives a notification of the existence of an anomaly, it can decide to initiate some countermeasure (the Data Fusion component in the Figure 1 does not make such decisions). When these countermeasures are launched, the notification to the Data Fusion is also made. This in turns leads to re-evaluation of risks, and to further input to the SDN controller. This feedback loop cannot continue definitely, thus some thought needs to be given for eliminating possible instability.

5. RELATED WORK

This paper concentrates on the probabilistic and dynamic aspects of risk management, but the overall picture is wider than this. The initial probabilities of threats are derived based on security models on the system, using modelling languages like UMLsec [14]. Modelling frameworks using such languages have been developed, and in a paper of Kearney the dynamic aspects of risk management are discussed [15].

The Bayes network based probabilistic risk assessment and decision analysis has been treated by Norman Fenton and Martin Neil in their book [7]. Also DS – Dempster Schafer – and DSmT – Dezert Smarandache – theories have been applied in a security setting. In [10] Glowacka et al. presents a concept of Dezert-Smarandache theory application for enhancing security in tactical mobile ad-hoc networks, and [28] presents a developed circuit partitioning algorithm, namely Fuzzy Neural Computing.

The design and usage of security based and real-time ontologies are discussed papers from Evesti et al. In [5], the several existing security ontologies have been studied, and vision of the run-time security management is presented. Moreover, their follow-up paper [4] describes the Information Security Measuring Ontology (ISMO) that combines existing measuring and security ontologies particularly for the needs of run-time security measuring.

SDN solutions are emerging widely, and in addition to its traditional use, SDN offers new opportunities for many research fields. The SIGMONA project [29] is studying the utilization of SDN for traffic, resource, and mobility management, and network virtualization solutions in the mobile transport networks. Additionally, in [24] Ruponen presents the idea on how to utilize SDN in practice in wireless rural area networking.

However, the security and dependability of the SDN itself has still open issues. The main concerns lie in the SDN's main benefits: network programmability and centralization of the control. As studied in [16], SDNs bring a very fascinating dilemma: an extremely promising evolution of networking architectures, versus a dangerous increase in the threat surface. Identified main threats contain attacks on vulnerabilities in switches and in controllers, attacks on control plane communications, and lack of mechanisms to ensure trust between the controller and management applications.

6. CONCLUSIONS AND FUTURE WORK

This paper presented basic building blocks and design ideas for the real-time risk management framework.

The system monitoring and state information gathered from standard monitoring equipment is at a fairly high abstraction level, but to gain a view to the state of an entire distributed system an additional step was necessary. Given the dynamic environment, minimum of tuning and interpretation was a prerequisite. One of the clustering methods described in the literature seemed to fit the bill and was implemented. Afterwards, the experimentation with it has shown that it suits for our purpose.

The application of probabilities on system monitoring data is still subject to human expertise, and ways to allow for automatic correction of initial values is a possibility (based on history data).

Of the combination methods of the threat probabilities, the DSmT – Dezert Smarandache – is considered as the first implementation candidate because of the match with problem, the fact that it is essentially the same as DS – Dempster Schafer – when limiting to the case without conflicting evidence. The performance is hoped to be made acceptable by placing limitation on the count of branches of the threat ontology tree nodes.

First versions of anomaly detection components and SDN framework and its applications are integrated into the testbed. The first results of the testbed and its components seem promising, and indicate that on-going work is going to the right direction. As such, they don't provide all required functionalities to the framework, and thus, the architecture is likely to evolve while we are gaining more knowledge and exploring best options and practices.

The described experimentation will be carried out in the context of the CELTIC SASER, and the findings will be reported in future papers.

7. ACKNOWLEDGMENTS

Our thanks to the CELTIC and National funding agencies, particularly the Finnish Funding Agency for Technology and Innovation (Tekes), for making this work possible.

8. REFERENCES

[1] An Introduction to MCMC methods and Bayesian Statistics. 2012. Retrieved June 6, 2014, from Centre of Multilevel Modelling, University of Bristol: http://www.ccsr.ac.uk/esds/events/2012-07-11/mcmc.pdf

[2] Dezert, J. and Smarandache, F. An introduction to DSmT. Retrieved June 6, 2014, from Smarandache Notions Journal: http://www.gallup.unm.edu/~smarandache/IntroductionToDSmT.pdf

[3] Enns, R., Bjorklund, M., Schoenwaelder, J., and Bierman, A. 2011. Network Configuration Protocol (NETCONF). Internet Engineering Task Force , RFC 6241, June 2011.

[4] Evesti, A., Savola, R., Ovaska, E., and Kuusijärvi, J. 2011. The Design, Instantiation, and Usage of Information Security Measuring Ontology. In *The Second International Conference on Models and Ontology-based Design of Protocols, Architectures and Services* (Budapest, Hungary, 17-22 April, 2011), MOPAS 2011,1-9.

[5] Evesti, A., Ovaska, E., and Savola, R. 2009. From Security Modelling to Run-time Security Monitoring. *In the Proceedings of European Workshop on Security in Model-Driven Architecture* (Enschede, the Netherlands, 24 June, 2009), SECMDA, 33-41.

[6] Fenton, N. 2012. Probability Theory and Bayesian Belief Bayesian Networks. Retrieved June 6, 2014: http://www.eecs.qmul.ac.uk/~norman/BBNs/BBNs.htm

[7] Fenton, N. and Neil, M. 2012. *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, October 2012. Retrieved June 30, 2014: http://www.eecs.qmul.ac.uk/~norman/BBNs/BBNs.htm

[8] Fielding, R. T. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Chapter 5: Representational State Transfer (REST). University of California, Irvine.

[9] Fonnesbeck, C., Patil, A., Huard, D. and Salvatier, J. 2013. PyMC Documentation. Retrieved June 6, 2014: http://pymc-devs.github.io/pymc/README

[10] Glowacka, J. and Amanowicz, M. 2012. Application of Dezert-Smarandache theory for tactical MANET security enhancement. *In Communications and Information Systems Conference* (MCC), Military, 1-6.

[11] Gruber, T.R. Towards principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies, 43* (5/6), 907–928.

[12] ISO/IEC 27005:2008. Information technology - Security techniques - Information security risk management. ISO, Geneva.

[13] Hu, W., Xie, D., Tan, T., and Maybank, S. 2004. Learning activity patterns using fuzzy self-organizing neural network. *In IEEE Transactions Systems, Man, and Cybernetics, Part B: Cybernetics, 34* (3), 1618 – 1626.

[14] Jürjens, J. 2005. *Secure Systems Development with UML*, Springer.

[15] Kearney, P. and Brügger, L. 2007. A risk-driven security analysis method and modelling language. *In Technology Journal, 25*, (1), 141-153.

[16] Kreutz, D., Ramos, F. M., and Verissimo, P. 2013. Towards Secure and Dependable Software-Defined Networks. *In Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking*, HotSDN'13, 55-60. DOI= http://doi.acm.org/10.1145/2491185.2491199

[17] Martin, A. Matlab toolbox for a general belief functions framework. Retrieved June 10, 2014: http://martin.iutlan.univ-rennes1.fr/Doc/GeneralBeliefFunctionsFramework.tar

[18] MikroTik Routerboard. Retrieved June 13, 2014: http://routerboard.com/RB2011UiAS-RM

[19] MikroTik RouterOS. 2010. Retrieved June 27, 2014: http://www.mikrotik.com/pdf/what_is_routeros.pdf

[20] Open Networking Foundation (ONF). 2009-2014. Open Flow Specifications. Retrieved June 6, 2014: https://www.opennetworking.org/sdn-resources/onf-specifications

[21] Open Networking Foundation (ONF). 2014. OF-Config 1.2 OpenFlow Management and Configuration Protocol. Retrieved June 27, 2014: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf

[22] Open Networking Foundation (ONF). 2012. Software-Defined Networking: The New Norm for Networks. ONF White paper. April 13, 2012. Retrieved June 27, 2014: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[23] Reineking, T. A Python library for performing calculations in the Dempster-Shafer theory of evidence. Retrieved June 6, 2014 from pyds by Reineking : http://reineking.github.io/pyds/

[24] Ruponen, S. 2013. On Software-defined Networking for Rural Areas: Controlling Wireless Networks with OpenFlow. *In the Fifth International Conference on e-Infrastructure and e-Services for Developing Countries*, (Blantyre, Malawi, 25–27 November, 2013), AFRICOMM 2013.

[25] Ryu - Component-based software defined networking framework. 2014. Retrieved June 13, 2014 from Ryu SDN Framework Community: http://osrg.github.io/ryu/

[26] SASER/SIEGFRIED Safe and Secure European Routing – Security In EnerGy-efficient Flexible and ResilIEnt Data networks. Retrieved June 27, 2014: http://www.celticplus.eu/Projects/Celtic-Plus-Projects/2011/SASER/SASER-b-Siegfried/saser-b-default.asp

[27] Sentz, K. and Ferson, S. 2002. Combination of Evidence in Dempster-Shafer Theory. Technical report, SAND2002-0835, Sandia National Laboratories. Retrieved June 5, 2014: http://www.sandia.gov/epistemic/Reports/SAND2002-0835.pdf

[28] Shen, T., Gan, J. and Yao, L. 1992. Application Of Fuzzy Neural Computing For Partitioning Circuits. *In Custom Integrated Circuits Conference*, (Boston, MA, USA, 1992), IEEE, 5.3.1 - 5.3.4.

[29] SIGMONA - SDN Concept in Generalized Mobile Network Architectures. 2014. Retrieved 30 June, 2014: http://www.sigmona.org/

[30] Sindre, G. and Opdahl, A. 2005. Eliciting security requirements with misuse cases. *Requirements Engineering, 10* (1), 34-44. DOI=http://dx.doi.org/10.1007/s00766-004-0194-4

[31] Sun, L., Srivastava, R.P., and Mock, T.J. 2006. An Information Systems Security Risk Assessment Model Under the Dempster-Shafer Theory of Belief Functions. *Management Information Systems, 22* (4), 109-142.

[32] User guide: Naive Bayes, 2010 – 2013. Retrieved June 5, 2014, from scikit-learn developers: http://scikit-learn.org/stable/modules/naive_bayes.html

[33] Vallivaara, V., Sailio, M., and Halunen, K. Detecting Man-in-the-Middle Attacks on Non-Mobile Systems, *in Conference on Data and Application Security and Privacy*, (San Antonio, Texas, USA, 2014), ACM, 131-133.

[34] Zhou, J. Knowledge Dichotomy and Semantic Knowledge Management. 2005. *In Proceedings of the 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web*, (Jyväskylä, Finland, 25–27 August, 2005), Springer, 305–316.