Distributed attack prevention using Dempster-Shafer theory of evidence

Áine MacDermott, Qi Shi and Kashif Kifayat

School of Computer Science, Liverpool John Moores University, Liverpool, L3 3AF

a.mac-dermott@2008.ljmu.ac.uk and {q.shi; k.kifayat}@ljmu.ac.uk

Abstract. This paper details a robust collaborative intrusion detection methodology for detecting attacks within a Cloud federation. It is a proactive model and the responsibility for managing the elements of the Cloud is distributed among several monitoring nodes. Since there are a wide range of elements to manage, complexity grows proportionally with the size of the Cloud, so a suitable communication and monitoring hierarchy is adopted. Our architecture consists of four major entities: the Cloud Broker, the monitoring nodes, the local coordinator (Super Nodes), and the global coordinator (Command and Control server - C2). Utilising monitoring nodes into our architecture enhances the performance and response time, yet achieves higher accuracy and a broader spectrum of protection. For collaborative intrusion detection, we use the Dempster Shafer theory of evidence via the role of the Cloud Broker. Dempster Shafer executes as a main fusion node, with the role to collect and fuse the information provided by the monitors, taking the final decision regarding a possible attack.

Keywords: Intrusion detection; Cloud computing; Security; Collaboration; Dempster Shafer; Fusion Algorithm; Autonomous systems.

1 Introduction

Adoption of Cloud technologies allows critical infrastructure to benefit from dynamic resource allocation for managing unpredictable load peaks. Given the public awareness of critical infrastructure and their importance, there needs to be an assurance that these systems are built to function in a secure manner. Appropriate security procedures have to be selected when developing such systems and documented accordingly. Most existing technologies and methodologies for developing secure applications only explore security requirements in either critical infrastructure or Cloud Computing. Individual methodologies and techniques or standards may even only support a subset of specific critical infrastructure requirements. Requirement based security issues can be quite different for these applications and for common IT Cloud applications but need to be considered in combination for the given context.

adfa, p. 1, 2011.

Automation has become an indispensable part of service provision and has increased exponentially as demand for digital services and interconnectivity has increased. The reliance on these systems has resulted in ICT playing a key role in the provisioning of services that critical infrastructures deliver to the general population. Disruptions in one part of an infrastructure may propagate throughout the system and have cascading effects on other sectors (Ten, Manimaran, & Liu, 2010). Critical infrastructure protection relates to application processes, electronic systems, and information stored and processed by such systems.

The concern is that critical IT resources and information in Cloud systems may be vulnerable to cyber attacks or unauthorised access. The primary security concerns with Cloud environments pertain to security, availability, and performance. Many attacks are designed to block users from accessing services and providers from delivering services, i.e. Denial of Service (DoS) or Distributed Denial of Service (DDoS). Service providers may face significant penalties due to their inability to deliver services to customers in accordance with regulatory requirements and Service Level Agreements (SLA) (Rak et al., 2012). DDoS attacks are a serious and growing problem for corporate and government services conducting their business via the Internet. Resource management to prevent DDoS attacks is receiving attention, as the Infrastructure as a Service (IaaS) architecture, effectively 'supports' the attacker. When the Cloud system observes the high workload on the flooded service, it is likely the Cloud federation (which is the practice of interconnecting the Cloud computing environments of two or more service providers for the purpose of load balancing traffic and accommodating spikes in demand) will start providing more computational power in order to cope with it.

Traditional network monitoring schemes are not scalable to high speed networks such as Cloud networks, let alone Cloud federations. It is clear that an Intrusion Detection System (IDS) alone cannot protect the Cloud environment from attack. If an IDS is deployed in each Cloud Computing region, but without any cooperation and communication, it may easily suffer from single point of failure attack. The Cloud environment could not support services continually, as it is not always easy for the victim to determine that is being attacked, or where the attack is originating from. A new and novel approach to the aforementioned problem is required, that is, providing Security as a Service in a Cloud federation. Our solution encompasses the following methodological attributes:

- We represent Cloud Service Providers (CSPs) within a Cloud federation as interconnected domains.
- Once a Belief is generated that an attack is underway (b_a), this is sent to a Command and Control server (C2). The C2 queries the Cloud Broker, and the Broker checks the value against its stored values (as it may not have been published yet), and invokes a global poll procedure in which other C2s within the other domains are queried.
- The Cloud Broker coordinates attack responses, both within the domain itself, and with other domains, and is facilitating inter-domain cooperation.

  - Dempster-Shafer (D-S) is used to fuse the generated beliefs and make a system wide decision. This cooperation between CSPs ensures that the scalable defence required against DDoS attacks is in an efficient manner; aiming to improve the overall resilience of the interconnected infrastructure.
- The development of a collaborative intrusion detection heuristic based on D-S theory of evidence, and the inclusion of confidence values for improved accuracy.
  - We are improving the decision making precision and accuracy for autonomous information sharing in a federated Cloud environment via a two stage fusion process.

2 Background

The Cloud Computing paradigm is increasingly being adopted in critical sectors such as energy, transport, and finance. Deploying high assurance services in the Cloud increases cyber security concerns, as successful attacks could lead to outages of key services that have high socioeconomic implications. This exposes these infrastructures to cyber risks and results in demand for protection against cyber-attacks, even more than traditional systems. Critical security issues include: data integrity; user confidentiality; availability of data; and trust among entities. Securing applications and services provided in the Cloud against cyber attacks is hard to achieve due to the complexity, heterogeneity, and dynamic nature of such systems.

Site management and monitoring has improved for critical infrastructure facilities as they have become more progressively connected to the Internet. The added convenience of connectivity, however, has turned the once-limited attack surface of these industries into a fertile landscape for cyber-attacks. Due to the potentially high profile effects of attacks to critical infrastructure systems, these industries have become even more attractive targets for cybercriminals (Trend Micro Incorporated, 2015). The sensitive nature of critical infrastructure services deems their protection critical, and their services hereof. This is predominantly caused by the inadequacies and limitations of current security protection measures which fail to cope with the sheer size and vast dynamic nature of the Cloud environment.

Attacks and failures are inevitable; therefore, it is important to develop approaches to understand the Cloud environment under attack. The current lack of collaboration among different components within a cloud federation, or among different providers, for detection or prevention of attacks is the focus of our work (Á. MacDermott, Shi, & Kifayat, 2015). Our research focuses on maintaining the availability of the data, as previously described, the service in question could be financial, organisational, or on demand. Protecting the Cloud environment from DDoS attacks is imperative as these attacks can threaten the availability of Cloud functionalities.

3 Security as a Service

Detecting intrusion patterns in the Cloud environment involves looking for behavioural changes. This process could involve signature based detection for DoS attacks, which must be robust against noise data, and false positives and false negatives produced. Anomaly detection is an approach for detecting behavioural changes as these schemes often aggregate normal behaviour through their modelling of normal versus abnormal traffic. The main requirement of our solution is to provide protection for critical infrastructure services being hosted in the Cloud environment through novel intrusion detection techniques (Á; MacDermott, Shi, & Kifayat, 2015).

Monitoring is a core function of any integrated network and service management platform. Cloud computing makes monitoring an even more complex infrastructure support function, since it includes multiple physical and virtualized resources and because it spans several layers of the Cloud software stack, from IaaS to PaaS, and SaaS. Using our Security as a Service method, collaborative intrusion detection is possible in a federated Cloud environment.

The system uses a Cloud Broker to propagate information to the C2 entities in each CSP domain – this is in the form of Black lists and White lists. Monitoring nodes are used to observe changes or suspicious activities in local domains. These values are stored in a Grey list of ambiguous observations. For pre-emptive warning, Beliefs are generated and assigned to all subsets of possible outcomes based on the trigger.

We assign Beliefs to the outcome in the form of, random attack variable x would have a basic probability assignment of { }, {attack}, {no attack}, and {attack, no attack}. { } represents an empty subset with a value of 0, whereas {attack, no attack} represents uncertainty, i.e. it could be either. An advantage of using D-S theory of evidence to fuse Beliefs is that the algorithm can start from an uncertain state and allow the observed evidences form in each of the subnets gradually. D-S utilises orthogonal sum to combine the evidences. We define the belief functions, describing the belief in a hypothesis A, as $\mathrm{Bel}_1(\mathrm{A}), \mathrm{Bel}_2(\mathrm{A})$; then the belief function after the combination is defined as:

$$\mathrm{Bel}(\mathrm{A}) = \mathrm{Bel}_1(\mathrm{A}) \oplus \mathrm{Bel}_2(\mathrm{A})$$

The mass function after the combination can be described as:

$$\mathrm{m}(\mathrm{A}) = \mathrm{K}^{-1}. \sum_{\mathrm{A_i} \cap \mathrm{B_i} = \mathrm{A}} \mathrm{m}_1(\mathrm{A_i})\, \mathrm{m}_2(\mathrm{B_j})$$

Where K is called Orthogonal Coefficient, and it is defined as:

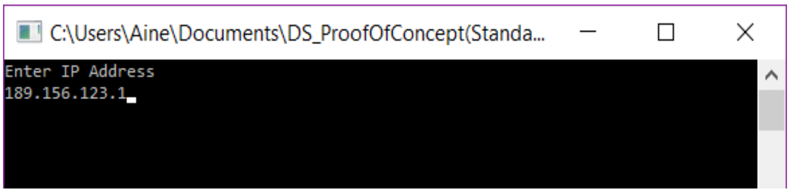
$$\mathrm{K} = \sum_{\mathrm{A_i} \cap \mathrm{B_i} \neq \emptyset} \mathrm{m}_1(\mathrm{A_i})\mathrm{m}_2(\mathrm{B_j})$$

3.1 Implementation

Collaborative security between CSPs in a Cloud federation can offer holistic security to those in this scheme. Information sharing in this approach is automated which we conceive to be an important aspect of our approach. Dividing the system into domains makes the system more scalable, and Belief generation and sharing of threat information could be used as a warning of an imminent attack. Previous work of ours (Á. MacDermott, Shi, & Kifayat, 2015) details our simulations using Riverbed Modeler 18.0 and convey how attacks could propagate throughout a Cloud federation. At this stage of the simulation, the main purposes were to analyse the role a Broker could have with autonomous sharing of information; the role of a single monitoring entity on the entire federation vs the C2s monitoring their own sub domains; and how an attack within a Cloud federation could affect the interdependent services present.

Next, we are showing the actions to be taken in the simulation, from the point where an intrusion is believed to have been detected. The integration of the decisions coming from different IDSs has emerged as a technique that could strengthen the final decision. Sensor fusion can be defined as the process of collecting information from multiple and possibly heterogeneous sources and combining them to obtain a more descriptive, intuitive and meaningful result (Thomas & Narayanaswamy, 2011). Related work in the field of sensor fusion has been carried out mainly with methods such as probability theory, evidence theory, voting fusion theory, fuzzy logic theory, or neural network in order to aggregate information.

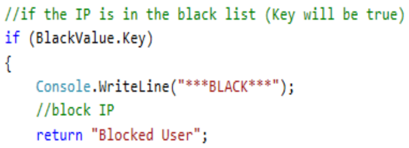
Our implementation of our D-S for collaborative intrusion detection is in C#, and focuses on demonstrating the application of the fusion algorithm in an autonomous information sharing scheme. For proof of concept we are using a lower amount of entities to convey how the communication occurs and the information would be exchanged within the infrastructure; future work would involve expanding this solution to cope with a larger scale. Firstly, an IP address is entered into the program and the value is compared to the Black list and White list to see if the values are present.



```
Enter IP Address
189.156.123.1
```

Fig. 1. Checking list values to determine if IP present

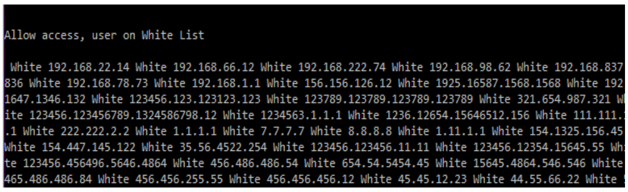
When compared against the lists, if the IP address is in the Black list then the user is ‘Blocked’ – source code for this is conveyed in Figure 2.



```
//if the IP is in the black list (Key will be true)
if (BlackValue.Key)
{
    Console.WriteLine("***BLACK***");
    //block IP
    return "Blocked User";
```

Fig. 2. Blocked user key return

If the entered IP address is present on the White list then the user is 'Permitted Access', as illustrated in Figure 3. The console outputs the other values from the White list, and this is also a separate file than can be viewed.

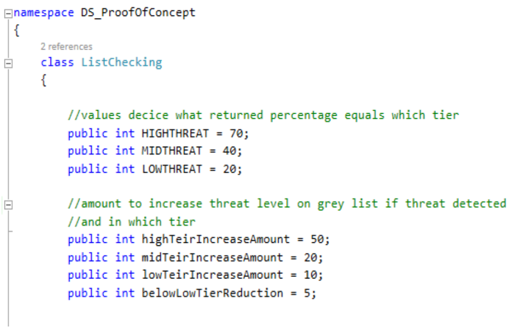


```
Allow access, user on White List

 White 192.168.22.14 White 192.168.66.12 White 192.168.222.74 White 192.168.98.62 White 192.168.837
836 White 192.168.78.73 White 192.168.1.1 White 156.156.126.12 White 1925.16587.1568.1568 White 192
1647.1346.132 White 123456.123.123123.123 White 123789.123789.123789.123789 White 321.654.987.321 Wh
ite 123456.123456789.1324586798.12 White 1234563.1.1.1 White 1236.12654.15646512.156 White 111.111.
.1 White 222.222.2.2 White 1.1.1.1 White 7.7.7.7 White 8.8.8.8 White 1.11.1.1 White 154.1325.156.45
White 154.447.145.122 White 35.56.4522.254 White 123456.123456.11.11 White 123456.12354.15645.55 Wh
te 123456.456496.5646.4864 White 456.486.486.54 White 654.54.5454.45 White 15645.4864.546.546 White
465.486.486.84 White 456.456.255.55 White 456.456.456.12 White 45.45.12.23 White 44.55.66.22 White
```

Fig. 3. Value on white list

If the entered IP address is not present on either list, the value is stored in the Grey list and assigned a threat value which we use to form the Belief. Hypothesis sets based on all values between 0 and 1 are included within the program, as well as mass values and plausibility functions.



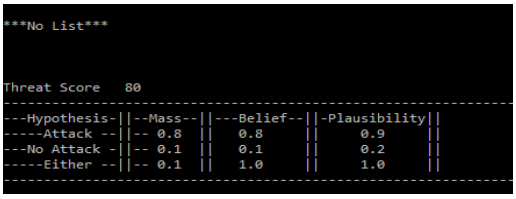
```
namespace DS_ProofOfConcept
{
    2 references
    class ListChecking
    {

        //values decice what returned percentage equals which tier
        public int HIGHTHREAT = 70;
        public int MIDTHREAT = 40;
        public int LOWTHREAT = 20;

        //amount to increase threat level on grey list if threat detected
        //and in which tier
        public int highTeirIncreaseAmount = 50;
        public int midTeirIncreaseAmount = 20;
        public int lowTeirIncreaseAmount = 10;
        public int belowLowTierReduction = 5;
```

Fig. 4. Threat value ranges

Figure 4 shows the threat value ranges used, and the ability to increase/decrease the associated risk due to occurrences on the list is also an option. Increased occurrences could cause the risk score to increase, e.g. beginning on the white list, moving to the grey list, but then being promoted to the black list. For a value over 70 this would trigger a Belief generation and the associated hypothesis values output. Figure 5 is an example of a threat score of 80 and the associated hypothesis set generated, and Figure 6 shows a score of 60.



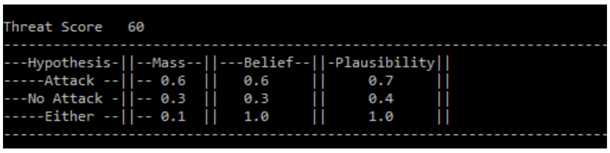
```
***No List***



Threat Score   80
---------------------------------------------------------------
---Hypothesis-||--Mass--||---Belief--||-Plausibility||
-----Attack --||-- 0.8  ||   0.8     ||     0.9     ||
---No Attack -||-- 0.1  ||   0.1     ||     0.2     ||
-----Either --||-- 0.1  ||   1.0     ||     1.0     ||
---------------------------------------------------------------
```

Fig. 5. Example hypothesis set generation for a threat score of 80

This value is sent to the Broker and compared against the Black and White lists, as the information may not have been propagated to the C2s within the federation. The Broker then queries the adjacent monitoring entities and requests they generate a Belief based on the original value.



```
Threat Score   60
---------------------------------------------------------------------
---Hypothesis-||--Mass--||---Belief--||-Plausibility||
-----Attack --||-- 0.6  ||   0.6     ||     0.7     ||
---No Attack -||-- 0.3  ||   0.3     ||     0.4     ||
-----Either --||-- 0.1  ||   1.0     ||     1.0     ||
---------------------------------------------------------------------
```

Fig. 6. A belief generation of 0.6

In the example of a Cloud federation, the Broker takes three belief values and fuses them together to make a system wide decision. The values would then be updated to the lists (White or Black) and calculations of these combinations are as follows:

Belief combination of two values – $\boldsymbol{Bel(A) = Bel_1(A) \oplus Bel_2(A)}$
(m_1) we have belief that the proposition is true for just state Attack is m_1({Attack}) = 0.8) and similarly m_1({No Attack}) = 0.1 with m_1({Either}) = 0.1

Then we take another assessment m_2 with m_2 ({Attack}) = 0.6, m_2({No Attack}) = 0.3 with m_2({Either}) = 0.1

The joint mass function would be $m_{1,2}$(A)=(1/1-K) m_1({Either})m_2({Either}) with
K= m_1({Attack}) m_2({No Attack}) + m_1({No Attack}) m_2({Attack}) = 0.8*0.3 + 0.1*0.6 = 0.30

So $m_{1,2}$(A) = (1/1-0.30) *0.1*0.1 = 0.007

Belief combination of three values - $\boldsymbol{Bel(A) = Bel_1(A) \oplus Bel_2(A) \oplus Bel_3(A)}$
(m_1) we have belief that the proposition is true for state Attack which is 0.3 (i.e. m_1({Attack}) = 0.8) and similarly m_1({No Attack}) = 0.1 with m_1({Either}) = 0.1

Then we take another assessment m_2 with m_2 ({Attack}) = 0.6, m_2({No Attack}) = 0.3 with m_2({Either}) = 0.1

m_3 associated values include {Attack}) = 0.0, m_3({No Attack}) = 0.5 with m_3({Either}) = 0.1

The joint mass function would be $m_{1,2,3}$(A) = (1/1-K) m_1({Either})m_2({Either}) m_3({Either}) with:
K = m_1({Attack}) m_2({No Attack}) + m_1({No Attack}) m_2({Attack}) + m_1({Attack}) m_3({No Attack}) + m_1({No Attack}) m_3({Attack}) + m_2({Attack}) m_3({No Attack}) + m_2({No Attack}) m_3({Attack}) = 0.8*0.3 + 0.1*0.6 + 0.8*0.5 + 0.1*0.0 + 0.6*0.5 + 0.3*0.0 = 1

So $m_{1,2,3}(A)$ = (1/1-1) *0.1*0.1*0.5 = n/a - cannot divide by 0

Belief combination of three values - $\boldsymbol{Bel(A) = Bel_1(A) \oplus Bel_2(A) \oplus Bel_3(A)}$
(m_1) we have belief that the proposition is true for state Attack which is 0.3 (i.e. m_1({Attack}) = 0.8) and similarly m_1({No Attack}) = 0.1 with m1({Either}) = 0.1

Then we take another assessment m_2 with m_2 ({Attack}) = 0.6, m_2({No Attack}) = 0.3 with m_2({Either}) = 0.1

m_3 associated values include {Attack}) = 0.9, m_3({No Attack}) = 0.05 with m_3({Either}) = 0.05

The joint mass function would be $m_{1,2,3}(A)$ = (1/1-K) m_1({Either})m_2({Either}) m_3({Either}) with:
K = m_1({Attack}) m_2({No Attack}) + m_1({No Attack}) m_2({Attack}) + m_1({Attack}) m_3({No Attack}) + m_1({No Attack}) m_3({Attack}) + m_2({Attack}) m_3({No Attack}) + m_2({No Attack}) m_3({Attack}) = 0.8*0.3 + 0.1*0.6 + 0.8*0.05 + 0.1*0.9 + 0.6*0.05 + 0.3*0.09 = 0.73

So $m_{1,2,3}(A)$ = (1/1-0.73) *0.1*0.1*0.5 = 0.01851

4 Evaluation

The use of D-S rule is mathematically possible only if m^a and $m^b$are not conflicting, i.e. if there is a focal element of m^a and a focal element z of m^b satisfying $(y \cap z) \neq \emptyset$. Merging two belief masses with the conjunctive rule defined above produces a sub-additive belief probability assignment, meaning that the sum of belief masses on focal elements can be less than one, in which case it is assumed that the missing or complement belief mass gets assigned to the empty set. If desirable, the normality assumption m(/0) = 0 can be recovered by dividing each belief mass by a normalization coefficient (Josang & Pope, 2012). This rule is associative, and the normalisation in D-S redistributes conflicting belief masses to non-conflicting ones, and tends to eliminate any conflicting characteristics in the resulting belief mass distribution. This rule of combination can be applied to avoid this particular problem by allowing all conflicting belief masses to be allocated to the empty set.

When performing the belief calculations by two values the returned result is quite surprising. When comparing two high belief generations the assumption is that the combined belief value would also be a high number, however it is a lower value. The correlation between high belief values and low fused outputs suggests that the lower the fused output the higher the risk. The same is understood for two fused low belief values generate a high fused output, which would be a low risk. It is not clear if this is due to our calculations but these metrics have been compared on numerous belief fusions and this is a similar occurrence. The mass value must be between 0 and 1 but not inclusive as this seems to skew the calculations, e.g. using a value of 0 would render the combination calculation ($m_{1,2}$(A)) uncalculatable as you cannot divide by 0 which would be a pertinent value. A coefficient value of 1 would leave the combination calculation having to divide by 0 (1/1-1) which is an impossible calculation. Also, having a coefficient of 0 would give a negative risk output, which is also an unusable value.

Implementing our Security as a Service solution, the below limitations of D-S can be evident:

- Associative – for rule combination, the order of the information in the aggregated evidences does not impact the result. A non-associative combination is necessary for many cases.
- Non-weighted – rule combination implies we trust all evidences equally. However, in reality, our trust on different evidences may differ, which means we should consider various factors for each evidence.

We have demonstrated how D-S can provide collaborative intrusion detection, however there may be cases where the decision may be inaccurate, and if a domain under attack generation the Belief of origin then it would still need to take action against the condition. D-S when applied in an autonomous collaborative environment should apply a weight of confidence when the belief generation occurs. If CSPs collaboratively vote no attack, but one CSP is adamant it is being attacked, there should be a way to overrule the decision based on the strength of the associated trust or confidence value. The algorithm should be extended further to take this into consideration, and we propose a two stage collaborative detection process for conflicting decisions.

Two stage D-S fusion for conflicting decisions is an option for solving this issue. Post Belief generation processing is needed for application to this area to facilitate information exchange for defence. Via the inclusion of confidence values the accuracy of decisions can be improved. Protecting the local services of the CSP but proactively warning others of the potential threat. If the fused decision is "No Attack" but the Belief of origin has a high confidence value, then the domain of origin would take action against the suspect observation, but send the Belief value $(B_A(IP, timestamp, Confidence\ Value)$ to the Broker to store in its local Grey list. Should an adjacent CSP query the Broker regarding the suspect IP in the future, it has the information from the origin CSP.

5 Conclusion

We have presented our Security as a Service solution, a novel platform for the protection of infrastructure services in a federated Cloud environment. D-S theory is extended to meet the domain management needs and to facilitate autonomous sharing of information. The novel contributions of this project are that it provides the means by which DDoS attacks are detected within a cloud federation, so as to enable an early propagated response to block the attack, particularly by the interdependent CSPs within the federation. This is effectively inter-domain cooperation as these CSPs will cooperate with each other to offer holistic security, and add to the defence in depth. D-S is used to facilitate this autonomous sharing of information, and to fuse the generated beliefs to form a system wide decision. This cooperation between CSPs ensures the scalable defence required against DDoS attacks is in an efficient and cost effective way. Protecting the federated cloud against cyber-attacks is a vital concern, due to the potential for significant economic consequences. The effects of attacks can span from the loss of data, to the potential isolation of parts of the federation. Our simulations offer proof of concept, and deem the applicability of D-S to this area promising but still with evident limitations.

Acknowledgements

The work reported in this paper is partly supported under the Newton Research Collaboration Programme by the Royal Academy of Engineering.

References

Josang, A., & Pope, S. (2012). Dempster's Rule as Seen by Little Coloured Balls. *Computational Intelligence*, *28*(4), 453–474.

MacDermott, Á., Shi, Q., & Kifayat, K. (2015). Collaborative intrusion detection in a federated Cloud environment using the Dempster-Shafer theory of evidence. In *European Conference on Information Warfare and Security, ECCWS*.

MacDermott, Á., Shi, Q., & Kifayat, K. (2015). Collaborative Intrusion Detection in Federated Cloud Environments. *Journal of Computer Sciences and Applications on Big Data Analytics in Intelligent Systems*, *3*(3A), 10–20.

Rak, M., Ficco, M., Luna, J., Ghani, H., Suri, N., Panica, S., & Petcu, D. (2012). Security Issues in Cloud Federations. In *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice* (pp. 176–194). http://doi.org/10.4018/978-1-4666-1631-8.ch010

Ten, C. W., Manimaran, G., & Liu, C. C. (2010). Cybersecurity for critical infrastructures: attack and defense modeling. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *40*(4), 853–865. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5477189

Thomas, C., & Narayanaswamy, B. (2011). Sensor Fusion for Enhancement in Intrusion Detection. In *Sensor Fusion - Foundation and Applications* (pp. 61–76).

Trend Micro Incorporated. (2015). *Report on Cybersecurity and Critical Infrastructure in the Americas*.

Xiao, Z., Xiao, Y., & Member, S. (2013). Security and Privacy in Cloud Computing, *15*(2), 843–859.