Implementing a Trusted Service-Oriented Network Environment

E. A. Adigun* and J.H.P. Eloff†
Information and Computer Security Architectures (ICSA) Research Group
Department of Computer Science, University of Pretoria
South Africa
Email: eadigun@cs.up.ac.za*, eloff@cs.up.ac.za†

Abstract—Trust has become an important feature of security. Originally having its roots in the social sciences, it has fast moved into the field of computer science and more recently into the domain of network security. It is becoming important for scientists to model human trust, especially in a network environment where a lack of trust between network devices, such as routers, can result in poor quality of service and an increase in security compromises. In a network environment there is no “behaviour” indicator for routing devices about other devices on their forwarding path. Thus, relying on another routing device might be difficult if a secure and reliable service is not available. In this paper we present our ongoing implementation of trust in service-oriented network environments and explain how trust is implemented between network devices.

I. INTRODUCTION

Trust is “a firm belief in the reliability of a person or thing” 1. Trust governs every aspect of human life. It is connected, directly or indirectly, to our daily activities. For example, computer users, to some extent, trust their computers operating system to run an application that may help them complete a specific computing task. A user uses the application because of its trustworthiness and reliability, which may be the result of recommendation by other people who found it useful. The trustworthiness of the application, in other words its reputation, gives it credibility among its users. In a human context trust is not quantifiable because it is invisible and implicit in society. It can be described as a “feeling” or attribute (good or bad) bestowed on the trusted or untrusted subject. Trust is also context and situation specific – it may depend on past interaction and experience to determine trustworthiness or untrustworthiness. Trust in something or someone helps us take risk in situations where there is partial information [1]. A network environment with special reference to the service level, as opposed to application level, essentially constitutes an environment where there may be little or partial information about its participants.

A network environment on the service level consists mostly of routing devices that communicate on an ad hoc basis. The interconnection of these devices spans multiple geographical boundaries and the network may vary from small (a local area network) to huge (a wide area network). In this network service environment only partial information, such as a forwarding address, is available to routing devices. There is a complete lack of behavioural information, for example, information regarding the availability of devices over time. A recent incident involving YouTube that affected the Internet traffic in the whole of Asia Pacific being routed via Pakistan is a good example [2]. It is argued that if a framework can be developed for modelling the trustworthiness of a network device, such as a router, then its reputation amongst other network devices can assist in preventing security damaging incidents.

Related work in this area has concerned itself with implementing trust models for distributed systems and virtual communities [3], [4]. Some authors have also drawn a comprehensive comparison and evaluation of known trust models [5], [6]. However, the models are application-oriented in that the entities implemented in these environments are completely autonomous and are executed on an application level. These entities manage their trust value and, based on their trust calculations, they decide to trust or not to trust another entity in another application.

The objective of this work is to illustrate how the trust-building process can be implemented in a service-oriented network environment (SONE). Trust for this type of environment aims to establish availability and secure communication. Different information gathered from the service-providing devices in this environment is used to compute trust on a network management server in our environment. For implementation purposes, we set up a live network consisting of routing devices and a server to manage the routing devices in this environment. Additionally, we used the trust model proposed by Abdul-Rahman [3] to determine trust. The trust model is implemented on the network management server. We aim to illustrate our proposed model with the Universal Modelling Language’s (UML) activity class diagram.

The layout of the paper is as follows: Section II provides background information on a service-oriented network environment and how we view trust in this environment. Section III gives a high level overview of the proposed TSONE, trust in service oriented network environments. Section IV contains the details of our implementation, the network environment and how trust is computed. The paper is concluded in Section V.

1Concise Oxford English Dictionary, Eleventh Edition
II. BACKGROUND

Service-oriented environments are associated with business activities such as buying, selling of goods, entrepreneurial activities [7] and provision of service for a particular need. These services are associated with the Internet and synonymous to services provided by, inter alia, eBay [8] or an online ticket-booking service. Taking the association a step further, on a conceptual level, we would like to consider network devices as service providers in a network environment.

In this section we describe what a service-oriented network environment is and provide concise information on routers as a main service provider. We also give a brief explanation of our view of trust in this environment.

A. Service-Oriented Network Environment (SONE)

Motivation to consider a network environment as a service environment came from service-oriented computing (SOC). SOC provides a framework for interoperation and communication between heterogeneous applications. From SOC came the idea of a service-oriented architecture (SOA). SOA defines principles for implementing a service-oriented environment which includes among others loose coupling, implementation neutrality and flexible configurability [9], [10].

A network environment - or more specifically, the Internet - embraces the three principles mentioned above because it is an environment that incorporates different devices/nodes. The devices are situated at different locations around the world (loosely coupled); they use different operating systems (implementation neutrality); and their setup properties are different. The setup properties, however, correspond to the extent that they can understand the different protocols used (flexible configurability).

Therefore, we define a Service-Oriented Network Environment (SONE) as a collaborative, shared and open community in which network devices provide secure and reliable service [11]. We view service provision on the Internet in terms of packet routing and forwarding. As mentioned earlier, routers are our main focus since they perform most of the routing on the Internet network and are known as Internetwork Devices (Fig. 1). Routers are networking devices that can decide how to forward packets through a network by examining network layer information [12]. The services provided by routers are packet forwarding, routing and more specialised services such as IPsec encryption and many others. These services depend upon the network service model adopted by the network architecture.

B. Trust

Trust varies from one context to another as was discussed above. However, trust is still a risk and a subjective probability [13] that a trusted entity can be relied on. The subjective probability of trust, if modelled according to human trust, cannot be quantified because situations where trust is applied are not always the same. This illustrates another property of trust - it is situation dependent. The fact that trust is not quantifiable does not mean that there are no numeric values that represent trust attributes. There are numeric values, but they will merely act as place holders since trust in an entity will not be represented as an absolute numeric value.

Various types of trust classification are available in literature. Two of the classifications that apply to this work are delegation and provision trust [14]. Delegation trust is a type of trust in another entity to act on your behalf regardless of your reputation, while provision trust is trust in a service provider to provide a reliable service. These forms of trust (delegation and provision trust) are applied in one way or another in this work.

Abdul-Rahman [3] gave specifications for a trust model called Network Trust Opinions (Ntropi), which is implemented for virtual communities. The Ntropi conceptual model is based on the formation of trust opinions from direct experiences and recommender information about an active entity referred to as an agent. Therefore, there are two types of trust identified in the model: direct trust and recommender trust. Trust opinions formed are context dependent; that is, a recommender’s recommendation is only trustworthy for a particular context. The model also emphasises that a bias might exist on the recommenders side in terms of its recommendations accuracy. In evaluating direct trust, trust degrees are defined: very trustworthy, trustworthy, untrustworthy and very untrustworthy. Two agents in a 'trust' relationship might also go through the following phases: unfamiliar, fragile, stable and untrusted.

III. TRUST IN SONE (TSONE)

TSONE is an environment for incorporating trust (T) in a Service-Oriented Network Environment (SONE). In the modelling of TSONE, the reputation of the routing devices must be part of the design as it allows a trustworthiness factor to be attributed to the devices. To model reputation in this environment, trust attributes of the routing devices are considered over a period of time. For example, a routing device with a low trust attribute calculated over a period of time will have more credibility compared to a new routing device with an average trust attribute.

Trust for a router is computed and maintained on a network management server (NMS). The TSONE network environment is depicted in Fig. 2. The trust model implemented on the
server computes and manages trust attributes for each router on the network. Routers’ activities are collected in a Syslog file, through the Syslog protocol, on the NMS, and are stored in a database. Based on the activities in the Syslog file (Fig. 3) and the trust model output, the network administrator – via the simple network management protocol (SNMP) – can increase or decrease a path cost on the router. A path cost represents some preference indicator for selecting a path among other paths for the purposes of packet forwarding. The higher a path cost on a router, the less likely it is that the router will be used for that path. The activities of a router cannot be predetermined unless via intervention from a network administrator. Hence, if a router functions as it should, its trust attribute will be “impeccable”. On the other hand, what if a well-behaved malicious router just pops up on the network? As part of the trust functionality of the network, routers must log their activities to the network management server and the NMS must know their read/write community (Section IV-A) to communicate with the router. If there are no logging details and the NMS is not aware of the router, the router should not be used. The network administrator configures logging details on the router and the NMS. The issue of trust in a network administrator was kindly pointed out by a reviewer, even though out of scope of this work, human trust or more specifically trust in humans is dynamic and unpredictable. However, the onus remains with an organization to set up a process of identifying a trustworthy network administrator and a reliable reporting process of changes made to the network.

IV. IMPLEMENTATION

The implementation of this model is divided into two parts, the network environment and the trust model (Fig. 2 above). To implement the network environment, a live network was set up with routers running OSPF (Open Shortest Path First) routing protocol. Further information about the live network environment is provided in the following sub-section IV-A. The second part of the implementation involves implementing and using a trust model to calculate trust in this environment. Abdul-Rahman [3] gave specifications for a trust model called Network Trust Opinions (Ntropi), which is implemented for virtual communities. We employ part of this trust model in our implementation.

A. The network environment

The live network environment consists of two routers and a switch: Cisco 1601R Series, Cisco 1760 Series and a Cisco Catalyst 4200 Series respectively. The routing devices are on loan from Cisco for the purpose of this project, and they run the most current IOS (Internetwork Operating System). The environment, shown in Fig. 4, also consists of a network management server (NMS), which collects log information from the routing devices. A summary of this information is passed on to the trust model (Section IV-B). The final decision, made by the trust model – based on calculations and past activities, is used by the administrator to manipulate path cost or route information available on the router.
of network management stations and network devices \(^2\)\(^1\)\(^5\), \(^2\). It enables network devices such as routers and the like to respond to get and set requests via a network management entity (NME) on network devices.

An NME, for the purposes of network management is referred to as an agent \(^2\)\(^1\)\(^5\) and it is a collection of software that performs network-management task on a network device. Agents in a network can be end systems supporting user applications as well as nodes providing communication services. Agents can also be routers and are thus used interchangeably in this article.

Communication and network related activities are gathered via get and set requests. These requests, respectively, access and modify objects that are data variables representing an aspect of a managed agent. The collection of these objects is referred to as management information base (MIB) \(^2\)\(^1\)\(^5\). Consequently, MIBs are used to access and sometimes modify objects on routing devices. To protect MIBs from unauthorised access, a managed agent and management station communicate via a predefined read and write community. The latter is locally defined at the managed agent \(^2\)\(^1\)\(^5\). To authenticate a management station, the managed agent checks that every message sent to it includes a community name. If the community name is missing or invalid, an error message is displayed (see Fig. 5 and Fig. 6).

\(\text{Fig. 5. Missing Community Name with SNMP request to MIB sysName}\)

\(\text{Fig. 6. Valid Community name with SNMP request to MIB sysName}\)

B. Trust Model

The trust model used in SONE is known as Ntropi (network trust opinions) \(^2\)\(^3\). Our trusted service-oriented network environment adopt some of Ntropi’s data structure, that is, trust level and phase value. However, other data structure in Ntropi such as recommender trust is not existent in TSONE because the NMS collects and aggregates experiences logged by a router. We define a new approach to determine the final trust attribute as shown in our activity diagram (Fig. 7 below).

In the trust model section of the activity diagram (Fig. 7), severity of an event or a logged message indicates the seriousness and/or importance of a Syslog message. An example of severity could be emergency, critical or debug. Each severity level is also represented as a single-digit code from 0 to 7, the lower the number the more serious an event is. See Fig. 3 above. The seriousness of an event, hence its severity, will impact a router’s trust level therefore the more serious an event the higher the impact on the router’s trust attribute and the less serious an event the lower the impact.

\(\text{RFC 3164}\) \(^3\) on BSD syslog mathematically defines priority as follows:

\[
priority = (facility \times 8) + severity
\]  \((1)\)

priority of logged messages is used because it can be represented as a numerical value as indicated above (Equ. 1). And it is also used to determine the impact of trust attribute on a router’s profile (Equ. 2 & 3).

\(^2\)http://rfc.net/rfc1157.html

\(^3\)http://rfc.net/rfc3164.html
In the activity diagram (Fig. 7), an event (instance of a syslog message) counter is kept for each router to keep track of newly logged information. If it is a new router, the counter will be zero. However, for a returning router, new events are added to previous event counts. The next function checks the phase value to determine if a new trust level must be calculated for the router. In our trusted service-oriented network environment, we use the trust model’s phase value as the main trust attribute and use numeric counters inside each phase value. When the highest value (threshold) is reached in a particular phase, the router moves on to the next “trust” phase. When the final phase value for a session is decided, the trust model will either stay the same, or it will decrease or increase the trust attribute of the router. The magnitude by which that trust attribute increases or decreases should be based on the priority calculated above (Equ. 1).

For example, the minimum value that priority can take is 0 and the maximum is 191. If we define an adjusted \( a \) scale between 0.5 and 1.5 to represent our real \( r \) scale 0 and 191 respectively and a chosen midpoint of 70. The factor by which the trust attribute increases or decreases will be defined by:

\[
a - 1.5 = \frac{0.5}{121} \times (r - 191) \tag{2}
\]

and

\[
a = 1.5 + \frac{0.5}{121} \times (r - 191) \tag{3}
\]

The value derived from \( a \) will be used to determine how much a router’s trust attribute increase or decrease. The midpoint value provided above is a rough estimate of the impact should have a minimum or maximum effect.

C. Development Environment

TSONE is being developed in Java. The routers interface with a third party application–Kiwi Syslog \(^4\) on the NMS to send their log files. Log files are in raw text format and are read into TSONE (Fig. 3) via classes that are created to extract information from the Syslog file. SNMP communication to the router is enabled via SNMPAPI \(^5\), a network administrator might use this to manually configure the routers based on trust attribute calculated in the trust model or latest information available in the Syslog.

V. Conclusion

It is important to note that the proposed scheme is not an architecture meant for the Internet network. Mainly because it is impossible to control all the routing devices on the Internet network with a centralised architecture. The proposed scheme is ideal for a local area network and scalable for as many routers are under the control of an Internet service provider. Hence communication with another network in a different domain authority will be based on trusting routers at the edge of that network.

---

\(^4\)www.kiwisyslog.com
\(^5\)www.adventnet.com

In this paper we discussed an ongoing implementation of trust in a service-oriented network environment. We defined what a service-oriented network environment is. We discussed how trust can be applied to this environment and how trust is being implemented in this environment. We presented our implementation, network setup and the different network management protocols used to monitor network devices. We introduced a trust model of our choice, which is being used for the implementation. We also diagrammatically represented our model in an activity class diagram.

The implementation is still ongoing thus no test validation has been carried out. When the implementation is complete an evaluation of the proposed scheme will be carried out to determine its effectiveness.

ACKNOWLEDGEMENT

This material is based upon work supported by Telkom, South Africa. Any opinion, findings, conclusions or recommendations expressed in this material are those of the authors and Telkom does not accept any liability there to.

REFERENCES

E. A. Adigun is currently studying towards a Master’s degree in Computer Science at the University of Pretoria. He obtained an honours degree in Informatics and prior to that a Bachelors degree in Data Communications and Networking at the University of Zululand.

J.H.P. Eloff received a PhD (Computer Science) from the Rand Afrikaans University, South Africa. He gained practical experience by working as management consultant specialising in the field of information security. Since October 2002 he has been Head of the Department and full professor in Computer Science at the Department of Computer Science, University of Pretoria.