In Whom Do We Trust - Sharing Security Events

Jessica Steinberger[1,2](✉), Benjamin Kuhnert[1], Anna Sperotto[2],
Harald Baier[1], and Aiko Pras[2]

[1] da/sec - Biometrics and Internet Security Research Group,
Hochschule Darmstadt, Darmstadt, Germany
{Jessica.Steinberger,Benjamin.Kuhnert,Harald.Baier}@h-da.de
[2] Design and Analysis of Communication Systems (DACS),
University of Twente, Enschede, The Netherlands
{J.Steinberger,A.Sperotto,A.Pras}@utwente.nl

Abstract. Security event sharing is deemed of critical importance to counteract large-scale attacks at Internet service provider (ISP) networks as these attacks have become larger, more sophisticated and frequent. On the one hand, security event sharing is regarded to speed up organization's mitigation and response capabilities. On the other hand, it is currently done on an ad-hoc basis via email, member calls or in personal meetings only under the premise that participating partners are personally known to each other. As a consequence, mitigation and response actions are delayed and thus security events are not processed in time. One approach to reduce this delay and the time for manual processing is to disseminate security events among trusted partners. However, exchanging security events and semi-automatically deploying mitigation is currently not well established as a result of two shortcomings. First, the personal knowledge of each sharing partner to develop trust does not scale very well. Second, current exchange formats and protocols often are not able to use security mechanisms (e.g., encryption and signature) to ensure both confidentiality and integrity of the security event information and its remediation. The goal of this paper is to present a trust model that determines a trust and a knowledge level of a security event in order to deploy semi-automated remediations and facilitate the dissemination of security event information using the exchange format FLEX in the context of ISPs. We show that this trust model is scalable and helps to build a trust community in order to share information about threats and its remediation suggestions.

Keywords: Sharing security events · Attack mitigation · Internet service provider · Network security

1 Introduction

Nowadays, large-scale cyber attacks (e.g., Distributed Denial of Service (DDoS) attacks) have become larger, more sophisticated (e.g., multi-vector attacks) and

R. Badonnel et al. (Eds.): AIMS 2016, LNCS 9701, pp. 111–124, 2016.
DOI: 10.1007/978-3-319-39814-3_11

frequent [1]. These large-scale cyber attacks are responsible for network and service outages and thus are causing brand damage and financial loss. To counteract these attacks, one approach that gained increasing attention in recent years is to semi-automatically disseminate cyber threat information among trusted partners [2,3] to facilitate collaboration. However, current collaborative cyber defense is founded on an ad-hoc basis via email, member calls or in personal meetings and thus a manual process [4,5]. This slows mitigation and response times and impedes mitigation and reaction efficacy [6]. Besides the fact that collaboration and information sharing often only takes place in case participating partners are personally known to each other, some legally binding orders (e.g., Executive Order 1363 [7,8]) have been published recently that force owners and operators of critical infrastructures to establish procedures to increase the volume, timeliness and quality of cyber threat information sharing.

To improve the timeliness of cyber defense, support collaboration among trusted partners and facilitate the dissemination of security events, a common data representation and security mechanisms to establish trust are required. Even though several exchange formats (e.g., Incident Object Description Exchange Format (IODEF) [9], Intrusion Detection Message Exchange Format (IDMEF) [10], Abuse Reporting Format (ARF) [11], Extended Abuse Reporting Format (x-arf v0.1 and v0.2) [12] and Flow-based Event eXchange Format (FLEX) [13]) have been published [14] to exchange security events or incidents, the majority of the exchange formats and protocols do not provide any security mechanisms to sign or encrypt a security event [14].

Besides the lack of a standardized exchange format and protocol, the development of trust is deemed of critical importance to share security events. Despite well-known and established trust models are used in other application contexts, the personal knowledge of each sharing partner to develop trust in order to share security events does not scale very well.

To overcome the constraint of personal knowledge of each sharing partner to develop trust in context of mitigation and response to large-scale cyber attacks and to establish an effective collaboration among trusted partners, this paper presents a trust model, called MiRTrust. MiRTrust determines a trust and a knowledge level of a security event in order to deploy semi-automated remediations and facilitate the dissemination of security events using FLEX in the context of ISPs. MiRTrust is based on the well known PGP trust model [15,16] and used to establish different levels of trust, determine the prioritization of the shared security event, sanitize the occurrence of security events and contributes to build a trust community in order to share information about cyber threats and its remediation suggestions.

The paper is organized as follows. In Sect. 2, we describe the scenario in which the trust model is going to be used. Next, we present the requirements that are derived from the presented scenario. Section 3 presents the foundation and related work. Our trust model MIRTrust is presented in Sect. 4. In Sect. 5, we evaluate our trust model MiRTrust. Finally, the paper is concluded in Sect. 6.

2 Scenario and Requirements

In this Section, we describe the main focus of this work. First, we define the networks in which we are going to place our trust model to facilitate the semi-automated assessment and deployment of remediation suggestions. Second, we define the requirements that a trust model should fulfill, as they emerged by the scenario described in Sect. 2.1. In the following, we will use these requirements to evaluate the trust model. Attacks targeting the trust model are out of scope of this work.

2.1 Scenario

The primary focus of this work are multiple high-speed networks using a link speed of 10 Gbps and higher [17]. In addition, we focus on network operators that cooperate among trusted partners to minimize or prevent damages caused by network-based attacks and use an automated threat information exchange. The collaboration is established using an infrastructure based overlay network [18] to prevent a full mesh within the network and to ensure scalability. Each participating partner receives security events from different origins as shown in Fig. 1. Security events originating from a detection engine within the own network infrastructure is defined as an internal security event and shown in Fig. 1a. Further, each participating ISP possesses a list of directly connected collaborating partners. In case of ISP a a directly connected collaborating partner is ISP c as shown in Fig. 1b. The networks of ISP b, ISP d and ISP e are not directly connected to ISP a and thus are regarded as external non collaborating partners as shown in Fig. 1c.

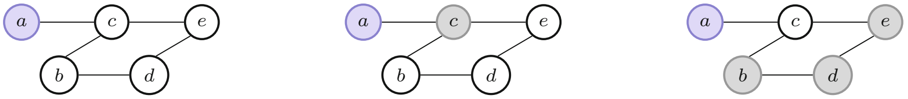
2.2 Requirements

In this section, we introduce five requirements that a trust model should fulfill in order to establish collaboration among trusted partners. These requirements are derived from European Network and Information Security Agency's (ENISA) position paper no. 2 [19] and the work of [20].

Ease of Deployment: The trust model and its underlaying implementation should support platform independency to ensures that they easily integrates with the existing infrastructure.

Access Control: The trust model should support the use of the Traffic Light Protocol (TLP) [21]. The reason is that the TLP provides a scheme for sharing different detail of information tailored for its intended receivers. The reason is that the amount of provided threat information depends on the trust and sharing relationship between collaborating ISPs.

Subjectivity: The trust model should provide the possibility that network operators are able to form their own trust options. These trust options represent the degree of belief about the behavior of collaborating partners.



(a) Internal **(b)** External collaborating **(c)** External non collaborating

Fig. 1. Origin of security events

Asymmetry: The trust model should support asymmetric levels of trust for both collaborating partners as they do not need to have similar trust in each other.

Decentralized: Each trust model within the mitigation and response (MiR) system should act as a self-contained unit and thus calculates its trust decisions locally. The MiR system should exchange these decisions in form of recommendations with its directly connected collaborating partners.

3 Related Work

In this section, we introduce the terminology by defining trust, review reputation-based trust models and analyze existing collaboration communities used to mitigate and response to large-scale attacks.

3.1 Terminology

Currently, there is no consensus in the definition of trust available. The authors of [22] reported that there are various definitions of trust based on the use-case context. In this paper, we adhere to the following definition of trust: “Trust is the quantified belief by a trustor with respect to the competence, honesty, security and dependability of a trustee with a specific context [22].” Besides trust, we also adhere to the following definition of distrust: “Distrust is the quantified belief by a trustor that a trustee is incompetent, dishonest, not secure or not dependable within a specific context [22].”

3.2 Collaboration Communities

The majority of the collaboration communities are private communities that require a membership application and are charging an annual fee. Recent well-known collaboration communities that require a membership application and are charging an annual fee are the Anti-Phishing Working Group (APWG), the Messaging, Malware and Mobile Anti-Abuse Working Group (M^{3}AAWG), the Research and Education Networking (REN) Information Sharing and Analysis Center (REN-ISAC) and the Forum of Incident Response and Security Teams (FIRST). In contrast to the fee-based collaboration communities are non-fee-based collaborations. The Advanced Cyber Defence Centre (ACDC), the Gigabit

European Academic Network (GÉANT) Task Force on Computer Security Incident Response Teams[1] (TF-CSIRT) and the Gigabit European Academic Network (GÉANT) Special Interest Group on Network Operations Centres[2] (SIG-NOC) and the DDoS Open Threat Signaling[3] (DOTS) working group within the IETF are well-known collaboration communities in context of security event sharing.

The collaboration of APWG focuses on eliminating the identity theft and fraud that result from the growing problem of phishing and email spoofing [23]. M^{3}AAWG is working against bots, malware, spam, viruses, DoS attacks and other online exploitation [24]. REN-ISAC is sharing sensitive information regarding cyber security threat, incidents, response, and protection located in United States, Canada and New Zealand [2], and FIRST cooperatively handles computer security incidents and promote incident prevention programs from around the world. ACDC focus on detection, mitigation and response of botnets. Further, ACDC also supports the mutal data sharing between partners (e.g., ISPs, government agencies, law enforcement, research groups, industry partners). TF-CSIRT and SIG-NOC facilitates knowledge exchange and collaboration in a trusted environment in order to improve cooperation and coordination. DOTS is developing a standards based approach related to DDoS detection, classification, traceback and mitigation in context of a larger collaborative system at service provider level.

All of the aforementioned collaboration communities require and provide different level of memberships, whereas the fees vary from \$250 to \$25 000. In addition, each application initiates a review process which is performed by the community and decides about acceptance to join. Some communities perform collaboration following the following the Chatham House Rules[4] (e.g., M^{3}AAWG). The number of community members within a fee-based community vary from \$200 to \$1 800 and FIRST is mentioned to be the oldest and biggest international collaboration community for CERTs [25].

3.3 Reputation-Based Trust Models

e-Commerce: The trust model of e-Commerce is often a centralized reputation-based system that rely on feedback of the involved parties. This feedback system is used in eBay, AirBnB, Booking and Amazon and is a primary resource for potential buyers to determine the trustworthiness of the seller. A feedback consists of comments and five different ranking levels to evaluate several aspects (e.g., price, condition, timeliness). Further, the overall feedback score consists of a positive, neutral or negative rating. A positive feedback adds +1, a negative feedback adds −1 and a neutral feedback 0 to the overall feedback score. To calculate the overall feedback percentage, the ratio of feedback scores is computed.

[1] http://www.geant.org/Innovation/SIG_TF/Pages/TF-CSIRT.aspx.

[2] http://www.geant.org/Innovation/SIG_TF/Pages/SIG-NOC.aspx.

[3] https://datatracker.ietf.org/wg/dots/charter/.

[4] https://www.chathamhouse.org/about/chatham-house-rule.

Web of Trust: The trust model *web of trust* (WOT) describes a decentralized public-key infrastructure (PKI) relying on trust decisions of individual participants [16]. It is used in PGP, GnuPG and OpenPGP. The basic WOT uses three levels of trust: complete, marginal and no trust. In addition, PGP, GnuPG and OpenPGP distinguish unknown trust from no trust and thus differentiate between 5 trust levels [15,16]. Each participating partner owns a personal collection of certificates called the key ring and is allowed to sign a key for any other participant. [16] reported that the trust model accepts a given public key in the key ring as completely valid, if either (i) the public key belongs to the owner of the key ring, (ii) the key ring contains at least C certificates from completely trusted certificate issuer with valid public keys and (iii) the key ring contains at least M certificates from marginally trusted certificate issuer with valid public keys. The default values in PGP are $C = 1$ and $M = 2$, whereas GnuPG uses $C = 1$ and $M = 3$. The calculation of the trust level is described by [16] as follows: The key legitimacy $L = \frac{c}{C} + \frac{m}{M}$, where c and m represents the number of certificates from completely/marginally trusted certificate issuers with valid keys. A key is completely valid for $L \geq 1$, marginally valid for $0 < L < 1$, and invalid for $L = 0$.

4 Trust Model

Our mitigation and response trust model (MiRTrust) is based on hTrust [26], a trust management model to facilitate the construction of trust-aware mobile systems and applications and on the PGP trust model [15,16]. In contrast to hTrust, MiRTrust uses the four GnuPG [27] trust levels: *unknown*, *none*, *marginal* and *full* and additionally the trust level *distrust*. Moreover, MiRTrust takes into account the EU Trusted Lists [28] and the Alexa top 10 million websites list in order to extract the use of certification authorities using a 3 months average ranking. Unlike hTrust, MiRTrust does not consider contexts as the security events are identified in the context of ISPs and result from a large-scale attack.

MiRTrust consists of several input parameters (yellow colored) and three components: trust formation (blue colored), trust dissemination (brown colored) and trust evaluation (green colored) as shown in Fig. 2. The component trust formation is responsible to determine the trustworthiness of a security event before a semi-automated mitigation and response action is taken. In case a security event from a new collaborating partner or an unknown source was received, the trust dissemination guarantees a minimum set of information upon the predication of trust can be calculated. The last component, trust evaluation, is responsible to continuously self-adapt the trust information kept in the ISP's local trust list.

A trusted-based collaboration relies on two participating partners exchanging security events, where as trust has the following three characteristics: (i) Trust is not symmetric. If ISP a trusts ISP b, it does not follow ISP b trusts ISP a. (ii) Trust is not inherently transitive. If ISP a trusts ISP b and ISP b trusts ISP c, it does not automatically follow that ISP a trusts ISP c. (iii) Trust of own detection engines varies in a range from marginal trust to full trust, as false

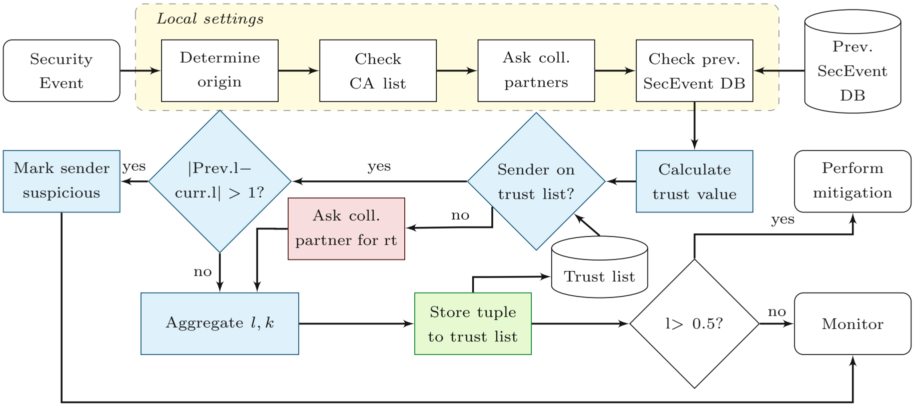


Fig. 2. Trust level calculation of MiRTrust (Color figure online)

positives are possible. A security event is described as the quadruple (a, b, s, t). The quadruple can be described as follows: ISP a informs ISP b about a security event of type s occurring at time t. The sender of a security event is referred to as trustee, whereas the receiver of a security event is called trustor. ISP b is a trustor that forms a trust opinion about the trustee ISP a based on b's previous trust experiences with a. The process to form a trust opinion about a trustee is shown in Fig. 2. The trust experiences are stored locally at each MiR system and are described by a 6-tuple: (a, b, s, l, k, t). The tuple can be described as follows: ISP b trusts ISP a at level l about the security event type s in context of large-scale attacks. The trust level l is denoted as $l \in [-2, 2]$, whereas -2 represents distrust, -1 represents unknown trust, 0 represents no trust, 1 represents marginal trust and 2 represents full trust. In accordance to [26], MiRTrust also considers only partial knowledge about the trustworthiness of collaborating partner. The reason is that only directly interconnected networks are collaborating and thus their trust opinions contain a level of uncertainty. This uncertainty is expressed as knowledge k and varies from a trust based decision *do not trust* to a lack of evidence based decision *do not know*. The knowledge k is denoted as $k \in [0, 1]$, whereas 0 represents unknown and 1 perfect knowledge. Both, the trust level l and the knowledge k is retrieved from local settings and past experiences. The better the experience in the past, the higher the trust level l and the knowledge k. To relate trust and knowledge to time, MiRTrust uses the variable t to refer at which time t the trust t and knowledge k was calculated.

Local Settings: In a first step, MiRTrust computes a trust range $\Upsilon[lb, ub]$ and an initial knowledge value based on the origin of the security event. In case of an internal security event $\Upsilon[lb, ub]$ is set to $\Upsilon[1, 2]$ and the knowledge value is set to $k = 1$. The trust range of security events originating from external collaborating partners is set to $\Upsilon[0, 1]$, where as the trust range of security events originating from external non collaborating partners is set to $\Upsilon[-2, 0]$. The knowledge value of security events originating from external collaborating partners is set to $k = 0.5$ and from external non collaborating partners to $k = 0$. Next, MiRTrust takes into account several local settings ls. The basic setup of MiRTrust considers three local settings: ls_1 = *check CA list*, ls_2 = *ask collaborating partners* and ls_3 = *check previous security event database* (DB) that all evaluate to a boolean value. ls_1 describes if IP addresses or domains within the security event are listed on the merged CA list. This CA list combines the EU Trusted list and the used certification authorities of the Alexa top 10 million websites. ls_2 describes whether the behavior that cause the security event has also been seen in collaborating partner networks [29]. ls_3 refers to security events with similar behavior that have been received and stored previously.

Trust Formation: The trust formation enables a trustor to predict a trustee's trustworthiness before mitigation and response actions are initiated. Therefore, the function p is used to calculate a trust value as shown in Eq. (1). p uses a weight w to emphasize the importance of a local setting ls. The importance of these values are defined by each participating ISP. The function $c(ls_i)$ is used to decide which value of the trust range $\Upsilon_{[lb,ub]}$ is multiplied with weight w.

$$p(w_1, \ldots, w_n, ls_1, \ldots, ls_n, \Upsilon_{[lb,ub]}) = \sum_{i=1}^{n} w_i \cdot c(ls_i), c(ls_i) = \begin{cases} \Upsilon_{lb} & \text{if} \quad ls_i = 0 \\ \Upsilon_{ub} & \text{if} \quad ls_i = 1 \end{cases} \tag{1}$$

Next, MiRTrust looks up the sender of the security event in the local trust list. In case, the sender is listed within the trust list, the previous level of trust within the trust list and the current level of trust of the function p are compared. If | prev. l − curr. l | > 1, the sender is marked as suspicious. Otherwise, the past trust experiences v and the current trust value p are aggregated using the weighted average, as the trust experiences evolve over time. The trust level l is set to $l = \frac{v+p}{2}$. In case, the sender is not listed within the trust list and thus no aggregated trust experience tuple is available, collaborating partners are asked for recommendations r. As a recommendation is transferred over the network, it uses encryption to ensure confidentiality and a signature to prove the recommendation's authenticity. Thus, the current trust value p and the trust value of the recommendation r are aggregated depending on the quality q of the recommendation and determined as shown in Eq. (2). Only those recommendations are considered that provide a quality q greater than a minimum level of trust. In addition, only recommendations with a time stamp $t(p) > t(r)$ are used. The trust value rl takes into account the inherent knowledge uncertainty k of the given recommendation. T represents the time interval in which security events are observed and the total number of security events.

$$q_i = max\left(l_{min}, l_i \cdot k_i \cdot max\left(0, \frac{T - (t_n - t)}{T}\right)\right) \tag{2}$$

Due to the collaboration, multiple recommendations r_n are received. Therefore, a unique recommendation trust value rl is computed using a weighted average of the individual trust range of the recommendations with a quality greater than the minimum level of trust as shown in Formula (3) [15].

$$rl(r_1, \ldots, r_n) = \frac{1}{n} \sum_{i=1}^{n} l_i \cdot q_i | q_i > l_{min} \tag{3}$$

rl and the current trust value p are aggregated using the weighted average. The trust level l is set to $l = \frac{p+rl}{2}$.

Trust Dissemination: The trust formation is used to predict the trustworthiness of an ISP. In case no aggregated tuples are available, an ISP exchanges recommendations r with its collaborating partners to guarantee a minimum set of information to decide about the trustworthiness. As a consequence, recommendations contain sensitive data that require the use of security mechanisms (e.g., encryption & signature). Therefore, the exchange format FLEX is used to disseminate the recommendations.

Trust Evaluation: MiRTrust continuously updates its local settings during the occurrence of a security event and based on the received recommendations. These updates are included with equal weight to ensure that a trust opinion can not change rapidly (e.g., caused by false good recommendation). In case a recommendation r is received that conflicts with the own calculated trustworthiness, the level of trust of ISP x is not aggregated into previous trust opinions and ISP x is marked as suspicious. In case, several security events of ISP x occur with conflicting trustworthiness, the level of trust of ISP x tends to drop to -2, that represents distrust and thus identifies x as an ISP with a suspect behavior. As a consequence, recommendations of x will be disregarded.

5 Evaluation

In this Section, we describe the qualitative and quantitative evaluation of our trust model MiRTrust. First, we describe the characteristics of the evaluation criteria. Second, we introduce five evaluation criteria for our trust model. Further, we evaluate MiRTrust using multi-method-modeling, describe the setup of the testbed and present the test scenario of the trust model. Finally, we present and summarize the results of the evaluation.

5.1 Qualitative Evaluation Methodology

The trust model MiRTrust is evaluated based on the following five criteria: Ease of deployment, authorization, subjectivity, asymmetry and decentralization. These criteria were derived from the requirements described in Sect. 2.2.

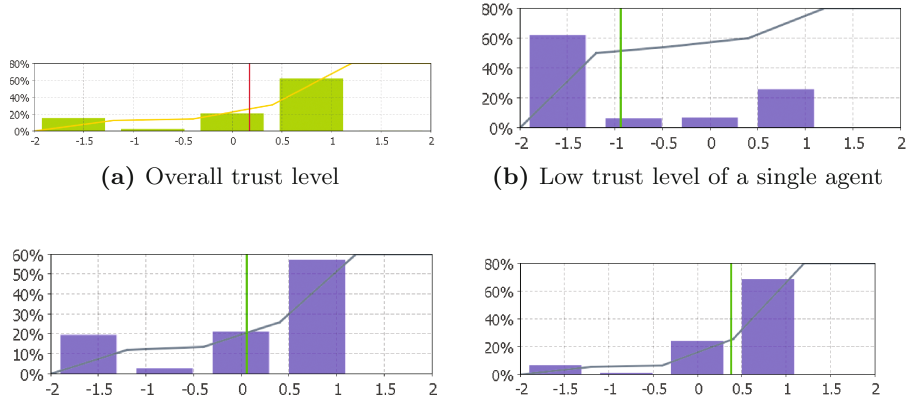
The criterion 'Ease of Deployment' describes the ability to use the trust model and its underlying implementation on different operating systems, infrastructure devices, exchange formats and protocols. The criterion 'authorization' refers to the ability to support the TLP protocol. The 'subjectivity' describes the possibility that network operators are able to form their own trust opinions. The criterion 'asymmetry' describes the possibility that two collaborating partners do not need to have the similar trust in each other. The criterion 'decentralization' refers to the ability that each participating ISP acts as a self-contained unit, calculates and stores its trust decisions locally.

5.2 Quantitative Evaluation Methodology

MiRTrust is evaluated using a multi-method-modeling approach consisting of an agent-based and a discrete event model using AnyLogic[5]. The model of MiRTrust is based on a scale-free network of ISPs that share security event information and perform mitigation actions based on the trust and knowledge level of each security event. The ISPs are modeled as agents. Each ISP has an individual behavior and attitude towards the trustworthiness of a sender of a security event. The process of mitigation is modeled in a discrete event way at each single ISP.

Initially, MiRTrust models a weekly contact rate to describe the assumption that 1 % of potential ISPs will want to join the MiRTrust community. Besides this contact rate, non-community members are able to join the community by using a sponsoring join process. Each community member possesses a list of directly connected collaborating partners and assigns an initial trust value range of $[0, 1]$ and a knowledge value of 0.5. Based on the findings in [30], each ISP receives 5 security events per month. The sender of the security event is set using a triangular distribution. This triangular distribution is used to create security events sent from an internal detection engine of an ISP, an external collaborating ISP and an unknown ISP. Based on the sender of the security event, an ISP starts its trust formulation calculation and looks up its past trust experiences w_1 with the sender of the security event. In case no trust experiences are available and thus no previous security events have been exchanged between the sender and receiver of the security event, the receiving ISP asks its collaborating partners to send recommendation tuples about the trustworthiness of the sender. Therefore, ISPs interact and share their trust experiences. Further, MiRTrust takes into account a local created trusted CA list w_2 and if this security event has also been seen by collaborating partners w_3. These local settings are weighted based on the formula (1) as follows: $w_1 = 0.5, w_2 = 0.15$ and $w_3 = 0.45$ with $w_1 + w_2 + w_3 = 1, 0 \leq w_i \leq 1$. Next, the trust level and the knowledge values are calculated. Finally, mitigation and response actions are deployed, if the trust level pass a threshold of 0. The duration of the mitigation process is set using a triangular distribution with lower limit $a = 20$ and upper limit $b = 1440$ min. These mitigation values are derived from [30]. Finally, the ISPs are waiting for the next occurring security event that restarts the trust formation process.

[5] The model can be downloaded on https://bitbucket.org/dasec/mirtrust.



(a) Overall trust level

(b) Low trust level of a single agent

(c) Medium trust level of a single agent

(d) High trust level of a single agent

Fig. 3. Distribution of trust levels

5.3 Evaluation Results

In this paragraph, we present and discuss the results of the qualitative and quantitative evaluation of MiRTrust.

Ease of Deployment: The heterogeneity of network devices and used operating systems requires a platform independent trust model that easily integrates within the existing infrastructure. Therefore, the implementation of MiRTrust is based on Java and thus can easily be deployed on different operating systems. Further, MiRTrust encodes its recommendation tuples in FLEX. The dissemination of those tuples among trusted partners is using STOMP and thus ensures platform independency.

Access Control: MiRTrust supports the semi-automated dissemination of security threat information based on the different level of trust. Therefore, MiRTrust differentiates between the following five different trust levels: distrust, unknown, none, marginal and full trust. The use of different trust levels allows to encode security event information using the TLP protocol and thus provide different detail of information within a security event tailored for its intended receivers. The trust level distrust, unknown and none trust are mapped to the color red of the TPL protocol. The color amber is used to encode the trust level margial trust and the color green is used to represent full trust.

Subjectivity: Through the different level of trust and sharing relationship between collaborating ISPs, MiRTrust supports that each collaborating partner is able to form its own trust opinion. The quantitative evaluation of MiRTrust shows the distribution of different level of trust in Fig. 3.

From	To	l	k	s	t
iSPs[359]	iSPs[11]	0.98	0.6	DDoS	2016-02-08 18:27:27.103+01
iSPs[289]	iSPs[11]	−1.5	0.1	DDoS	2016-02-08 18:27:29.571+01
iSPs[168]	iSPs[11]	0.83	0.6	DDoS	2016-02-08 18:27:30.253+01

(a) Trust levels of ISP 11

From	To	l	k	s	t
iSPs[11]	iSPs[359]	0.5	0.5	I	2016-02-08 18:27:27.457+01
iSPs[11]	iSPs[289]	0.5	0.5	I	2016-02-08 18:27:29.429+01
iSPs[11]	iSPs[168]	0.5	0.5	I	2016-02-08 18:27:32.089+01

(b) Trust of other ISPs in ISP 11

Fig. 4. Asymmetric trust level

Asymmetry: MiRTrust supports that two collaborating partners have different level of trust in each other as trust is not symmetric. If ISP a trusts ISP b, it does not follow ISP b trusts ISP a. Further, trust is not inherently transitive. If ISP a trusts ISP b and ISP b trusts ISP c, it does not automatically follow that ISP a trusts ISP c. Therefore, each single MiRTrust instance possesses a list of calculated trust level and knowledge values of each exchanged security event as shown in Fig. 4.

Decentralization: MiRTrust is deployed at each collaborating partner and thus acts as a self-contained unit that calculates and stores trust opinions locally. Further, the recommendation tuples are transfered to collaborating partners using FLEX and thus are signed. Each ISP is able to form its own trust opinion about collaborating partners similar to the principle of web of trust.

6 Conclusion

Nowadays, large-scale cyber attacks have become larger, more sophisticated and frequent. One approach to mitigate and respond to large-scale network-based attacks focuses on collaboration. In this paper, we introduced the trust model MiRTrust that facilitates the semi-automated assessment and deployment of remediation suggestions within a security event. MiRTrust is used to determine different levels of trust, set the prioritization of the shared security event, sanitize the occurrence of security events and contributes to built a trust community in order to share information about cyber threats and its remediation suggestions. We have shown that MiRTrust is able to support the formation of a subjective and asymmetric trust level and can be used to encode cyber threat information using TLP for dissemination.

Based on our qualitative and quantitative evaluation, MiRTrust constitutes a viable and collaborative approach to assess the trust level of collaborating ISPs and thus deploy semi-automated remediations of a security events.

Acknowledgment. The work has been funded by CASED and by EU FP7 Flamingo (ICT-318488).

References

1. Anstee, D., Bussiere, D., Sockrider, G., Morales, C.: Worldwide Infrastructure Security Report. Technical Report IX, Arbor Networks Inc., January 2014. http://www.arbornetworks.com/resources/annual-security-report
2. Research and Education Networking Information Sharing and Analysis Center: REN-ISAC Research and Education Networking Information Sharing and Analysis Center (2015). http://www.ren-isac.net/
3. Advanced Cyber Defence Centre: ACDC Deliverables (2015). http://acdc-project.eu/acdc-deliverables/
4. Reitinger, P.: Enabling Distributed Security in Cyberspace. U.S. Department of Homeland Security (2011). https://www.dhs.gov/xlibrary/assets/nppd-cyber-ecosystem-white-paper-03-23-2011.pdf
5. Internet Architecture Board, the Internet Society: CARIS Workshop Template Submissions. Internet Architecture Board and the Internet Society, June 2015. https://internetsociety2.wufoo.com/reports/caris-workshop-template-submissions/
6. Morrow, C., Dobbins, R.: DDoS Open Threat Signaling (DOTS) Working Group Operational Requirements. IETF 93, July 2015. https://www.ietf.org/proceedings/93/slides/slides-93-dots-3.pdf
7. The White House: Executive Order - Improving Critical Infrastructure Cybersecurity (2013). https://www.whitehouse.gov/the-press-office/2013/02/12/executive-order-improving-critical-infrastructure-cybersecurity
8. National Parliament of the Federal Republic of Germany: Gesetz zur Erhung der Sicherheit informationstechnischer Systeme (IT-Sicherheitsgesetz) (2015). http://dipbt.bundestag.de/extrakt/ba/WP18/643/64396.html
9. Danyliw, R., Meijer, J., Demchenko, Y.: The Incident Object Description Exchange Format RFC 5070 (Proposed Standard), December 2007
10. Debar, H., Curry, D., Feinstein, B.: The Intrusion Detection Message Exchange Format (IDMEF) RFC 4765 (Experimental), March 2007
11. Shafranovich, Y., Levine, J., Kucherawy, M.: An Extensible Format for Email Feedback Reports RFC 5965 (Proposed Standard), August 2010
12. abusix GmbH: x-arf Network Abuse Reporting 2.0. http://www.x-arf.org/
13. Steinberger, J., Sperotto, A., Baier, H., Pras, A.: Exchanging security events of flow-based intrusion detection systems at internet scale. Internet Architecture Board and the Internet Society, June 2015. https://www.iab.org/wp-content/IAB-uploads/2015/04/CARIS_2015_submission_3.pdf
14. Steinberger, J., Sperotto, A., Golling, M., Baier, H.: How to exchange security events? overview and evaluation of formats and protocols. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM 2015), May 2015
15. Abdul-Rahman, A., Hailes, S.: A distributed trust model. In: Proceedings of the 1997 Workshop on New Security Paradigms, NSPW 1997, pp. 48–60. ACM, New York (1997)
16. Jonczy, J., Wüthrich, M., Haenni, R.: A probabilistic trust model for GnuPG. In: 23C3, 23rd Chaos Communication Congress, Berlin, Germany, pp. 61–66 (2006)
17. Golling, M., Hofstede, R., Koch, R.: Towards multi-layered intrusion detection in high-speed networks. In: 2014 6th International Conference On Cyber Conflict, June 2014

18. Esposito, C., Ciampi, M.: On security in publish/subscribe services: a survey. IEEE Commun. Surv. Tutorials **17**(2), 966–997 (2015)
19. European Union Agency for Network, Information Security: Reputation-based Systems: a security analysis (2007). https://www.enisa.europa.eu/publications/archive/reputation-based-systems-a-security-analysis
20. Fullam, K.K., Klos, T.B., Muller, G., Sabater, J., Schlosser, A., Topol, Z., Barber, K.S., Rosenschein, J.S., Vercouter, L., Voss, M.: A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, pp. 512–518. ACM (2005)
21. The Department of Homeland Security's United States Computer Emergency Readiness Team: Traffic Light Protocol (TLP) Matrix and Frequently Asked Questions (2015). https://www.us-cert.gov/tlp
22. Grandison, T.: Trust management for internet applications. Ph.D. thesis, Imperial College of Science, Technology and Medicine University of London (2003). http://www.doc.ic.ac.uk/~tgrand/PhD_Thesis.pdf
23. Anti-Phishing Working Group: Charter and Saga - Unifying the global response to cybercrime though data exchange, research and public awareness (2015). http://apwg.org/
24. Messaging, Malware, Mobile Anti-Abuse Working Group: Member application (2015). https://www.m3aawg.org/
25. European Union Agency for Network, Information Security: CERT cooperation and its further facilitation by relevant stakeholders (2006). https://www.enisa.europa.eu/activities/cert/background/coop/files/cert-cooperation-and-its-further-facilitation-by-relevant-stakeholders
26. Capra, L.: Engineering human trust in mobile system collaborations. In: Proceedings of the 12th ACM SIGSOFT Twelfth International Symposium on Foundations of Software Engineering, SIGSOFT 2004/FSE-12, pp. 107–116. ACM, New York (2004)
27. The Free Software Foundation: Trust in a key's owner (1999). https://www.gnupg.org/gph/en/manual.html
28. General, European Commission's Directorate for Communications Networks, Content & Technology: EU Trusted Lists of Certification Service Providers (2014). https://ec.europa.eu/digital-agenda/en/eu-trusted-lists-certification-service-providers
29. Steinberger, J., Kuhnert, B., Sperotto, A., Baier, H., Pras, A.: Collaborative DDoS defense using flow-based security event information. In: 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), April 2016
30. Steinberger, J., Sperotto, A., Baier, H., Pras, A.: Collaborative attack mitigation and response: a survey. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM 2015), May 2015