cryptography

MDPI

Article

Security Incident Information Exchange for Cloud Service Provisioning Chains

Christian Frøystad [*,†], Inger Anne Tøndel [†] and Martin Gilje Jaatun [†]

SINTEF Digital, Postbox 4760 Torgarden, 7465 Trondheim, Norway; ingeranne.tondel@sintef.no (I.A.T.); martin.g.jaatun@sintef.no (M.G.J.)

* Correspondence: christian.froystad@sintef.no; Tel.: +47-45217066

† Current address: Strindvegen 4, 7034 Trondheim, Norway.

Received: 30 October 2018; Accepted: 5 December 2018; Published: 11 December 2018

Abstract: Online services are increasingly becoming a composition of different cloud services, making incident-handling difficult, as Cloud Service Providers (CSPs) with end-user customers need information from other providers about incidents that occur at upstream CSPs to inform their users. In this paper, we argue the need for commonly agreed-upon incident information exchanges between providers to improve accountability of CSPs, and present both such a format and a prototype implementing it. The solution can handle simple incident information natively as well as embed standard representation formats for incident-sharing, such as IODEF and STIX. Preliminary interviews show a desire for such a solution. The discussion considers both technical challenges and non-technical aspects related to improving the situation for incident response in cloud-computing scenarios. Our solution holds the potential of making incident-sharing more efficient.

Keywords: incident response; cloud computing; accountability

1. Introduction

Cloud computing comes with significant benefits. These benefits include increased agility and reliability; better scalability and elasticity; improved maintenance; device and location independence; and reduced cost [1]. Thus, it is understandable that the popularity of cloud infrastructures is rapidly rising both when considering small and larger companies.

Whereas the benefits of cloud computing are well known, there are drawbacks or challenges which need to be taken into account by stakeholders considering a move to the cloud. This paper focuses on incident response, which is the process of dealing with an incident from detection, through analysis, containment, eradication and recovery, and preparation [2]. These activities have increased in complexity since the time when servers were physical machines running a single system for one organization, possibly at their own physical premises. A set of security issues related to incident-handling in the cloud were examined by Grobauer & Schreck [2] back in 2010, who called for more research in several areas. In the years which have passed since, surprisingly little research addressing those challenges has been published [3]. Most of this research, however, is mainly concerned with digital forensics in the cloud, or more traditional incident-response scenarios.

Traditionally, exchanging incident information has been conducted based on personal trust relationships. The actual exchange of incident information normally happens by means of email, phone, incident trackers, help desk systems, conference calls, and face to face meetings. With the advent of cloud computing, however, the human element is much less prominent. A cloud service can be made up of a chain of providers where none of the Computer Security Incident Response Team (CSIRT) members have ever communicated directly with a representative from any of the other providers. In addition, an incident in the cloud may need to involve different parts of the provider chain, potentially even in an automated, real-time fashion, to minimize business disruption [4]. This poses new requirements to the way incident response needs to be managed and supported by tools which can communicate effectively across rapidly changing constellations of organizations.

An essential challenge is that there is no single part of the supply chain which has access to all events and all areas to monitor, thus nobody can immediately see the full picture. In a survey conducted by Torres [5], *little visibility* into system/endpoint configurations/vulnerabilities as such was considered one of the top hindrances to effective incident response in the participants' organizations. The cloud actors therefore need to be able to communicate efficiently to provide each other with information to ease detection or assist responding to an incident. It has also been claimed that attackers are better at handling information sharing than those protecting services and systems [6]. This adds to the importance of providing good tools and solutions to incident handlers [7].

> *"Our adversaries are amazingly coordinated. They do a far better job-sharing information than we do. it is becoming clear that the good guys need to find ways to share actionable information in real time to counter this threat."* [6]

The terms provider and subscriber are used in the following way throughout this paper:

- **Provider—**someone offering services to a cloud customer
- **Subscriber—**the cloud customer consuming services from the provider

A CSP can be both provider and subscriber, if it relies on services from other providers when offering their services to cloud customers.

In this paper, we extend our previous contribution [7] through the Incident Information Sharing Tool (IIST) tool and provide an overview of technical challenges (Section 2) and non-technical aspects (Section 3), that directly impact incident-response abilities for cloud-computing scenarios. The overall concept is introduced in Section 4 before going into details on the incident message format in Section 5. Moreover, the Application Programming Interface (API) for managing and exchanging incident notifications is presented in Section 6. A prototype implementation is presented in Section 7, while Section 8 presents the result of two focused interviews. An overall discussion of the approach is provided in Section 9, while Section 10 concludes the paper.

2. Functional Considerations

As a basis for our proposed specification, we have investigated functional aspects of how incident information should be represented and shared. The following provides an overview of how these aspects relate to cloud-computing scenarios in particular.

2.1. Notification

If an incident occurs at a CSP, he will eventually need to inform his customers. Customers includes both end users and other providers in the supply chain which relies on services from the CSP. Service Level Agreements (SLAs) and applicable law (e.g., the General Data Protection Regulation (GDPR) [8]) will regulate which notifications to give the end users. In [7] we discussed the different models for sharing incident information between provider and subscriber. These include a-priori agreements, provider decides it all and a hybrid where the subscriber may choose to receive incident information from categories predefined by the provider.

Figure 1 shows an example of the relationship between service providers and services used, and thus provides an example of the amount of unneeded or unwanted incident information a subscriber could receive if he is notified for all incidents in the entire service.

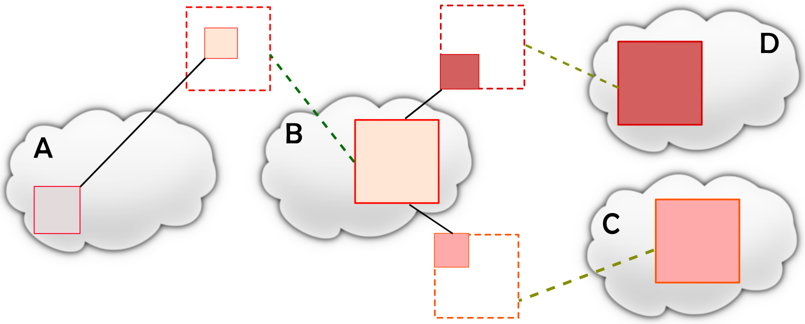


Figure 1. Data flow in cloud supply chain. Each cloud represents a service provider. Each colored square represents the services offered by the provider. The colored square inside each stippled square, represents the data or parts of the services actually used by the subscriber.

2.2. *Propagation of Incidents*

Incidents which affect one CSP could in turn affect other CSPs, and the incident message should hence be propagated. In [7], we discussed two approaches to representing propagated incident messages. First, we discarded including the entire parent incident message as leaking too much information down the chain.

An alternate approach would be to allow the incident message sender to be more in control of his incident information, by only referencing the parent incident when propagating down to a subscriber of a subscriber. In this way, only relevant information would propagate directly, through the actions of an entity, while there would still be a hard link to follow to establish exactly what happened during the incident-handling. This could, e.g., be useful when auditing a provider or during criminal investigations.

While referencing the parent incident message is a better solution than embedding the entire incident message, it still has noteworthy flaws. The referencing system first designed as part of IIST was leaking vital information about the value chain of a cloud provider. If a cloud provider received an incident notification from e.g., Amazon and notified their customers about the incident, all their customers would know that the provider uses Amazon for some part of their offering. Given enough time, a customer could map not only the external services used by a provider, but also their internal incident detection tools connected to the IIST since these would be the parent of some received incidents. Some providers would consider this trade secrets vital in differentiating their service offering from the competition. For this reason, the parent incident message is currently not referenced or included in the incident format.

The lack of information about the parent incident message does not affect the traceability of an incident, as the provider would need to store information about which incident message a subsequent incident message is derived from. This will still allow an auditor to follow the trail of an incident message, going from provider to provider collecting the relevant information. Furthermore, it removes the possibility to short-circuit the cloud service provisioning chain and maintains the proper communication channels. If company A buys a service from company B, they would have a contract in place. This contract would regulate the relationship between A and B, not A and C in the event B buys services from company C. Thus, even though the root cause of the incident originates at company C, the incident report to company A should be properly abstracted according to the service delivered and originate from company B.

2.3. *Security*

Security incident messages might contain vital information about a computer system and potentially personally identifiable information (PII). This makes it important to secure the incident information both in transit and at rest. This includes compliance with GDPR such as making sure to either anonymize any PII or avoiding its inclusion in the incident message. Given that incident information can be used to decide which changes to apply to a production system, it is also important to know that an incident message was received from the correct CSP. This makes it important that the subscriber has a way to validate the providers he subscribes to notifications from.

Transportation of incidents should only be allowed over a secure channel such as Transport Layer Security (TLS). To ensure that the only valid cloud customers or CSPs receive/provide incident information from/to another entity, authentication should be performed. While securing the incident information during incident exchange is important, the amount of information an attacker can obtain by eavesdropping is dependent on time—unless e.g., system credentials are transferred as part of an incident message. Hence, it is equally important to secure the incident information in the backend system, since an attacker would gain access to all incidents messages the entity has sent and received if having access to the incident system just once. The incident information must thus be protected using access control mechanisms and cryptographic protection in line with current good practice [9], the details of which are outside the scope of this article.

3. Non-Technical Aspects

For this solution to be useful, it needs to be adopted by businesses and CSPs. Given how most businesses strive to improve their financial results, it is likely that for the system to be adopted, one of the following criteria must be fulfilled:

- Use of the solution results in reduced costs or increased revenue—directly or indirectly
- Actors are required by law to use a system similar to this solution

A *Level 1* implementation, that is exchange of security incident information without any automation in incident-handling, is unlikely to result in significantly reduced costs, if reduced costs at all. However, the solution has been designed with implementation cost in mind, so the cost of adopting the solution should be quite low. The CSP could integrate the interface with their existing incident management tool and use an incident format adapter or translator to convert between their local format and the format used by the solution outlined in this document. If the solution was separated into microservices, one could further decrease the implementation cost by offering them as open source. The incremental nature of the solution allows the implementer to gradually introduce more formats and automation. As the implementation progresses into a *Level 2* implementation, with an increasing amount of automation, the reduced costs are expected to become noticeable. Metzger et al. [9] claim that more than 85% of abuse cases can be partly or fully automated, which in turn would free up resources allowing for reduced costs or for the incident handlers and the security team to focus more energy on improving security and handling the more difficult cases where full automation is not desired. Some incidents might require inspection and decisions to be made by a human before any information could be passed on to subscribers.

This solution is unlikely to contribute directly to a higher revenue stream, but might contribute indirectly e.g., by making it easier to notify affected parties and thus be more in compliance with the GDPR. If a CSP, or any other organization adopting this solution, is diligent in sharing information about incidents, this could contribute to building an image of trustworthiness and professionalism. Such an image could in turn result in more customers, and thus increased revenue. This would, however, require the organization to be careful to explain incidents and their process in an understandable manner, so the customer is reassured rather than unnecessarily alarmed. CSPs could even offer incident management dashboards for customers, so the customer would not need to administer their own instance of such a system. If the organization fails to appear trustworthy, it is likely to lose customers which in turn would affect the organization's revenue. It is, therefore, important to have competent incident handlers operating the system and any automation put in place, to ensure quality in both incident-handling and communication.

In [7], we discuss how more efficient incident response can result in financial benefits and improve the provider's reputation. Furthermore, there is a need to move the trust relationship from a personal level to an organizational level.

In [7], based on the findings of Bandyophyay et al. [10], it is argued that reporting an incident might increase its cost due to it becoming publicly known and causing secondary losses. Thus, the provider might be reluctant to share incident information. Additionally, the problem of having to share incident information before the incident is fully understood arises. The worst case would be to provide an attacker with information about how his attack is progressing. Technological solutions that provide secure means to distribute incident information can to some extent reduce the risk of sharing such information, as providers can have an improved overview of who has received what information regarding the incident. Terms regarding sharing and receiving incident information can also be covered in contracts.

Many other problems can be defined as sub-problems of trust. Legal worries about sharing incident information comes from the fear of legal action as a result of information sharing, but this can be traced back to not trusting how the recipient uses the received information. Public relations worries related to public perception of the company being damaged as a result of sharing information can also be traced back to lack of trust in how the recipient uses the information received. While our solution does not automatically make service providers trust each other, it could make a way for communication to happen in a flexible fashion between distinct and already known organizations, through a secure channel and in an environment controlled by the sender of incident information. It is expected that most non-technical problems, such as trust and who can forward what information to whom, can be solved by adding terms to the respective SLAs and adopt and enforce the Traffic Light Protocol (TLP) [11,12]. Use of sanctions for breach of contracts and trust might also assist in making it easier for organizations to share incident information, as they would know that any misbehavior would have consequences.

Laws are powerful incentives for changing behaviors in entire industries within a country. When large entities such as the United States or the European Union introduce similar laws, this affects the entire western world and, to some degree, the rest of the world [13]. Due to the substantial fines mandated by the GDPR [14], service providers are given an incentive to ensure accurate and timely notification about breaches relating to personal information.

Laws are not only an incentive, but sometimes also an obstacle. The difference between laws covering PII could complicate information exchange. The organization wishing to send incident information needs to make sure that no PII is included. It has been claimed that information disclosure is the biggest potential contributor to CSIRT liability ([15], p. 57). To mitigate the fear of being sued for sharing incident information, the US introduced a bill to protect companies that share information with the government from liability [16].

4. Concept

To allow cloud providers and cloud customers to exchange security incident information, we propose an agreed-upon subscription-based API following the Publish-Subscribe pattern [17]. An agreed-upon API, and potentially format, allows for increased automation of incident-response tasks, yet it does not require automation to be implemented anywhere. The goal is not to define how every aspect of the system should be design and implemented, but rather define the interactions and formats for exchanging information. The underlying system is left to the implementers of each incident-handling system.

5. Incident Format

We use the small base format presented in [7], with the ability to represent the most common information in a simple way, while providing a structured way of attaching other incident formats such as Incident Object Description Exchange Format (IODEF) and Structured Threat Information eXpression (STIX). In addition, the format supports custom fields, which allow the provider and the subscriber to agree upon extra information to be included in the base format without altering its base structure. This allows for three levels of implementation. If an implementation is at Level 1 (Figure 2), the base format would be the only implemented artifact, thus only being usable with humans as the primary actor.To reach Level 2 (Figure 3), various attachment formats would be added, allowing for some automation while reusing existing incident formats as well as remaining open to new formats. Level 3 (Figure 4) would be a fully automated incident information exchange tool.

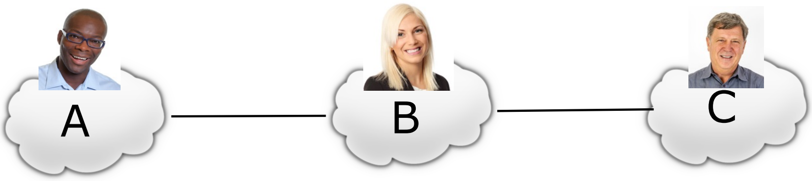


Figure 2. A level 1 implementation where humans handle all sharing of incident information.

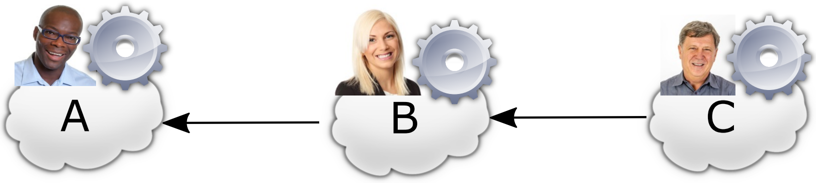


Figure 3. A level 2 implementation where humans and automated services share the burden of sharing incident information.

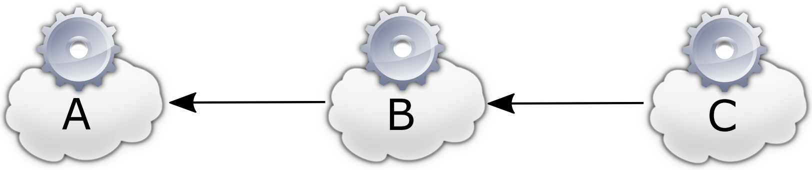


Figure 4. A level 3 implementation where automated services handle all sharing of incident information.

The fields included in the core format are inspired by IODEF [18], the Federal Incident Notification Guidelines [19], the EU Commission Regulation No 611/2013 [20], STIX [21], and a shared mental model for incident-response teams described by Flodeen et al. [22]. In Table 1 each data field is explained further.

5.1. Custom Fields and Attachments

Having a defined set of incident formats, agreed upon between the provider and subscriber through SLAs, simplifies the implementation as a cloud customer would only have to support the attachment formats agreed upon with the cloud provider, and the cloud provider only provides the formats agreed upon with its customers. Thus, not all cloud providers and customers have to support all incident formats, but our scheme still gives flexibility to the provider and the subscriber in identifying the formats most suitable for their use case and applying those. The attachments are not attached to the incident notification in the same way as email attachments are attached to an email, but is rather a reference to where the receiver can fetch the files if he desires.

Table 1. Explanation of the parameters in an incident. * required if parent parameter is included.

#	Parameter	Type	Description	Mandatory
1	id	UUID	Each incident message is assigned a Universally Unique Identifier (UUID), to not be confused with other incident messages from the current or other providers	yes
2	type	complex	Each incident is associated with an incident type which is further described in Table 2	yes
3	language	string	The language in which the incident is written	yes
4	status	string	The status of the incident, e.g., unresolved or resolved	yes
5	impact	float	An estimate of the consequence of the incident on the provider. The meaning of this estimate (represented as a float) is determined by the SLA	yes
6	summary	string	Short summary of the incident	yes
7	description	string	Free text description of the incident, allowing a more detailed description	yes
8	occurrence_time	ISO 8601	Estimated time of occurrence. This will be a best guess value until the investigation into the incident has progressed enough to pinpoint a more exact time	yes
9	detection_time	ISO 8601	The time the incident was detected	yes
10	liaison	complex	The person/entity to contact about this incident	yes
10.1	id	UUID	Each liaison is assigned a UUID, to not be confused with other liaisons from the current or other providers	yes
10.2	name	string	Full name of the liaison	yes
10.3	email	string	Email address on which the liaison can be contacted	yes
10.4	phone	string	Phone number on which the liaison can be contacted	yes
10.5	address	string	Address on which the liaison can be contacted/visited	yes
10.6	zip	string		yes
10.7	city	string		yes
11	attachments	complex[]	Array of files the receiver can fetch from the provider	no
11.1	format	string	The format in which the attachment is encoded	yes *
11.2	uri	url	The location at which the receiver can fetch the attachment	yes *
12	custom_fields	complex[]	Array of additional fields related to the incident type	no
12.1	id	UUID	Each custom field is assigned a UUID, to not be confused with other incident types from the current or other providers	yes *
12.2	value	string	The value of the custom field	yes *
12.3	type	complex	The type of custom field	yes *
12.3.1	id	UUID	Each custom field type is assigned a UUID, to not be confused with other custom field types from the current or other providers	yes *
12.3.2	name	string	The name of the custom field	yes *
12.3.3	description	string	Description of the custom field	yes *
12.3.4	type	string	Information used to interpret the value in 13.2	yes *
13	tlp	complex	Instructions from the provider to the receiving cloud customer on how the received incident information can be shared	no
13.1	schema	string	The schema used to interpret the TLP value, this is needed due to the difference between ENISA's TLP [12] and the TLP described by US-CERT [11]. Could be ENISA or US-CERT	yes *
13.2	value	string	The TLP value according to schema: Red, Amber, Green or White	yes *
13.3	fields	complex[]	Array of fields and accompanying TLP value. This allows specialization of TLP values per field. For any field not added specifically to the fields array, the main TLP value applies.	no
13.3.1	field	string	Name of the field to receive a custom TLP value	yes *
13.3.2	value	string	The TLP value of the field according to schema: Red, Amber, Green or White	yes *
13	next_update	ISO 8601	The time when the provider will send out a new update regarding the incident	no

Custom fields represent an easy way of exchanging a few extra values that the subscriber wants, but without the overhead of another incident representation format. There are multiple ways custom fields could be implemented, including allowing any values to be included in any format, allowing only a predefined set of basic data values, and providing data building blocks that allow representation of anything in a structured way.

Custom fields are related to incident types; thus, incident types can be viewed as a template for incidents by providing a customized set of fields to be completed by the incident handler. This reduces the mental burden of having to browse through large amounts of unrelated fields; only the fields needed for the incident type in question are made available to the incident handler.

6. Exchange Interface

The API is implemented through Representational State Transfer (REST), which acts as an *adapter pattern* [23]. Reducing coupling to a minimum increases flexibility, modifiability, and portability, and makes it easier to adopt the solution also for established systems and solutions. A list of all endpoints along with their HTTPS method can be found in [7]. The specifics of each method are presented later in this section, but Figure 5 provides a high-level overview of how the content of the methods relate.

Given a chain of CSP and cloud customers, a cloud customer can subscribe to incident information from a CSP. The CSP controls which kind of incident information each subscriber can subscribe to.

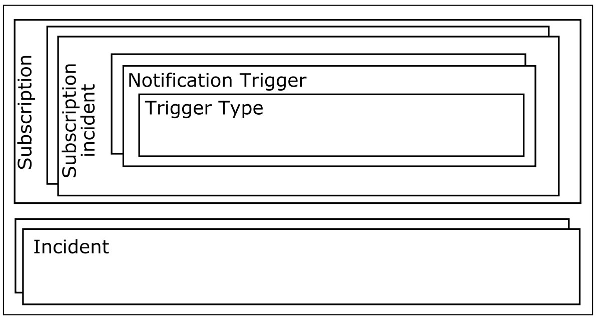


Figure 5. Overview of the content in a Notification.

6.1. Incident Type

This endpoint gives the subscriber an overview of the incident types available for notification. The incident types are specified by the provider, and the provider would need to have conducted an assessment on how and when incidents of each type can be shared. This is specified by the provided trigger types. The provider also needs to decide whether an incident of the given type can be automatically pushed to subscribers or if humans needs to manually approve the notification. If the incident handler decides not to send an incident, to which cloud customers have subscribed, this needs to be logged so that the necessary information will be available for an eventual audit of the provider. This could be achieved by using dedicated logging utilities such as Transparency Log as defined by Pulls et al. [24].

Table 2 describes the parameters and their types relating to the incident type payload.

Table 2. Incident Type parameters.

#	Parameter	Description
1	id	Each incident type is assigned a UUID, to not be confused with other incident types from the current or other providers
2	name	A short name for Incident Type, making it easier to talk about the incident type and get an indication to what it is about
3	description	The description is supposed to give the subscriber enough information to decide if this is an incident type he needs to subscribe to or not, thus the description should cover the important aspects of the type
4	consequence	Consequence is a value between 0 and 1, where the provider estimates the consequence of the incident occurring. Given that the provider does not necessarily know how the subscriber uses his system, the consequence will be more about how much damage or how deep into the system a perpetrator could come, rather than how large the consequence would be for the subscribers' system. This could become a way of identifying providers' weak spots to facilitate attacks, but it could also become an incentive for providers to ensure security at every corner. An alternative would be for the consequence values to be calculated based upon which services the subscriber uses and for which purpose he has stated the services are used

6.2. Trigger Type

This endpoint gives the subscriber an overview of the available types of triggers for the incident type in question. The provider creates trigger types matching its infrastructure and the conditions upon which it has assessed the need to share incident information. The cloud customer uses these trigger types to manage under which conditions he wants to receive notifications. This will help the provider stay in control of when different types of data are shared with subscribers, while still allowing the subscribers to choose when to receive notifications.

Table 3 describes the parameters and their types relating to the payload of a trigger type.

Table 3. Trigger Type parameters.

#	Parameter	Description
1	id	Each trigger type is assigned a UUID to be able to tell triggers apart, also between different providers
2	name	The name is a short description to make it easier to talk about and recognize the trigger type
3	description	A complete description of the trigger type must be provided for the subscriber to decide if the trigger is relevant for his use case and to make an informed choice about how and which threshold to set. The description also needs to include information about how the threshold is to be interpreted
4	comparators	Array of comparators (<, >, =, !=, etc.) available to triggers of this type. Comparators defines the comparators to use in relation to the threshold value. Even though threshold is restricted to have the type float, this allows for easy integration with most scenarios—if an adequate description is assisting the subscribers in interpreting the values, e.g., true and false values are translated to 1 and 0, system states are enumerated, etc.

6.3. Notification Trigger

Notification Triggers define when a cloud customer is to be notified. The subscriber creates notification triggers, which is instantiations of trigger types, for each incident type he subscribes to. A notification trigger consists of a trigger_type and a threshold.

Table 4 describes the parameters and their types related to the payload of a notification trigger.

Table 4. Notification Trigger parameters.

#	Parameter	Description
1	id	Each trigger is given a UUID to not be confused with other triggers, potentially from other providers
2	type	Every trigger needs to be of a type defined by the provider, as such the trigger needs to be assigned a trigger type. See Table 3 for details about trigger types
3	method	Describes the trigger's relation to the next trigger in the list. AND has precedence over OR
4	threshold	The value at which a notification is triggered
5	comparator	The operator that interprets the threshold and how the threshold relates to the value computed by the service provider

6.4. Subscription Incident

When the subscriber adds an incident type to his subscription, he creates a subscription incident. This is a combination of an incident type and incident triggers with the accompanying threshold values defined by the subscriber. A subscription can hold one or more subscription incidents.

Table 5 describes the parameters and their types related to the payload of the subscription incident.

Table 5. Subscription Incident parameters.

#	Parameter	Description
1	id	Each subscription incident is assigned a UUID to keep them apart
2	name	Name of the subscription
3	type	The incident type the cloud customer subscribes to. See Table 2 for details on the incident type
4	triggers	Array of triggers to define when the cloud customer wants to be notified. See Table 4 for details on the triggers

6.5. Subscription

A subscription is a set of incident types and triggers specified by the cloud customer. A subscription holds at least one incident type and one incident type holds at least one trigger, but each could hold an infinite number of incident types or trigger types. The triggers can be assigned methods on how they relate to the other triggers as AND, OR, or NONE. The trigger method operates on the current trigger and the next trigger in the list. The none operator can only be used for the last or only trigger in a list.

Incident → Trigger 1 AND Trigger 2 OR Trigger 3

AND has higher precedence than OR, so the above statement is interpreted as (Trigger 1 AND Trigger 2) OR Trigger 3.

Table 6 describes the parameters and their types related to the payload of a subscription.

Table 6. Subscription parameters.

#	Parameter	Description
1	id	Each notification is assigned a UUID to be interoperable with other systems
2	name	Name of the subscription
3	endpoints	The url to which a notification shall be sent
4	incidents	Array of incidents. See Table 5 for details on subscription incidents

6.6. Notification Validation

This is an endpoint the subscriber can use to validate that a received incident notification is correct and was actually sent by the claimed sender.

The payload is the notification the subscriber wants to validate. He receives either 200 OK or an error:

```
{
"error": ERRORCODE,
"error_msg": STRING_DESCRIBING_ERROR
}
```

6.7. Sent Incident Notification

When the provider pushes an incident notification to the endpoint defined by the subscriber, he uses the format described in Table 7.

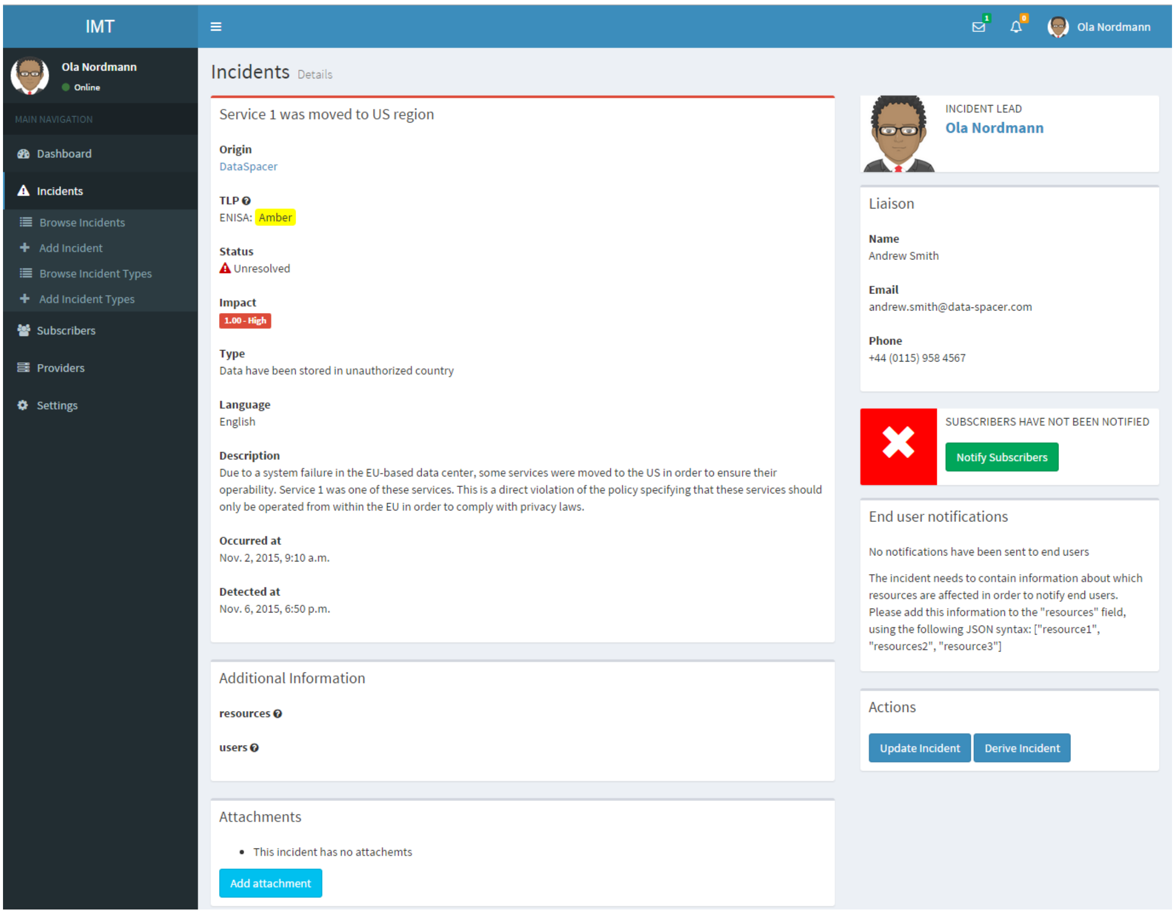
Table 7. Incident notification parameters.

#	Parameter	Description
1	id	Each notification sent, is assigned a UUID, so that the receiver has a way of easily separating notifications
2	type	The type refers to the notification type, created by the subscriber that initiated the notification. See Table 6 for details
3	generated	The time at which the notification was created
4	sent	The time at which the notification was sent
5	sender	The UUID of the sender which would be the provider in most cases. This allows the subscriber to validate the received notification by sending it, in its entirety, to the validation Uniform Resource Identifier (URI) at the provider. Only the sender id is provided, so the subscriber must consult his records of subscriptions to find the correct validation URI
6	hash	Computed hash value. This allows the message to be validated and authenticated
7	incidents	Array of incidents. The actual incidents sent to the cloud customer. See Table 1 for details on the incident format

7. Prototype

The customers of a cloud provider may have varying preferences when it comes to which systems, services, and incident types they are interested in notifications for, as well as which thresholds of severity these should operate in relation to. Supporting such individual preferences can be accommodated by publishers e.g., through offering customers a subscription mechanism. The individual provider will need to have the final say as to what information their customers are allowed to receive, regardless of preferences. To avoid data leakage and enforce the principle of least privilege, access to the API is only provided through a secure channel, over HTTPS. Both senders and receivers should be authenticated using established methods of authentication.

Figure 6 shows a screen-shot from a prototype built around the concept and specification outlined in this paper. The prototyped graphical user interface presents a minimal way to manage the basic part of the incident format which is meant to be easy to understand for human incident handlers. As a subscriber, the dashboard allows you to browse through the received incident notifications. Currently the functionality to subscribe to incidents is not implemented in the user interface, but supported through the API. As a provider, the prototype has functionality for defining possible incident types which others can subscribe to, as well as composing new incident notifications and updates related to these. For those who are both a subscriber and a provider, the interface contains functionality for deriving a new message from a received notification, which must retain internal links to the origin of the notification. The source code of the prototype can be obtained from our Github repository [25].



Incidents Details

Service 1 was moved to US region

Origin
DataSpacer

TLP
ENISA: Amber

Status
Unresolved

Impact
1.00 - High

Type
Data have been stored in unauthorized country

Language
English

Description
Due to a system failure in the EU-based data center, some services were moved to the US in order to ensure their operability. Service 1 was one of these services. This is a direct violation of the policy specifying that these services should only be operated from within the EU in order to comply with privacy laws.

Occurred at
Nov. 2, 2015, 9:10 a.m.

Detected at
Nov. 6, 2015, 6:50 p.m.

Additional Information

resources

users

Attachments

- This incident has no attachemts

INCIDENT LEAD
Ola Nordmann

Liaison

Name
Andrew Smith

Email
andrew.smith@data-spacer.com

Phone
+44 (0115) 958 4567

SUBSCRIBERS HAVE NOT BEEN NOTIFIED

End user notifications

No notifications have been sent to end users

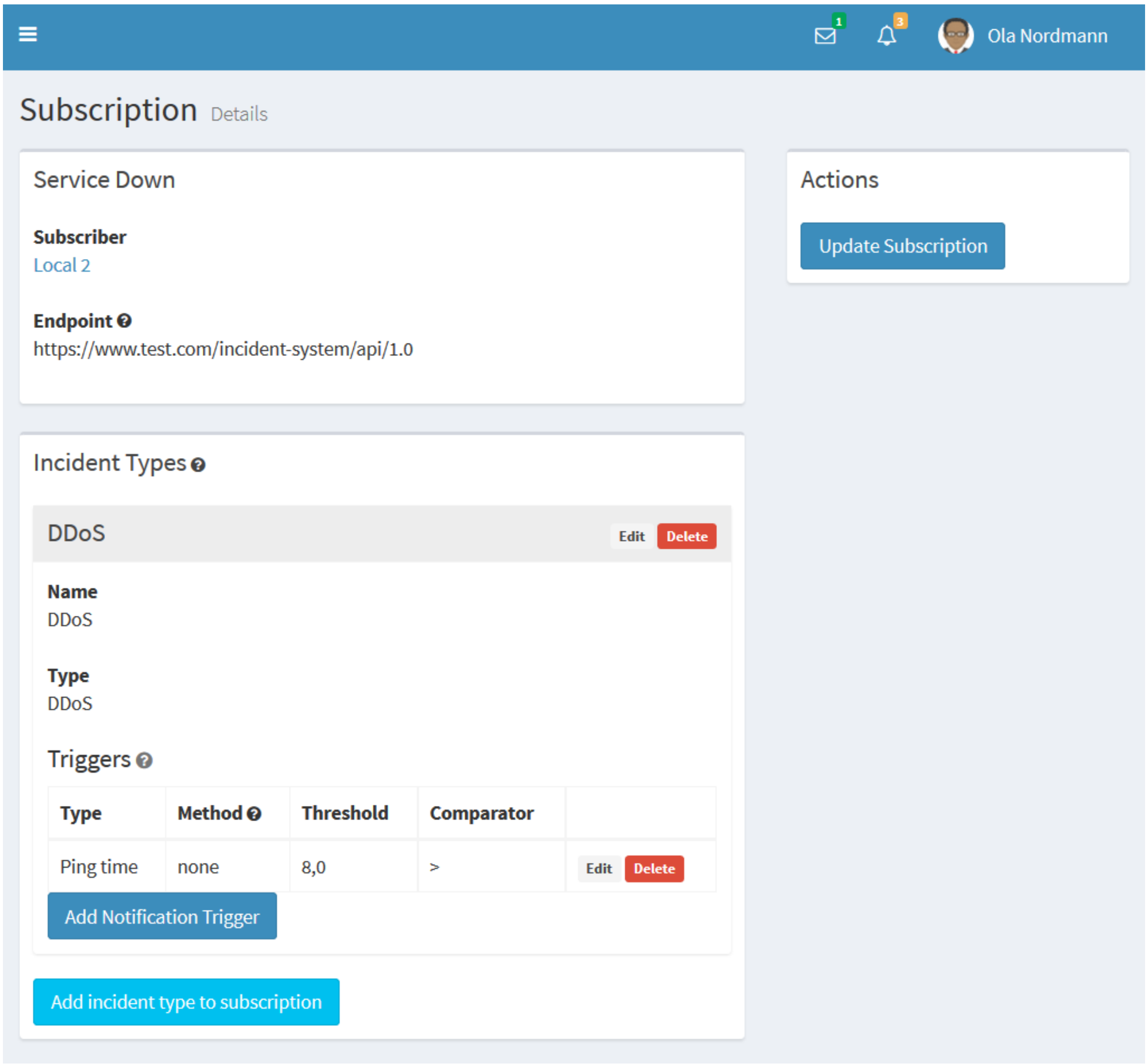
The incident needs to contain information about which resources are affected in order to notify end users. Please add this information to the "resources" field, using the following JSON syntax: ["resource1", "resources2", "resource3"]

Actions

Figure 6. Incident Details from the prototype built upon the proposed solution.

As the message format indicates, the user interface supports an indication of the message's origin, its status, impact, type, language, description, and timestamps for occurrence and detection time. In addition, there is information to add about the company's liaison with relation to this incident, and any custom fields which the incident type supports. The message composer also supports uploading attachments, allowing for extra data to be transferred in a deterministic and parsable way upon agreement between two parties. As mentioned earlier, standardized event exchange formats such as IODEF and STIX could be uploaded here for organizations wishing to represent their incidents in this level of detail. While incident notification theoretically could be fully automated, this particular prototype requires a human-in-the-loop to decide whether to notify subscribers or not. Attachments could increase the flexibility of incident information exchange while still maintaining a simple common base format. This will also contribute to reducing the coupling of information in incident reporting compared to having one large common format to represent everything.

Figure 7 gives an example of a simple notification subscription with the incident types and triggers the subscriber is interested in.



Subscription Details

Service Down

Subscriber
Local 2

Endpoint
https://www.test.com/incident-system/api/1.0

Incident Types

DDoS

Name
DDoS

Type
DDoS

Triggers

Type	Method	Threshold	Comparator	
Ping time	none	8,0	>	Edit Delete

Figure 7. A subscription with accompanying incidents and triggers.

8. Interviews

In connection with a demo of the dashboard prototype, we have conducted two focused interviews with experienced incident handlers from two organizations to evaluate our approach. Due to the early stage of the development, the number of participants was limited to two experienced incident handlers—the interviews were used as an initial test of the concept.

The interviewees noted that if most providers supported the exchange interface and the base incident format, it would be most useful in assisting notification to subscribers, customers, and users. There is also a potential added value in this increasing the amount and quality of the incident reports to the national Computer Emergency Response Teams (CERTs). Both interviewees wanted a simple way of requesting additional information regarding an incident. One of them suggested using a comment section visible to all recipients of the incident, thus giving the sender the option to reply once for every question rather than once for every recipient. Additionally, this could foster collaboration between receiving incident handlers to work around the issue before a final solution is released.

9. Discussion

Our main contribution is a simplified method of incident-sharing, making it available for organizations of all sizes, while still allowing for implementing a large ecosystem with automation if so desired. While the approach does not guarantee that all participants of the chain understand and correctly act upon the information, it gives all participants in the chain the possibility to receive any information they are eligible to receive by setting up subscriptions with its adjacent links. The approach also supports preserving traceability for incidents propagating through the Cloud Provision Chain. Finally, this facilitates the notification of end users, since the last provider in the chain is in the best position to know who to notify.

9.1. Adoption

It was strongly suggested to us, in response to presenting our approach at the A4Cloud [26] Advisory Board 2015, that the entire process should be completely automated and the incident handler or privacy officer should not have any say in whether incidents are provided to cloud customers or not—this would be a *level 3* implementation. Still, we have decided to make the prototype with a human-in-the-loop. Please note that there is nothing in the API or incident format that hinders removing the human from the loop. The reason for using an approach with a human-in-the-loop, boils down to incident handlers claiming that they would not adopt a system in which the customer and potentially end users were notified without them being allowed to ensure the correctness and quality of the notification [27]. The proponents of complete automation use the GDPR [20] and its 72 h notification rule as the main argument. The rationale is that if you have a chain of cloud service providers relying upon each other, a small delay in notification from each of them would soon amount to more than 72 h. This raises the question of who is responsible for the incident in the first place and when an incident is detected. Given a cloud service provider chain A-B-C-D and, for example, say A only has a relationship with B, B with C and C with D. As far as A is concerned, B is his only supplier and should therefore be responsible for any incident occurring further down the chain. Thus, A could sue B, but B could in turn sue C for not upholding his responsibilities. If this is the case, then the incident would be detected by A at the time of B revealing the incident, not when the root cause occurred at D. We have not been able to obtain a definitive answer to when the incident occurs for A in such a case, but our approach supports both full automation and human-in-the-loop. However, while full automation could help with timely notifications, it would require solid ground work to ensure not to violate other regulations by blindly passing on any information.

A *Level 1* implementation, i.e., exchange of security incident information without any automation in incident-handling as illustrated in Figure 2, is unlikely to result in significantly reduced costs. However, the solution has been designed with implementation cost in mind, so the cost of adopting the solution should be quite low. The incremental nature of the solution allows implementers to gradually introduce more formats and also automation. As the implementation progresses into a *Level 2* implementation, with an increasing amount of automation as illustrated in Figure 3, the reduced costs are expected to become noticeable. Metzger et al. [9] claim that more than 85% of abuse cases can be partly or fully automated, which in turn would free up resources allowing for reduced costs. Finally, a *Level 3* implementation would not need a human-in-the-loop but rely solely on automated services for sharing of incident information as illustrated in Figure 4.

"Even the smallest organizations need to be able to share incident information with peers and partners to deal with many incidents effectively" [28]. Large and small organizations are different in many ways, particularly in resources available to implement and adhere to new standards and systems. The solution presented here, was designed from the ground up to reduce the cost of initial adoption while facilitating added value as the complexity of the implementation increases—e.g., full automation.

9.2. Early Decision Making

Cichonski et al. [28] state that *"The incident-response team should discuss information sharing with the organization's public affairs office, legal department, and management before an incident occurs to establish policies and procedures regarding information sharing."* Our proposed solution facilitates such decisions to be taken before incidents occur, as subscribers can subscribe to incident types made available to them by the CSP. Thus, the CSP needs to have decided beforehand which incident types each subscriber can subscribe to. In addition, each organization is free to decide how to implement the backend and can thus require a human-in-the-loop, allowing for a second screening of incidents before they are sent to subscribers.

9.3. Prior Research and Industry Efforts

Cusick and Ma [29] describe a solution conceptually similar to the one proposed here, but for internal use in the organization. Users might subscribe to be notified when events are created or changed, which has led to improved communication around the incidents. The authors state that this feature alone made it worth their while to create a new process and implement new tools. Our proposal takes this one step further and allows other entities to be notified just as easily as the human subscribers.

Metzger et al. [9] describe a system that notifies subscribers about malware and other unwanted programs running on their systems, and is able to handle 85% of all incidents in a fully or partially automated manner. While our proposal does not provide any assistance with detection, it might be of assistance in notifying subscribers. It does also offer more fine-grained control over which notifications they wish to receive. The proposed solution could also be used as a communication channel for sensors submitting incidents to the central incident database. However, it would probably not replace email and phone as reporting channels for all users. For collaborating organizations, the solution could replace such means of communication, but for single individuals a simpler way of reporting would probably be needed, such as a web interface or email and phone as described in the article.

9.4. Further Work

There are multiple important challenges to tackle for sharing of security incident information to become common. However, further work should spring out of industry needs and evolve during use. Therefore, there is a need for a larger test and demonstration of the different approaches to incident and threat sharing to find the most practical approach for organizations of all sizes.

10. Conclusions

In this paper, we have presented a solution for propagating security incident information along Cloud Service Provisioning Chains. We have defined a format and an API which has been successfully tested in a lab environment. The solution would ease the propagation of incident information along the Cloud Service Provisioning Chain, which in turn can facilitate more accurate and timely information for CSIRTs and eventually improve overall security. Finally, the solution must be tested in a large real environment before more firm conclusions are made.

Author Contributions: Conceptualization, C.F. and M.G.J.; Methodology, C.F. and I.A.T.; Software, C.F.; Investigation, C.F.; Writing—Original Draft Preparation, C.F.; Writing—Review & Editing, M.G.J. and I.A.T.; Project Administration, M.G.J.; Funding Acquisition, M.G.J.

Funding: The research presented in this paper is partially funded by the European Commission's Seventh Framework Programme (FP7/2007–2013) under grant agreement no: 317550 A4Cloud—Accountability For Cloud and Other Future Internet Services.

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Kalloniatis, C.; Mouratidis, H.; Vassilis, M.; Islam, S.; Gritzalis, S.; Kavakli, E. Towards the design of secure and privacy-oriented information systems in the cloud: Identifying the major concepts. *Comput. Stand. Interfaces* **2014**, *36*, 759–775. [CrossRef]
2. Grobauer, B.; Schreck, T. Towards Incident Handling in the Cloud. In Proceedings of the 2010 ACM Workshop on Cloud Computing Security, Chicago, IL, USA, 4–8 October 2010; pp. 77–85. [CrossRef]
3. Jaatun, M.G.; Tøndel, I.A. How Much Cloud Can You Handle? In Proceedings of the 10th International Conference on Availability, Reliability and Security (ARES), Toulouse, France, 24–28 August 2015; pp. 467–473. [CrossRef]
4. Gjære, E.A.; Meland, P.H.; Vilarinho, T. Notification Support Infrastructure for Self-Adapting Composite Services. In Proceedings of the DEPEND 2014, The Seventh International Conference on Dependability, Lisbon, Portugal, 16–20 November 2014; pp. 17–24.
5. Torres, A. Incident Response: How to Fight Back A SANS Survey. Available online: http://westoninfosec.com/landing-pages/documents/enterprise-security/wp-sans-incident-response-fight-back.pdf (accessed on 7 December 2018).
6. Horne, B. On Computer Security Incident Response Teams. *Secur. Priv. IEEE* **2014**, *12*, 13–15. [CrossRef]
7. Frøystad, C.; Gjære, E.A.; Tøndel, I.A.; Jaatun, M.G. Security Incident Information Exchange for Cloud Services. In Proceedings of the International Conference on Internet of Things and Big Data, Rome, Italy, 23–25 Apri 2016; pp. 391–398. [CrossRef]
8. The European Parliament and the Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation) 2016. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32016R0679 (accessed on 6 December 2018).
9. Metzger, S.; Hommel, W.; Reiser, H. Integrated Security Incident Management—Concepts and Real-World Experiences. In Proceedings of the 2011 Sixth International Conference on IT Security Incident Management and IT Forensics, Stuttgart, Germany, 10–12 May 2011; pp. 107–121. [CrossRef]
10. Bandyopadhyay, T.; Mookerjee, V.S.; Rao, R.C. Why IT managers don't go for cyber-insurance products. *Commun. ACM* **2009**, *52*, 68–73. [CrossRef]
11. US-CERT. Traffic Light Protocol (TLP) Matrix and Frequently Asked Questions. Available online: https://www.us-cert.gov/tlp (accessed on 4 May 2015).
12. European Union Agency for Network and Information Security (ENISA). Information Disclosure. Available online: https://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/information-disclosure (accessed on 4 May 2015).
13. Greenleaf, G. The influence of European data privacy standards outside Europe: Implications for globalization of Convention 108. *Int. Data Priv. Law* **2012**, *2*, 68–92. [CrossRef]
14. Hon, K.; Kosta, E.; Millard, C.; Stefanatou, D. White Paper on the Proposed Data Protection Regulation. A4Cloud Deliverable D:B-5.1. Available online: http://cloudaccountability.eu/sites/default/files/D25.1%20White%20paper%20on%20new%20Data%20Protection%20Framework.pdf (accessed on 7 December 2018).
15. Brown, M.W.; Stikvoort, D.; Kossakowski, K.P.; Killcrece, G.; Ruefle, R.; Zajicek, M. *Handbook for Computer Security Incident Response Teams (CSIRTs) | SEI Digital Library*; Technical Report April; Software Engineering Institute: Pittsburgh, PA, USA, 2003.

16. Osborne, C. Threat-Sharing Cybersecurity Bill Unveiled—ZDNet. Available online: http://www.zdnet.com/article/threat-sharing-cybersecurity-bill-unveiled/ (accessed on 5 May 2015).
17. Uzunov, A.V. A survey of security solutions for distributed publish/subscribe systems. *Comput. Secur.* **2016**, *61*, 94–129. [CrossRef]
18. Danyliw, R.; Meijer, J.; Demchenko, Y. The Incident Object Description Exchange Format. Available online: http://www.ietf.org/rfc/rfc5070.txt (accessed on 6 December 2018).
19. US-CERT. Federal Incident Notification Guidelines. Available online: https://www.us-cert.gov/incident-notification-guidelines (accessed on 6 December 2018).
20. EuropeanUnion. Commission Regulation (EU) No 611/2013 of 24 June 2013 on the Measures Applicable to the Notification of Personal Data Breaches under Directive 2002/58/EC of the European Parliament and of the Council on Privacy and Electronic Communications. Available online: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2013.173.01.0002.01.ENG (accessed on 6 December 2018).
21. Barnum, S. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX$^{\text{TM}}$). Available online: http://stix.mitre.org/about/documents/STIX_Whitepaper_v1.0_(Draft).pdf (accessed on 6 December 2018).
22. Floodeen, R.; Haller, J.; Tjaden, B. Identifying a shared mental model among incident responders. In Proceedings of the 7th International Conference on IT Security Incident Management and IT Forensics, IMF 2013, Nuremberg, Germany, 12–14 March 2013; pp. 15–25. [CrossRef]
23. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Pearson Education: London, UK, 1994.
24. Pulls, T.; Peeters, R.; Wouters, K. Distributed Privacy-preserving Transparency Logging. In Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, Berlin, Germany, 4–8 November 2013; pp. 83–94. [CrossRef]
25. SINTEF-Infosec. Incident Information Sharing Tool. Available online: https://github.com/SINTEF-Infosec/Incident-Information-Sharing-Tool (accessed on 14 December 2015).
26. A4Cloud. Overview | Accountability for the Cloud. Available online: http://www.a4cloud.eu/ (accessed on 15 December 2015).
27. Frøystad, C. Exchange of Security Incident Information in the context of Cloud Services. Master's Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2015.
28. Cichonski, P.; Millar, T.; Grance, T.; Scarfone, K. *Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology, 800-61. Revision 2*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2012. [CrossRef]
29. Cusick, J.J.; Ma, G. Creating an ITIL inspired Incident Management approach: Roots, response, and results. In Proceedings of the Network Operations and Management Symposium Workshops (NOMS Wksps), Daejeon, Korea, 15–17 September 2010; pp. 142–148. [CrossRef]