Procure Secure

A guide to monitoring of security service levels in cloud contracts

About ENISA

The European Network and Information Security Agency (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Contact details

For contacting ENISA or for general enquiries on continuous monitoring, please use the following details:

Report editors: Dr Giles Hogben, Dr Marnix Dekker
E-mail: resilience@enisa.europa.eu
Internet: http://www.enisa.europa.eu

For questions related to continuous monitoring, please use the following details:
E-mail: resilience@enisa.europa.eu

Contributors

This paper was produced by ENISA editors using input and comments from a group selected for their expertise in the subject area, including industry, academic and government experts. The views expressed in this publication are those of the editors, unless stated otherwise, and do not necessarily reflect the opinions of the participating experts. We would like to thank the following for their contributions:

- Paolo Balboni, ICT Legal Consulting, Tilburg University, European Privacy Association
- Art Barnes, Dell Secureworks
- Matt Broda, Oneforo Corporation
- James Bryce Clark, OASIS
- Daniele Catteddu, Cloud Security Alliance
- George Chetcuti, Government of Malta
- Nick Coleman, IBM
- Dr. Peter Dickman, Google
- Dr. Niels Fallenbeck, Fraunhofer AISEC
- Julia Herman, European Aviation Safety Agency
- Brian Honan, BH Consulting
- Jens Jensen, Science and Technology Facilities Council, UK, Funded by EU Contrail Project
- Ben Katsumi, IPA, Japan
- Kieran McCorry, Hewlett Packard
- Mark Penny, UK Department of Health Informatics Directorate
- David Pollington, Microsoft
- James Reynolds, Left Coast Logic
- Dobromir Todorov, Amazon Web Services
- Dr. Nicolas Vanderavero, Federal Public Service Finance, Belgium
- Beau Woods, Dell Secureworks

We would also like to thank Kelly Ward, ENISA IT Department, for review and comment.

Table of Contents

Contributors 3
Executive summary 5
Who should read this document? 7
Related work 7
Introduction 9
Scope 10
Monitoring parameters 11
Parameter breakdown 13
Parameter groups 14
1. Service availability 14
2. Incident response 20
3. Service elasticity and load tolerance 24
4. Data lifecycle management 30
5. Technical compliance and vulnerability management 34
6. Change-management 40
7. Data isolation 42
8. Log management and forensics 44
Checklist guide to the document 48
How to use the checklist 48
How not to use the checklist: 48
Checklist 49
ANNEX Contractual considerations from ENISA 2009 Risk Assessment 58
Glossary of acronyms 59
Bibliography 60

Executive summary

Public procurement accounts for nearly 20% of the EU's gross domestic product- around 2.2 trillion Euro, according to Eurostat figures from 2009. Cloud computing is an area of growth in public procurement because of the substantial cost and efficiency savings cloud computing can offer. However, the use of effective SLAs (service level agreements) and common security requirements is one of the most important issues for the further adoption of cloud computing (1). ENISA therefore supports the European Commission's European Cloud Partnership initiative, with its focus on developing common requirements for public sector cloud procurement (2).

This document is a practical guide aimed at the procurement and governance of cloud services. The main focus is on the public sector, but much of the guide is also applicable to private sector procurement. This guide provides advice on questions to ask about the monitoring of security (including service availability and continuity). The goal is to improve public sector customer understanding of the security of cloud services and the potential indicators and methods which can be used to provide appropriate transparency during service delivery.

One-off or periodic provider assessments, such as ISO 2700x, SSAE 16 or ISAE 3402, assure that for the evaluation period, a certain set of controls and procedures was in place. These assessments are a vital component of effective security management. However, they are insufficient without additional feedback in the intervals between assessments: they do not provide real-time information, regular checkpoints or threshold based alerting, as covered in this report. The security monitoring framework is provided in the form of:

- **A Checklist guide to the document.** Use this if you have little time available- if you have read this, you will have covered the most important points. It is important to be aware that not all issues will be significant in all contexts; it is therefore strongly advised that you actively engage with the material and ensure you understand the extent to which each issue is relevant to your situation.
- **A detailed description** of each parameter which may be part of the security monitoring framework. This is the complete, unabridged version- it contains examples and looks at some of the more subtle points in more detail. It covers:
    - **What to measure.** Which security-relevant parameters of the service should be monitored throughout the contract.
    - **How to measure them.** How the data can be collected in practice.
    - **How to get independent measurements.** Which security relevant features of the service can be monitored independently from the provider and how.

- **When to raise the flag.** Considerations for setting reporting and alerting thresholds.
- **Customer responsibilities.** Whose problem is it? What needs to be taken care of by the customer on an on-going basis.

The parameters covered are (followed by some examples of issues explored):

1. **Service availability**
   - Which functions should be covered by availability monitoring?
   - How to define when a system is unavailable.
   - How availability is measured (e.g. user reports, sample requests).
2. **Incident response**
   - Definition of minimum response times.
   - Severity classification of incidents.
   - Incident management capabilities in place for systems customer control.
3. **Service elasticity and load tolerance**
   - For which resources should elasticity be monitored?
   - Elasticity tests (e.g. burst tests).
   - Elasticity in customer architectural choices.
4. **Data life-cycle management**
   - Monitoring of back-up operations and tests. e.g. age of most recent data restored.
   - Export test results: e.g. integrity check and parse according to well-defined formats.
   - Independent testing of availability and performance of back-ups.
5. **Technical compliance and vulnerability management**
   - Definition of a set of security-related configuration options.
   - Software updates and patches to be applied.
   - Procedures for vulnerability discovery and reporting including by a trusted third-party.
6. **Change management**
   - Notice periods for critical changes to system configuration.
   - Notification triggers implemented for critical events, such as loss of certification status (e.g. ISO), significant changes in security processes e.g. key lengths.
7. **Data isolation**
   - Types of data isolation monitored, e.g. memory, data at rest, secure deletion.
   - How to define criteria for a failure in performance isolation.
   - How can data and performance isolation be tested independently?
8. **Log management and forensics**
   - Are logs tested frequently for availability?
   - Cross-checks with customer's own event-logging systems (e.g. firewall logs).
   - Do you log relevant events in the systems under your control?

Who should read this document?

This guide is aimed at teams responsible for setting procurement requirements and ensuring their on-going fulfilment. This includes IT officers, IT security officers and procurement officers and service managers in the public sector. It should also be useful to C-level executives and legal departments, to gain an understanding of the customer-side security aspects of cloud or other outsourced IT services.

The focus is on public sector procurement, but much of the report will be equally applicable to private sector procurement. Although focused on cloud computing, much of this guide can be used also for other types of outsourced IT services, such as business process outsourcing.

NB: In reading this document, it is important to differentiate between small projects, in which the customer will simply make a choice between different types of service and their SLAs (service level agreements) offered on the market, and large projects, where the customer may be in a position to negotiate with providers about the kind of service or SLA required. This document focuses on public procurement and it is reasonable to expect that for some large procurement projects, public customers will be in a position to negotiate the SLA. However, even the largest public procurement projects may not justify customising some elements of the service or contract offered by the cloud provider: cloud computing offers elasticity and scalability benefits through the application of common requirements to a very large user base.

The goal of this document is to align the expectations of the public authority and Cloud Service Provider (CSP) on service/security monitoring requirements to expect and to provide in the market. Therefore, even for customers not in a position to negotiate contract terms, this guidance can serve as a basis for selecting between offerings on the market.

Related work

There is a large body of related work on the security and governance aspects of cloud computing. Some ENISA work related to this guide is:

- Cloud Computing: Benefits, Risks and Recommendations for Information Security (2009) (3), published by ENISA, covers the evaluation of security risks of migrating to the cloud, legal consideration (in an annex) and the ENISA Cloud Computing Information Assurance Framework (4).
- Security and Resilience in Governmental Clouds (2011) (5), published by ENISA, provides a guide for public bodies in the definition of their security and resilience requirements and how to evaluate and choose from the different cloud computing service delivery models.
- ENISA's survey of current practice in public procurement, covering over 140 public organisations across Europe (6).

This guide builds on the above ENISA reports. Previous ENISA reports focused on the security risks of adopting cloud services, and how to choose between the different types of cloud service in the market: this guide instead focuses on the parameters customers can monitor to assess the security of the service on a continuous basis.
There are numerous IT governance frameworks, such as ISO 2700x (7), SSAE 16 (8) or ISAE 3402 (9) that specify (high-level) security controls that should be deployed by the cloud service provider (CSP) to ensure the service is secure. There are also some control frameworks, such as CAMM (10), CSA CCM (11) , the ENISA Assurance framework (4), and ISO 27017 (12) (in development), which are tailored specifically to cloud computing services.

In the US, NIST has also published a number of white papers on cloud computing, for example NIST's Definition of Cloud Computing (13) is widely cited and used. NIST recently issued Special Publication 800-137 on Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations. SP800-137 (14) is a guideline for organising internal security processes focussed on maintaining on-going awareness of information security, vulnerabilities, and threats. We focus in this document on the customer, to allow the customer to continuously monitor security properties of the acquired service.

The Federal Risk and Authorization Management Program (FedRAMP) is the programme for federal government agencies to abide by in the procurement of cloud services. It provides a standardised and centralised approach to security assessment, authorisation and continuous monitoring for cloud-based services, and federal security requirements (e.g. FISMA). The programme not only sets security requirements, it also monitors the implementation of security measures, for example, by quarterly periodic vulnerability scan reports. FedRAMP may become a reference point for public contracts in the US. (15)

To assess compliance with FISMA, the US government has published a questionnaire for CIOs (16) with a specific focus on continuous monitoring. Although not going into much technical detail, the questionnaire puts an emphasis on continuous monitoring by the (government) customer.

For definitions of cloud computing terms, we refer the reader to NIST SP 800-145 (13).

Introduction

Procuring and managing service contracts for cloud services and other IT services is an increasingly important task for IT officers. It is important to specify security requirements upfront but it is even more important to be able to monitor and verify whether these security requirements are being met throughout the lifetime of the contract.

Previous ENISA reports have covered considerations for selecting cloud providers and the set-up and evaluation of key aspects of the contractual relationship (see Annex). But even with the best intentions, environments change and not all risks can be addressed pre-emptively. Both the CSP and the customer must be able to respond to changes in the threat environment on a continuous basis. It is essential to monitor the on-going implementation of security controls and the fulfilment of key security objectives. This is also described as a priority in the US government's 2010 report on the implementation of the federal information security management act (FISMA). The report notes a shift in strategy 'from periodic security reviews to continuously monitoring and remediating IT security vulnerabilities.' (17)

> The US government's 2010 report on the implementation of the Federal Information Security Management Act (FISMA) noted a shift in strategy 'from periodic security reviews to continuously monitoring and remediating IT security vulnerabilities.'

It is worth emphasising that certain security controls can only be implemented by the cloud customer. For example, in an IaaS (Infrastructure as a Service) service, the cloud provider is responsible for the underlying hardware, network infrastructure and connectivity, while the customer is responsible for the operating system, application platforms, software and data. In general, customers bear responsibility for elements of the end-to-end service delivery which are under their control, for communicating changing requirements to the CSP, and for co-ordinating between independent service components which have been composed into a service. This is discussed in more detail in the sections in this report on customer responsibilities (within each parameter), as well as in ENISA's 2009 report (3).

> This report builds on previous ENISA work, by giving guidance on how to monitor the security of a cloud service on an on-going basis.

This report builds on previous ENISA work, by giving guidance on ***how to monitor the security of a cloud service on an on-going basis***.

As a starting point, we surveyed over 140 public organisations across Europe to determine current practices in this area (6). The survey responses show that in most projects there is a one-off

assessment of the security of the service, but customers rarely seek or receive information about the security of the service on an on-going basis.

The document has been developed in consultation with a subset of 70 survey respondents who agreed to collaborate and provide feedback from a customer perspective, as well as industry experts who have provided input from a vendor perspective.

This document contains many examples and illustrations- each organisation's situation is different, however. Readers should create use-cases that apply to their needs, and think through how they relate to each section of the report (see [Checklist guide to the document], which provides a tool for supporting this process). We emphasise that any numbers quoted in examples are purely for illustrative purposes and should not be taken to reflect a given customer's needs.

Scope

The goal of this document is to give guidance to customers on continuous monitoring of security service levels and governance of outsourced cloud services. This is achieved through the reporting and alerting of key measurable parameters, as well as a clear understanding of how to manage the customer's own responsibilities for security. The scope is restricted to:

- **Security:** The parameters covered by this document are restricted to indicators of information security.
- **Cloud services:** The recommendations are developed for cloud procurement, but they may be applied to *certain* other outsourced IT projects such as storage, hosting services and even certain kinds of third-party IT service management offerings. In this document we refer to them as services or IT services. We note, however, that there are many types of outsourced IT services for which these considerations are less applicable (e.g. where the service provider's personnel are working on-site at the customer's premises, or where the customer formally 'owns' the service).
- **Public procurement:** The public sector represents a large share of the IT market, and public procurement is an important driver for cloud adoption and, more specifically, the adoption of more secure cloud services. Much of this document, however, applies equally to procurement in other sectors.

The full life-cycle of procurement of a cloud service, can be split into three phases:

- **Contract phase, e.g. request for proposal (RfP)**: Security requirements are defined (often by the customer agreeing to a set agreement). An important part of the agreements between provider and customer is the service level agreement (SLA).
- **Service delivery**: Service levels are monitored, using reports and alerts from the provider. The customer may also verify the service levels independently, by using appropriate tests or log

samples. The customer may also add their own triggers, alerts and response/escalation procedures based on the information from the provider.

- **Exit phase**: The customer ends the service and moves data to another provider or back in-house.

In this document **we focus on the service delivery phase and how to prepare for monitoring of security-related service levels in the RfP/contract phase**.

It is worth noting that some controls common in on-premise infrastructure solutions do not apply to cloud services. For example, secure disposal of electronic data on premise typically involves degaussing and/or shredding the media. In cloud environments, a public IaaS/PaaS/SaaS provider will dispose of the media at their end of life (in a similar secure fashion) but will typically not implement secure disposal on request by the customer; some CSPs however may offer this service function.

Monitoring parameters

This guidance comprises a set of parameters which can be used to provide more effective assurance and governance through monitoring the implementation and effectiveness of controls. The parameters listed are based on the results of consultation with the expert group and actual practice as captured by the survey (6). As input we used the ENISA report on Resilience Metrics and Measurements (18) which provides an overview of literature on security metrics and categorises existing security metrics.

We aim to describe the most relevant parameters which can be practically monitored, but we do not claim an exhaustive list. **Parameters should be selected according to the use-case** (e.g. IaaS, PaaS and SaaS have different monitoring requirements and/or division of responsibilities). Parameters should also be selected based on an analysis of an organisation's principal areas of risk and the impact that the IT service will have on these. Where the aspects of the IT service significantly impact an organisation's risk or its mission, the corresponding monitoring parameters should become the focus of SLA monitoring activity. **Considerations and sub-parameters are listed in order of importance.** Guidance on how to adapt the parameters to your use-case and, in particular your risk-profile is given in the subsections on Risk Profile Considerations and in the check-list guide.

Where the aspects of the IT service significantly impact an organisation's risk or its mission, the corresponding monitoring parameters should become the focus of SLA monitoring activity.

At RfP production stage, required parameters should, where flexibility exists, be prioritised, e.g. using the MoSCoW (Must, Should, Could, Would) scheme, or IETF RFC2119[1] (19), to determine those which are most crucial and which vendors can support them. The choice of parameters may discount specific vendors or types of service offering from consideration.

Note that while some metrics frameworks emphasise reporting mean values, in general, we have chosen not to recommend the use of mean values in several cases, since the use of mean values makes it more difficult to detect and deal with statistically rare but high impact events, which may be the most important indicators of insecure behaviour. For example, the mean value of CVSS vulnerability scores for a software provider may be 5/10, but they may have a single vulnerability which allows remote tampering of critical functions. In cases where this is an issue, un-aggregated data should be reported or where mean is used, standard deviation can be a useful additional indicator of the prevalence of rare but high impact events.

Reporting for each parameter can be divided into three different categories, depending on the time-criticality of the information:

- Real-time service level data/feeds, including service level dashboards.
- Regular service level reports.
- Incident reports raised by the CSP.

NB: Some reports contain confidential information and appropriate measures should be taken to protect confidentiality in transmission and storage (e.g. encrypt reports in transit and at rest).

Remark: Continuous monitoring can generate a very large volume of data and alerts. Only a subset of the parameters listed will be relevant to your specific project. Customers should use a risk management approach (including the risk profile considerations provided in this report) to ensure that their reporting framework prioritises information and alerts relevant to their situation. The monitoring framework should prioritise the information included in real-time alerts according to the greatest risks, and implement appropriate customer-side automated processing and escalation procedures.

[1] *MoSCoW and RFC2119 are alternative schemes for Note that each scheme includes different terms and interpretations (e.g. RFC2119 includes SHALL NOT, MUST NOT, SHOULD NOT clauses)*

Parameter breakdown

In order to support cloud customers in setting up a clearly defined and practical monitoring framework, which is appropriate to their risk profile, each parameter is broken down as follows. For all parameters, the provider should clearly communicate the provided service level to customers and also be able to show that there is appropriate internal oversight of the provision, e.g. via third-party audit:

- **Parameter definition**: **What is being measured?** The emphasis is on helping cloud customers select or draft service requirements which:
    a. Are clearly defined
    b. Can be verified in practice
    c. Are clearly understood by both parties in the case of a dispute (which may not even arise if terms are defined clearly enough).

  For example, we explain how availability can be defined in terms of basic service functions and their expected operation, in order to clarify the notion of 'up-time'.
- **Risk profile considerations**: **should I care about this parameter?** Guidance on how your organisation's risk profile and the risk profile of the overall service offering should determine the selection of sub-parameters and alerting thresholds. Example: a service for batch-processing non-personal scientific data only needs to be available for one hour every night when experiments are run, and is backed up by an alternative service. Thresholds for average availability could be set very low for such a service.
- **Monitoring and testing methodologies: how to measure it.** Methods and techniques for measuring parameters in practice. This includes techniques for obtaining objective measurements. For example, availability can be measured using regular sample requests, or by passive monitoring of logs.
- **Considerations for customer/independent testing: how to get trustworthy measurements.** Some monitoring techniques may be easily and economically carried out by the customer themselves or an independent party on behalf of the customer. Others can only be implemented on a system-wide basis by the service provider (or are too expensive for a customer to implement). For example, availability and business continuity can be tested independently but load-tolerance cannot (this is usually a function of the overall load of all customers).
- **Thresholds: when to raise a flag.** How to determine the ranges of parameters that would trigger an incident report, or response and remediation based on real-time or regular service level reports. For example, for elasticity and load tolerance, a trigger point can be set on the number of resource provisioning failures reported according to the risk profile of the customer's service.
- **Customer responsibilities: whose problem is it?** Outsourcing to a cloud provider can never relieve the customer of all responsibility for security. This section contains considerations for the relevant parameter on which aspects should be taken care of by the customer.

Parameter groups

1. Service availability

Availability and continuity requirements have often already been addressed in the RfP phase. The service agreement should be clear about what is meant by availability and how it will be monitored.

Parameter definition:
CSPs will express things differently- it is important to understand how availability is defined in the service level agreement and if this is adequate. The starting point for a definition of availability is typically a **target percentage of total operational time or requests**, for which a service should be considered available over a given period (typically a month or a year). For this to be meaningful, the SLA should clearly define **when a service is considered available**. There are several ways of achieving this, depending on what kind of service is measured. The target percentage should also be supplemented by a clarification of what is meant by:

- **Service request**: what functions of the service are included in the measurement? For example, an IaaS provider might define a service request to include a set of possible operations on a virtual machine instance.
- **Failure of a service request**: when are those functions considered non-operational? Are there standardised criteria or tests which can be applied to establish whether a request has failed? For example, when an http request to a web service provided by a SaaS service is not responded to within a given time (and there is evidence that this was due to a CSP fault and not a fault between the CSP and the end-point).
- **Sample size:** what is the time period or number of requests to which the availability criterion is applied? The 'sample size' is the time or number of requests over which a percentage of failures should persist before the system is classified as unavailable. If the sample size is too low, the measure is not statistically significant (e.g. a single failed request could be reported as unavailability). If it is too high, then extreme events will not be highlighted.
- **The scope of the service:** for example, does this apply to requests from a single customer, service-wide user requests, requests from a specific geographical region, etc.? Does the service cover end-to-end fulfilment of requests, or only as far as the nearest Internet connection point?
- **Commitment period:** does the SLA commit to an average percentage availability over one year, one month, one week etc.? The commitment period may also be broken down into different sub-periods, such as 'during weekends', 'during office hours' etc... The periods may also be 'sliding window'- i.e. a commitment to an average availability for any given one month/week/year period or over fixed periods.

Additionally, an SLA may define a **recovery time objective (RTO),** which is measured against **mean recovery time (MRT).** Some use-cases have strong requirements for the service to recover quickly

from downtime. For example, if a customer is processing financial transactions, they might state a requirement for the service to be down for no longer than 1 continuous minute. In this case, the CSP would report the performance against this objective in periodic reports, by either reporting MRT or the percentage of availability incidents resolved within RTO.

In applying RTO metrics, it is very important to understand exactly what recovery applies to and how it relates to service delivery. In an IaaS environment, the provider is likely to provide recovery time objectives for system components, rather than at the overall system level. For example, if an IaaS provider specifies an RTO for storage volume availability, it could be relatively long, on the assumption that the customer will be using multiple redundant volumes in a RAID configuration and can use this to offer a much more resilient service to its own customers. On the other hand, if the same provider specifies an RTO for the storage volume provisioning system, this might be relatively shorter because the customer cannot apply appropriate elasticity measures when the storage volume provisioning system is down. An IaaS provider would typically design a service with the expectation that the customer will build a resilient system using less resilient components.

In applying RTO metrics, it is very important to understand exactly what recovery applies to and how it relates to service delivery.

Finally, an SLA may define MTBF (mean time between failures), which can be useful in the case where long periods of uninterrupted operation are critical, for example, when processing a job which must be processed in a single transaction and will have to restart if the process fails at any point. In practice, large systems should engineer around such failures, e.g. using intermediate recovery points or partial replay.

Risk profile considerations:
Availability requirements should be set according to the criticality of a service. The customer should perform a risk assessment to determine availability and/or performance requirements. Factors to consider are:

- How critical is it for the service to be available in certain time intervals (e.g. during business hours, during weekends)?
- How critical is it for the service to recover quickly from any given outage?
- Is usage of the application predictable/planned or unpredictable, which would require high availability, leaving hardly any room for planned down-time?
- How critical is it for the service to be available for a given percentage of time (as opposed to available for more than a certain period at a stretch)? A customer may have availability requirements about the overall availability of a system over a period of time.

NB. Any numbers used are purely for illustration and should not be taken as a recommendation of any kind.

Example 1: A customer requires the service to be available for a minimum proportion of a given month (e.g. for non-time-critical processing of a large data set) and does not have strong requirements for the service to be up at any specific times in the month. They therefore prefer an SLA which specifies total uptime in a month.

Example 2: A customer processes financial transactions using a SaaS service. In case of system downtime, they have an independent queuing service which is able to store unprocessed transactions for an average of 2 minutes. They therefore look for an SLA which specifies a recovery time objective of less than 1 minute. A monthly report offered by the selected provider specifies mean recovery times.

Example 3: A customer has availability requirements of 99.99% for office hours in their time-zone, 0 for out of office hours. (99.99% corresponds to approx. 50 minutes of downtime per year, thus it is quite stringent.)

Monitoring methodology:

As mentioned above, the definition of when the service is unavailable can be based on a number of criteria. The following are different methods to monitor service availability:

- **Relying on users:** Depending on the setting, users may notice unavailability and report via a customer call or web form.
- **Relying on CSP logs:** Examination of logs by the provider, to detect errors. The provider continuously monitors logs, and may set triggers to respond to periods of unavailability.
- **By running sample requests/service health-check:** The service is polled using predefined sample requests, *simulating normal usage* (excluding load). A number of services and (open source) products exist for monitoring network connectivity and system up-time.
- **Relying on CSP monitoring tools:** CSPs may have deployed tools for monitoring systems and services.

Example 4: As part of a disaster recovery plan, a government provisions a number of virtual servers from an IaaS provider to maintain public services. It maintains them on continuous 'warm' stand-by so that they can be used at short notice when required. By agreement with the CSP, the servers are polled every 10 minutes by an automated system. If any of the servers do not respond, the customer administrator is notified by SMS.

Example 5: A customer with real-time messaging requirements decides to define a set of 'standard' messages which will be sent on a regular basis (e.g. every hour) together with an expectation of the responses ('known good responses') that will be received to those messages. The messages defined are used to test:

(a) That a secured connection can be made from the requesting system to the receiving system (e.g. mutually authenticated TLS (Transport Layer Security)).
(b) That a message can be sent from the requestor and received by the receiver in a pre-defined period of time
(c) That the message can be processed by the receiver in a defined period of time and a response sent to the requestor in a defined period of time
(d) That the response received is a 'known good response' for the message sent.

Any failure in any aspect of the above four criteria can be used to show that there is a problem with availability in relation to an aspect of the provided service, e.g. if (a) fails, depending on the reason for failure this could show either that the receiver's service is unavailable, or its End Entity digital certificate has expired or been revoked.

Relying on customers and logs for monitoring service availability is only suitable for service functions that are being used frequently. In such a setting, availability issues will be noticed quickly by users and reports will provide a statistically meaningful indication of service levels[2]. When managing services that are less frequently used or which have peak usage periods (e.g. once a year for tax returns), and in particular when these services are critical, then it becomes necessary to test the availability of the service proactively.

If it is critical for a service to be available all the time, then it should be monitored continuously, whereas for a service used only for short periods without time-criticality, we are more interested in the number of successful requests as a fraction of the total sample size. In both cases, the availability is a percentage between 0% and 100% which is a statistical measurement of service availability.

[2] *NB- Individual user reports may not indicate a problem with the cloud provider since the failure could be related to the user's equipment.*

Considerations for customer/independent monitoring:
Customers or third parties may monitor the availability of a service from the point of view of the end-user. The customer and provider should agree on the volume and frequency of tests, as well as how monitoring can yield a measure of availability.

Care should be taken in designing independent tests, to reflect normal usage (except loading) and not to trigger anti-DDoS or CAPTCHA systems. Cooperation with the CSP on these aspects is of paramount importance.

Care should be taken in designing independent tests, to reflect normal usage (except loading) and not to trigger anti-DDoS or CAPTCHA systems.

Incident and alerting thresholds:
Unavailability is typically raised as an incident (see also the Incident response parameter). This may depend on the duration of the outage, the set of functions affected, the number of users affected, and whether or not it was part of planned maintenance.

Customer responsibilities:
Customer responsibilities include:

- Designing the appropriate amount of resilience into system architecture over which they have control (e.g. in IaaS environments, using load balancing, geographical redundancy of availability zones, auto-scaling etc.).
- Testing the resilience of systems under their control e.g. using with load testing or 'chaos monkey' approach (deliberately removing random resources, within design tolerance, to test resilience- where permitted by the CSP).
- Dependencies: The customer should take into account all subsystems and infrastructure that may affect the service availability (not just the CSP's- see example 5 above)
- Thresholds and monitoring: If using sample requests, the customer and the provider should have a common understanding of how monitoring results translate to a measure of availability. They should also agree on the volume and frequency of tests involved in monitoring.

Worked examples:

[*Square brackets refer to the bulleted considerations in parameter definitions*]

Example 6: For a customer using an IaaS provider to run web servers, if the percentage of all http requests [*request definition*] to servers for all the provider's customers, as registered in logs shared by the provider [*scope of the service*], with a specified error status, is more than 5% [*failure of a service request*] for at least 5 minutes [*sample size*], then the server is considered to be unavailable (in those 5 minutes). This assumes that the provider is willing to publish the availability figures recorded in their logs.

Example 7: For a SaaS email service, the service is considered unavailable (or down) [*failure of a service request*] when 5% of all the service's users [*scope of the service*] are unable to send or receive messages [*request definition*] over a period of 5 minutes [*sample size*].

The (average) availability of the email service is then defined as the percentage of the total operational time the service was supposed to be available (the commitment period). In June, the service was down three times for 5 hours. The average availability over the full month [*commitment period*] was therefore 98%. The CSP offers the customer a minimum average monthly availability of 99.99% [*percentage of operational time*], therefore a service credit may be claimed.

Example 8: A customer procuring a SaaS office suite wants to have no more than 10 minutes of total downtime during office hours each month, and outside office hours at most 2 hours. The SLA they select specifies that the back-end server should be 99.9% available during office hours (9 to 6), and 99.8% overall [*commitment period*] and that the CSP will provide monthly reports about the obtained availability.

Example 9: A research centre uses a specialised scientific analysis service run as a SaaS service, to analyse DNA data of plant samples. The service works on 1GB batches of data sent by the customer and processes around 1000 batches a month. The SLA states that over a given month [*commitment period*] at least 99% [*percentage of operational requests*] of well-formed data batches should be processed [*request definition*] within 1 hour without returning an error code [*failure of a service request*].

It is important to take into account dependencies, especially in a cloud computing scenario, where services may depend on a variety of other subsystems and connections. In some settings a dependency may not be covered by the contract/SLA with the cloud provider, for example a WAN connection hired from a telecoms company. In such cases it is important to have enough information to determine who is responsible for a certain outage.

Example 10: The average availability commitment of SaaS office software requiring network connectivity is 97% and that of the network connecting the end-users to the server is 97%. The end-to-end availability across the chain, could therefore be as low as 94% (97% times 97%). It should be clear to the customer whether the availability commitment covers only the backend or the network as well [*scope*].

In this case, the customer asks the CSP to report on the availability of the backend server (a service level), and to monitor the end-to-end availability for users of the service and the network connectivity between the customer's site and the CSP's data centre. These are not related to the provider's service levels but are nevertheless important parameters for the customer (the provider may have to use client-side software to achieve this).

Finally, it is important to define how periods of unavailability count towards the fulfilment of a contract or SLA. When calculating availability, unscheduled downtime, and scheduled downtime are distinguished. The 'total uptime' over a month is 1 month minus the scheduled downtime. Periods over which availability is calculated can be days, weeks, or months.

Example 11: A contract may specify the following: Availability achieved = (operational_days - total downtime) / (operational_days - scheduled downtime)

2. Incident response

An incident is any event which is not part of the normal operation of the service and which causes, or may cause, an interruption or a reduction in the quality of the service, as defined by the SLA (20). This group of parameters relates to how the CSP responds to and recovers from incidents.

Incident response is **horizontal to all other parameters** since incidents and reporting thresholds are defined in terms of other parameters included in the SLA. For example, an incident can be raised when availability falls below 99.99% for 90% of users for 1 month, when elasticity tests fail or when a vulnerability of a given severity is detected. This group of parameters deals with the provider's response to incidents, independently of their nature.

Parameter definitions:
Following the ITIL model (21) (except that we do not use mean values), the service level of a provider's detection and response to incidents is typically defined in terms of:

- Severity: should be classified according to a well-defined scheme. It is unlikely objectives will be set for severity, since this is not under the control of the CSP. However, this information is required to understand the efficacy of the response procedures.
- Time to respond (from notification/alerting): the time to implement a remedial response.

Time to recovery is often not specified: an incident should be remediated as soon as possible.

Depending on the setting, the following indicators may also be provided as part of continuous monitoring and/or regular reporting of incident management.

- Percentage of incidents of a given severity resolved within a defined period. For example, 90% of severity 2 incidents resolved within 4 hours. Note that:
    - Criteria for resolution should be clearly defined, e.g. in terms of system functionality and/or controls applied. For availability incidents, the resolution period corresponds to time to recover- for other incidents, such as data loss, resolution will be defined differently.
    - Time to resolve may also depend on the availability of customer resources (for troubleshooting information).
- Aside from incident resolution, it is also important that the provider reports on the recovery process and expected time to recover. For example, the provider might undertake to give an update and revised estimated resolution time every 2 hours. This is usually conditional on recovery exceeding the defined recovery period. Service providers typically make such information available on their Web site rather than communicating individually with every customer.
- Time to report (for incidents which are required to be reported in real-time by the CSP to the customer): the time lapse between an incident occurrence and an incident report to the customer. This is only applicable for incidents where the time of occurrence can be verified independently of the reporting time (e.g. downtime verified by user reports) and it can be verified that the incident falls under the CSP's reporting obligations.
- Time since last incident of a given severity level (in ITIL, defined as mean *time between system incidents* (TBSI))
- Specific incident data (e.g. number of records breached, downtime, time to respond). Such information is typically only provided to the affected customer.

Risk profile considerations:
As this is a meta-parameter (incidents are reported according to the thresholds of other parameters), the thresholds set depend on the parameters triggering the incident. For example, a service dealing with batch-processing of sensitive data may set thresholds for the above parameters according to strict criteria for data leakage incidents and very generous criteria for availability incidents. It is important to note that in IaaS environments, the service provider may have no visibility into the data.

Monitoring methodology:
In order to provide accurate reporting of this parameter, an appropriate incident classification scheme should be in place to provide the above information. Logging and reporting schemes should be in place to record the relevant features of each incident. Examples of incident data to be recorded include (note that the amount of information provided will vary):

- Time of first reporting (by customer or other third-party);
- Incident discovery time;
- Incident resolution time;
- Incident severity;
- Type of incident (e.g. affecting integrity or confidentiality of data);
- Affected assets;
- Residual impact (what could not be corrected).

This information is to be provided to each customer in a way which only discloses information regarding that customer and no others.

Considerations for customer/independent testing:

Such data is largely dependent on the CSP's internal processes. However, the customer can keep track of:

- Time to respond to customer incident reports.
- Time delay between independent detection of service failure and provider reporting. Whether this is possible, depends on whether the customer is able to monitor the parameter triggering the incident independently. As well as monitoring the parameters triggering the incident report, the customer should also keep and monitor logs of detection time and reporting time.

Incident and alerting thresholds:

Alerts may be set according to:

- Thresholds in time-to-invoke and response times or mean times;
- Excessive reporting times or failure to detect (where the CSP is responsible for reporting the incident and independent incident detection is available, e.g. unavailability, vulnerabilities, etc.).

Customer responsibilities:

The customer is responsible for:

- Ensuring their infrastructure, systems, devices, procedures, and activities are not failing in a manner which appears to be an incident in the provider's service offering.
- Ensuring that they accept and understand any incident classification scheme used by the provider (or if appropriate, negotiating an appropriate scheme)
- Providing any customer-side resources required to resolve an incident within the appropriate time frame (e.g. if customer-side diagnostics need to be run).
- Depending on the service offering, the customer may have different responsibilities in terms of incident response and recovery:
    - In a SaaS environment, the customer is likely to have the least incident response and recovery responsibilities, and the CSP would be responsible for providing response and recovery for the entire application stack. The customer may be responsible for handling user incidents (such as accidental or deliberate deletion of user data, and data integrity

compromise). The SaaS provider may allow user data or settings to be restored by the customer in a self-service manner, or may provide an issue tracking system for such requests.

- In a PaaS environment, the customer will typically be responsible for incident response and recovery for the customer application level and above, including the responsiveness and integrity of the application, as well as user data integrity.
- In an IaaS environment, the IaaS provider generally covers incident reporting and response for the underlying hardware, network infrastructure and connectivity. Some of these may not be visible to the customer other than via general service dashboard. The customer is responsible for the operating system and any platforms, applications, and user data running on the CSP's infrastructure, as well as for configuring the infrastructure.

Worked examples:

Example 12: An SLA for an online tax return service states the following objectives:
The mean time to respond to an incident will be under 2 hours for a given month. A severity classification scheme detailing levels from 1 to 5 is defined in the agreement. A monthly service level report includes data on:

- Average time to respond to customer-reported incidents.
- The number of severity 3 and above incidents resolved (system functionality restored) within 1 day.
- The number of exercises performed to test response procedures.

Incident alerts are provided to the customer via encrypted email, based on triggers defined in the SLA:

- When time to respond to customer-reported incidents rises above 4 hours.
- When an incident occurs with severity level of 4 or above.

Where serious intrusions are detected, the CSP will report:

- Assets affected
- Data breaches detected (number of records or data classification, to the extent this is known)

In the case of a severity 4 or above incident, the provider will report on a 2 hourly basis measures taken to resolve the incident and an estimated time to recover.

Example 13: An example of a severity classification scheme could be a scale from 1 to 5 which also includes an 'N/A' level for incidents which have no security impact. The criteria for each level are based on harm to individuals and the number of individuals (so from a breach affecting a single individual to a breach affecting 1000+ individuals), the level of reputation damage/media interest (from none to national media outlets), and the impact on levels of service from none to a long term-impact. The specific criteria used depend on the type of security incident which has occurred.

3. Service elasticity and load tolerance

In certain settings it is important to be able to rely on the provisioning of a predictable or less predictable amount of computing resources over a given period. This parameter is related to availability in that it reflects how the availability of resource provisioning systems changes with demand– i.e. elasticity, which is an essential element of cloud computing. It monitors demand-related failures of availability, as well as giving confidence that the service is able cope with future load. This helps to ensure that as demand for a service increases or decreases, the availability of the service remains constant or degrades in an acceptable way (e.g. with slower response times) for the duration of peak demand.[3]

A CSP may allow service bursts within specific limits, which may depend on either SLAs/contracts, or on available service capacity on the provider's side. If the customer expects volatile usage patterns and regular bursts, which considerably exceed their normal usage, this needs to be communicated to the CSP and/or allowed for in the agreement.

Some CSPs offer reserved capacity which may guarantee (within limits) available elasticity regardless of the current service usage by other users. Such reservation provisions may be critical for disaster recovery (for example, if a regional disaster such as a flood affects a number of businesses in the same geography, and all businesses require burst capacity). The customer is often required to pay an up-front fee for the reserved service resources. The CSP provisions the hardware, software and network resources required to meet the reservation requirements at any one time, which often necessitates overriding provider over-subscription policies, where such policies exist.

CSPs often offer facilities that automate elasticity activities based on customer requirements. For example an IaaS provider may allow the customer to automate virtual machine scale-up/scale-down/scale-out/scale-in activities based on parameters such as CPU or memory usage on virtual machines. If a sudden peak of user activity requires more resources to be allocated in real time (more often within minutes), such automated elasticity provisions may be able to meet the increased resource needs without customer intervention. Likewise, automated elasticity policies should allow resources to be scaled down to a minimum when they are no longer required.

[3] *Scaling down is also an issue (e.g. de-allocating resources after absorbing a denial of service attack to minimise infrastructure cost for resources that are no longer required) but we consider the risk of this mechanism failing to be minimal in most scenarios therefore it is not covered in this document.*

Example 14: A cloud-based service for critical e-government services designed for use as a back-up system in disaster recovery scenarios, wants to be assured that whenever demand for http requests to their servers increases 1000-fold during a disaster, the servers will continue to be available.
Example 15: A tax office using cloud servers to host a service for filing online tax returns would want to be assured that when approaching the deadline for returns to be filed enough extra virtual machines and bandwidth would be available from the provider.

Automated elasticity provisions may serve different needs:

- Absorbing denial of service (DoS), or distributed denial of service (DDoS) attacks with minimal or no user impact (although cost implications must be considered in setting elasticity policies);
- Automatic provisioning of new resources in the case of resource failure (by maintaining a configurable minimum pool of active resources);
- Narrowing the gap between provisioned and actual resource usage with a view to optimising costs and improving user experience and responsiveness

Finally, it is important for the agreement/SLA to set appropriate limits on capacity increases to prevent resource theft and 'economic denial of service' (when an attacker consumes paid-for resources, running up a large bill for the customer) in the event of account compromise (as well as helping to stay within budgetary constraints).

Parameter definitions:

The most important factor to monitor is the ability of the service to securely provision the required resources in the event they are needed. The simplest way to verify this is to perform regular tests and log monitoring of capacity provisioning (reserved or within allowed limits) and check for failures. When monitored in this way, elasticity is reported as the pass/fail value of a regular test. This can be done by the customer for their own reserved capacity (or within contracted limits) or on a service-wide basis by the provider. It should be noted that elasticity for ad hoc (on demand) resources is often a function of overall demand and therefore customer tests may not produce realistic results. Reserved resources should always be available.

The most important factor to monitor is the ability of the service to securely provision the required resources in the event they are needed.

More quantitatively, elasticity as a measure can be described as the ratio of failed resource provisioning requests to the total number of resource provisioning requests over a commitment period. This may be reported either as a historical record of actual values of this ratio, or as a report of the result of scheduled tests reflecting this parameter. The parameter may include requests for a specific resource type only, or a set of requests for all resource types, which comprise the customer's cloud system.

Depending on the context, many different types of resources can be monitored in this way. Affected resources may include, for example:

1. Number of CPU cores;
2. CPU Speed;
3. Memory size;
4. VM quantity;
5. VM storage;
6. VM storage throughput;
7. Bandwidth;
8. Account provisioning;
9. Messaging capacity;
10. Application response capacity;
11. Queue service allocation (maximum size);
12. Allocation of IP addresses;
13. Network bandwidth- within the cloud, across the Internet, over private links.

The resources selected for monitoring will vary depending on the type of service- for example for an IaaS service, it might be appropriate to measure the elasticity of the number of CPU cores available, whereas for a SaaS service, it would be more appropriate to measure the elasticity in the application response capacity.

This parameter is related to measures such as load tolerance (ability to provision extra application or network resources) and peak traffic tolerance, which are sometimes used in monitoring contexts.

Risk profile considerations:
Services with highly volatile demand will have more stringent requirements for this parameter. Highly static applications (e.g. running a set of low-traffic web servers with no demand variation) may not need to include this requirement, although it may still be required to ensure resilience against DoS/DDoS attacks (check your exposure to this risk).

Where resource theft or financial implications of unrestricted resource use is an issue, limits on resource provisioning should also be regularly tested (resulting in a pass/fail report).

Monitoring methodology:

For IaaS and PaaS, elasticity can be monitored using:

- Burst testing to verify the capability of a provider to scale up your capacity in case of need. A burst test involves provisioning a percentage of the extra capacity (using test data, where applicable) for a short time window and measuring the percentage of provisioning failures. After the burst test, the resources are immediately de-provisioned (note considerations for independent testing- for customer testing, this should be within allowed limits). Note that some providers may require notification before the customer commences such testing to suppress fraud and anomaly detection alarms. Such notification, though, should not be used by the provider for purposes other than suppressing alarms, so that there is no interference with burst testing.
- Real-time monitoring or log inspection of resource provisioning, either through test provisioning requests or by analysing service logs. Reports and triggers can be based on actual requests for resources which were denied or delayed, for example, application requests arriving at load balancer vs. requests processed or increased queue sizes (which reflect delayed requests). Note that criteria may also be defined according to geographical location (e.g. US data-centres during the daytime).
- For resource limit testing, requesting resources beyond set limits (where this does not have financial consequences).

Burst testing verifies the capability of a provider to scale up your capacity in case of need. A burst test involves provisioning a percentage of the extra capacity (using test data, where applicable) for a short time window and measuring the percentage of provisioning failures.

Considerations for customer/independent testing:

If resource reservations are provided, it is recommended that the customer test bursts up to the maximum reservation limit on a regular basis. This is especially important where the cloud service is being used as a business continuity measure. Customers are advised to test all resource types that comprise the reserved system or solution, not just individual resource types. Furthermore, customers should test that any associated security measures such as encryption scale in an appropriate way.

Due to possible incidents resulting from such tests, service elasticity and load tolerance testing should only be performed by customers within pre-defined limits (either contracted or notified subsequent to the original agreement). It is also important for the customer that appropriate limits and authentication procedures should

In a public cloud situation, where only on-demand resources are used, such tests may not be effective if carried out by a single customer.

be set up around testing, in order to avoid abuse of testing windows by malicious attackers (e.g. to consume customer resources). The provider should not use such prior notification for temporary provisions that may influence the test results.

In a public cloud situation, where only on-demand resources are used, such tests may not be effective if carried out by a single customer, since the ability of the system as a whole to respond to sharp increases in demand is a function of the overall user demand, rather than the characteristics of a single user's behaviour.

Testing may have financial implications. Especially in SaaS and PaaS environments, the provisioning of a new resource often carries terms and conditions for minimal billing and use.

Testing may have financial implications. Especially in SaaS and PaaS environments, the provisioning of a new resource often carries terms and conditions for minimal billing and use. For example, the provisioning and immediate de-provisioning of a user account may still carry a monthly charge for this user account.

Incident and alerting thresholds:
An incident alert will typically be triggered when a periodic test fails, or when production events signal inadequate resource provisioning. Every failure or considerable delay to provision a resource, whether under normal or disaster recovery conditions is likely to be considered an incident.

User responsibilities:
The user is responsible for:

- Defining the required level of service elasticity and load tolerance during the RfP and contract engagement phases (where applicable). The provider is not responsible for anticipating service elasticity and load tolerance requirements for the customer. However, the CSP is often responsible for managing their own capacity with a view of covering all tenants' needs under normal load, as well as for provisioning reserved resources.
- Defining business continuity/disaster recovery (BC/DR) requirements- these are tightly knit with business processes, and vary considerably across businesses. Providers will typically define their services within preset SLAs and will take responsibility for meeting such SLAs for individual services but not for the customer's business continuity/ data recovery strategy.

Worked examples:

Example 16: An online tax return service has 100 virtual servers in constant active use. In order to cope with periods of peak demand, it has a further 400 virtual servers reserved for immediate availability and a contracted ability to grow their provision by 200 virtual servers per day, daily for up to one week. Thus they could grow to 1800 virtual servers in a week. It would be sensible for them to test this, by both moving some of the 400 reserved virtual servers into active use and by growing the reserved pool (within the 200 VM/day growth limit). If any of those requests failed this should raise an incident report. Such a test might cost the customer a small amount of money, but is a sensible validation of their provider's ability to deliver on the contracted service.

Example 17: The service provider of the online tax return service provides regular reports detailing the ratio of provisioning failures to the overall capacity in operation, as recorded in their logs.

Example 18: SaaS: suppose a customer administrator reserves 1000 user IDs and wants to start using them all the next day. The administrator tries to provision (and immediately de-provision) 100 test accounts in 2 minutes (using a comma separated values (CSV) file) and measures the number of failed requests.

Example 19: An IaaS service provider sets an internal alerting limit on the queue for the provisioning of new virtual machines in a particular data centre. When this limit is exceeded, an indicator appears on the service dashboard (this can be compared to a call-centre which tells customers how long they are likely to have to wait to talk to a representative- it shows customers when they are likely to have problems obtaining extra resources).

Example 20: Consider a customer using a cloud storage service for non-sensitive files. Each user has an allocation of 100GB. The customer has used an online interface to set limits on individual users within the company so that each user can increase their storage allocation (with cost implications), but they can only increase it to a maximum of 250 GB. This prevents compromised accounts and employees from consuming storage (and financial resources) beyond a certain limit. Since the failure of this limiting system could have financial implications, the customer tests it once a month by requesting resources beyond the set limits.

4. Data lifecycle management

This group of parameters measures the efficiency and effectiveness of the provider's data handling practices, including the service's back-up or data replication system, the ability to export data and data loss prevention systems.

Parameter definition:
Relevant parameters include:

- Back-up test frequency and results. In IaaS and PaaS, there may be back-up and storage systems under the control of the customer, which should be subject to detailed testing. In SaaS, back-up is likely to be entirely under the control of the provider (details of the operation of back-up systems may be confidential).
- Restoration speed: the time taken to obtain data from back-up from the time of request. NB. Where back-ups are encrypted, this may depend on the performance of encryption systems.
- Success or failure of operational back-ups (successful means: delivered, complete, passing an integrity check, and conforming to the pre-defined format.)
- Data recovery points: the age of the most recent data restored, where applicable.
- Export test results: e.g. simulation of termination of service- data export to customer, data integrity check and parse (format) according to pre-defined output or exchange formats. Note that the export rate will usually be limited by telecommunications links. It is nevertheless useful to test this as part of overall service monitoring by the customer.
- Percentage of response to requests for data export successfully completed within pre-defined turnaround parameters.
- DLP (data loss prevention) system logs and system test results, where available.
- Data durability: some providers specify a durability parameter which relates to the amount of data which can be lost in a time period. For example a 99% durability SLA would mean that if you store 100 MB of data for a year that you don't back up, up to 1% of it (1 MB) may get lost by the end of the year. Note that this is a statistical measure which does not guarantee the integrity of any specific data block – rather it says that there is a 1% probability of losing an individual data block.
- Scheduled deletion failure: is data present in back-ups when it should have been deleted (according to retention periods)?
- Legally disclosable (i.e. not subject to suppression requests) regulatory requests to the cloud provider for information affecting the system may be logged and reported to the customer.

Risk profile considerations:
The following considerations apply:

- Organisations with high business continuity requirements should set strict thresholds for these parameters.

- Thresholds depend on the mission-criticality of data being backed up and the ease of recovery from other sources. For example, an index can be recreated but design documents which are not backed up elsewhere cannot.
- Customers should bear in mind that, where personal data is concerned, a scheduled deletion failure may represent a failure to meet data protection obligations (retention period).

Thresholds depend on the mission-criticality of data being backed up and the ease of recovery from other sources.

Monitoring methodology:
Output from these parameters is available from:

- Logs of back-up operations: operational back-up and restore errors or failures should be detected and logged.
- Back-up system test frequency and results:
    - The simplest method is a request for back-up data allowing the monitoring of the timing of data restoration and the integrity of the results.
    - A more radical test is the 'chaos monkey' approach where a random portion of the data set is deleted (under controlled conditions) to test success and speed of recovery. In this case, the data needs to be either test data or backed up independently (in case the test fails).
- Recovery points for data back-ups can be tested by writing data, deleting it at predefined intervals and testing the recovery point of recovered data.
- Data deletion tests: it is impossible to prove that a provider **does not** have a copy of data (this is an example of the so-called "open world problem" in logic) but it can be demonstrated that they **do have** a copy of the data. This can be achieved by requesting data which should have been deleted- if it is available, the test fails.
- DLP systems should be regularly tested. There should also be a pre-defined reporting and alerting format (including encryption mechanisms) when serious breaches are detected.
- The CSP, who should, where permitted, keep logs of government requests and which customers are affected (e.g. requests for information under the UK RIPA- Regulation of Investigatory Powers Act). Laws and regulations however may prevent the Service Provider from disclosing such requests or logs to the customer.

Considerations for customer/independent testing:
The above parameters can only be monitored independently where they are under the control of, or can be verified by, the customer. Depending on the test, different permissions are required- for example, for recovery point tests, the customer must be able to request data restoration on demand, where applicable.

For data retention, it may be possible for the customer to create a test data set that falls within a specific data retention/back-up policy, and request restoration of the test data when it should have been deleted according to the policy.

By definition, independent testing of data portability mechanisms is always possible for this parameter. Data portability is one of the most crucial parameters to be tested by the customer or an independent party.

Incident and alerting thresholds:
Thresholds should be set on test failures, breaches detected and operational back-up and restore failures.

Customer responsibilities:
The customer is responsible for:

- Appropriately defining and communicating back-up, data retention and deletion requirements during the RfP and contract engagements. This includes clearly communicating any legal or regulatory requirements which may apply to the customer's data may.
- Designing an appropriate degree of redundancy and back-up into systems over which they have control (e.g. virtual machines in IaaS environments). Depending on the service offering, customers may be responsible for data back-up. In this case, the customer will need to develop and test their own procedures and not the provider's procedures.

The customer is responsible for designing the appropriate amount of redundancy and back-up into systems over which they have control.

- Designing processes to handle data retention and deletion requirements in systems they control.
- Handling any legal requests directed at them. The customer may have the tools and information readily available, or alternatively ensure that the provider is under a contractual duty to cooperate – to the extent permitted by the law – in order to furnish them with relevant information that may be required under such a request.

Worked examples:

Example 21: An emergency service for handling police operations uses a private IaaS cloud as failover for its services in case of the failure of its primary data centre. Periodic tests are co-ordinated between the CSP and the customer, providing the following information:

- Time to migrate;
- Interoperability failures;
- Data recovery points.

During any operational use of the failover service, the same data are collected.

Example 22: A health service uses a private IaaS cloud service to provide a back-end for mobile health-professionals accessing patient records. Availability of the system is highly critical and it is designed (by the customer) to withstand the failure of a large percentage of server instances. The resilience of the service is tested on a non-production test system by measuring the degradation in service resulting from shutting down a percentage of random instances.

Example 23: A SaaS provider carries out regular tests of its data back-up systems. The customer is informed via a dashboard of the frequency of these tests. For security reasons, the SaaS provider does not disclose the test results.

Example 24: A SaaS customer uses a cloud based office suite. The SLA states that the customer will be able to export all data (documents, spreadsheets, slide-sets etc...) from the service in a number of specified standard formats, using a specially provided bulk export service. The bulk export service commits to an availability level including a minimum export speed. The customer performs regular tests of the bulk export service, by exporting a fraction of their data and checking formats, integrity and performance.

Example 25: A national central bank uses a web content filtering service to protect its staff against malware and to prevent downloading of content which contravenes their policy. Local regulations stipulate that government employee data can be stored for the purposes of fraud detection for a maximum of 3 months. The web content filtering company commits to the deletion of all records within 3 months. The web content filtering service provides a dashboard where the customer can browse the web history of an individual employee. The customer tests this system for presence of records older than 3 months.

Example 26: A national pension scheme uses a private cloud to manage citizen data. The service uses a data loss prevention (DLP) process and toolset. The SLA specifies a standard reporting format for data loss incidents detected by the process, including number of database records breached and the percentage containing personal citizen data. Furthermore, the SLA specifies a testing schedule for the DLP process and a reporting format for the test results. A clear process is defined for dealing with any data breaches of personal citizen data including informing , if and when appropriate, relevant data protection authorities (DPAs) and the citizen of the breach[4].

4 Please note that that there is no express obligation to notify the DPA or citizens throughout the EU Member States. However, such provision is set forth in the EU Commission Proposal for General Data Protection Regulation.

5. Technical compliance and vulnerability management

This group of parameters measures the ability of a service to comply with a technical security policy, including controls in place and the handling of vulnerabilities. It also allows customers to apply customer-side workarounds in situations where a fix is not yet available at the time of vulnerability reporting. Monitoring of technical compliance and vulnerability management is often understood in terms of deviations from a baseline security policy. Even if this is not formally defined in a single place, the following elements of a technical security baseline may be found in the agreement with the provider:

- Security-related configuration and options to be used.
- System components covered by security controls and configuration elements (e.g. does it include network components, guest OS layers etc.).
- Selection, schedule and information on software updates and patches to be applied, e.g. what patches, patching frequency and covered systems.
- Criteria and procedures for vulnerability discovery, reporting and remediation. Are vulnerabilities checked from a specific vendor-approved list, a public database, an approved penetration testing tool etc...?

It is important to take into account the following considerations in relation to technical compliance and vulnerability management.

1. Automatic vulnerability scanners are usually designed to detect flaws in COTS (commercial off the shelf) and some commonly-found open source software. Specialised software written in-house by the provider is likely to be out of scope and COTS vulnerability scanners generally do not work in a cloud environment due to the specialised software used. Whether COTS, open source or specialized software is used, automated scanners may report Type I (false positive) or Type II (false negative) errors. For example, it may show vulnerabilities where in fact compensating controls are in place, or system functionality is not enabled (e.g. a service is installed but not running), meaning that there is no risk. For this reason, it is especially important to define vulnerability management in terms of an agreed scope and set of baseline controls.

   Specialised software written in house by the provider is likely to be out of scope.

2. Vulnerabilities may be addressed in many different ways- patching may not always be required due to other mitigating controls. For example, web application firewalls may address threats without altering the underlying system (such approaches should be part of a defence-in-depth strategy). Vulnerabilities may be addressed using conventional patches, virtual patches or other mitigations, such as procedural controls. Where a security baseline definition is part of the agreement with the provider, a process should be in place for agreeing to changes to the baseline, including the timely application of updates required to address vulnerabilities where there is no other control to mitigate risk.
3. Providers usually do not have information about affected asset values, which are an important factor in vulnerability management and risk assessment, especially in IaaS and PaaS clouds.

4. Providers may not disclose detailed information about vulnerabilities for security reasons (publicly disclosing vulnerabilities may assist attackers) or because of commercial sensitivity (vulnerability descriptions may include proprietary information). There are three key possibilities:
   a. The provider runs custom software developed, run, and operated internally as part of the service. In this case, the provider could balance security with transparency by (for example) periodically reporting numbers of vulnerabilities addressed or patched before public disclosure vs. those that went public before the fix.
   b. Provider operated software sourced from a third-party. In this case, the vulnerabilities are often discovered and reported by a third-party and the provider should monitor such reports and provide a statement of whether or not their service is affected and what is the potential impact on their customers.
   c. Customer operated/developed software run in the provider environment (IaaS or PaaS). In this case, the provider could offer a service to its customers in providing warnings or patch alerts for third-party software or even security verification services for custom developed applications. This is equivalent to a separate managed security service offering and would most likely fall under a separate SLA or best-effort service. NB. The scope of the provider environment should be clearly defined.

Providers may not disclose detailed information about vulnerabilities for security reasons.

5. It is not possible to address all vulnerabilities, so it is important to focus on whether or not vulnerability introduces risk, and whether the baseline (or additional compensating controls) addresses that risk. Addressing some vulnerabilities can create more vulnerabilities, i.e. patching may be more dangerous than not patching (e.g. if a poorly tested patch includes an updated library which causes dependent components to fail). Additionally, applying software updates or patches without regression testing can impact service integrity and availability. Customers procuring IaaS and PaaS clouds must consider any role which may be required of them in testing updates to software before approving them for deployment.

It is not possible to address all vulnerabilities, so it is important to focus on whether or not vulnerability introduces risk, and whether the baseline (or additional compensating controls) addresses

Parameter definitions:

Based on the above, the customer can receive reports on deviations from the baseline security policy, which provide useful indications of the effectiveness of technical compliance and vulnerability management processes and trends in their performance.

As well as prompting remedial action, this also enables the establishment of trust between the provider's system and other systems with which it integrates. For example, other components of a customer's system integrating the cloud service could use information about the security

It is important that this information is communicated securely, and not made available to potential attackers.

configuration, patching levels and vulnerability management as a basis to decide whether or not to use a particular service component (it is important that this information is communicated securely, and not made available to potential attackers). In practice, this provides a framework for reporting on (subject to confidentiality considerations):

- Information on patches and controls in place vs open vulnerabilities.
- Information on compensating controls applied.
- Data on specific vulnerabilities and trends, such as their classification and severity scores. A commonly adopted vulnerability classification scheme is the Common Vulnerability Scoring System (CVSS) (22)[5]. Note the above caveats on the potential confidentiality of certain vulnerability information for the cloud provider. Also, as noted above, assets affected by a vulnerability are an important factor in assessing its impact. However, CSPs often cannot assess the value of assets affected as data protection prohibit access to the customer's data. Aggregate information on base vulnerability scores can nevertheless provide a useful indicator of trends which may need to be addressed (in CVSS, this is the score given to the vulnerability without taking into account the assets affected). Furthermore, the customer may be able to add information on the assets affected.

  Reports may also specify whether the vulnerability is the result of the CSP's software/configurations or whether it is the result of third-party components integrated into the CSP's service.

Risk profile considerations:
Sensitive services should set a stricter security baseline policy and stricter criteria for deviations from the policy (e.g. trigger alerts for lower CVSS scores of vulnerabilities discovered).

Monitoring methodology:
Monitoring and detection of vulnerabilities and/or deviations from stated security baseline is possible via:

- Regular scans of all covered systems for vulnerabilities;
- Reporting and detection of configuration changes (see [**Change management**]);
- Reporting channels from users and independent security researchers;
- Public vulnerability reporting programmes such as CVE (common vulnerabilities and exposures);
- Vendor mailing lists which report discovered vulnerabilities and associated patches;
- Public mailing lists such as 'Bugtraq' (23) and 'Full Disclosure' (24).

[5] *For an illustration of a tool for classifying vulnerabilities see example the CVSS Calculator (28).*

The following data can be used as input to reporting of vulnerability management processes:

- Vulnerability disclosure/discovery times;
- Vulnerability classification data (e.g. CVE number, remote exploitability etc.);
- Patch/fix release dates;
- Patch/fix application dates.

Considerations for customer/independent testing:
Customer vulnerability and technical compliance testing is often limited by terms of service. The contract should clearly specify the conditions for independent testing to take place- on which systems or system components it may be performed and which kinds of tests are permitted (the CSP may not allow specific types of testing such as DoS). Even if it is possible to carry out vulnerability scans, there will more than likely be restrictions on what can be carried out by an external party (e.g. only the systems managed by the account holder). It is also important to consider that independent testing generally only covers 'outsider' threats. Data reported by the CSP may be verified independently by:

- Cross-checking with public vulnerability disclosure sources (e.g. CVE), independent abuse reports.
- Third-party security testing of systems and services (assuming relevant access to the host systems and agreement from the CSP), for example, using standard or customised vulnerability testing tools and methods (e.g. fuzz testing of user interfaces), either by third-party security services, AV scanners, customer security teams or vendor tools. For third-party testing, various certification schemes exist for testing organisations. These can provide an extra level of assurance to the results. For example, the UK NHS recommends the use of CHECK (25) or CREST (26) approved testing organisations.
- Another possibility for independent testing is to provide an independent disclosure channel for third-party researchers. However the terms of such an arrangement must be clearly agreed to by the provider.

Incident and alerting thresholds:
Thresholds for alerting, and incident reporting may be set according to:

- Lack of baseline compliance;
- Infrequent updates to baselines without evidence that compensating controls are addressing newly discovered vulnerability;
- High (single or aggregate) values of (time to patch * vulnerability severity)- this indicator flags failure to address severe vulnerabilities quickly.

Customer responsibilities:
Customer responsibilities in this area vary according to the service model. The following are some considerations:

	IaaS	PaaS	SaaS
Detection and remediation	Customer is responsible for technical compliance and vulnerability management in virtual machines and all layers above, including regression testing of patches and software updates.	Customer is responsible for technical compliance and vulnerability management, and management above API layer. In some cases, customers can perform regression testing and request that updates to base environment and API layer not be applied.	CSP may not be able to disclose information for security or commercial reasons.
Classification	As above.	Customer should add weighting factor according to value of assets affected.	Customer should add weighting factor according to value of assets affected.

In addition, the customer is responsible for :

- Taking reasonable steps to ensure that their processes, systems, devices and activities are not the cause of any perceived vulnerability in the CSP's service(s).
- Alerting the provider of any independent testing carried out, according to the terms of service.
- Understanding what independent tests are permitted according to the provider's terms of service.
- Alerting the provider to any vulnerabilities detected by the customer whose remediation is within the provider's responsibility.
- Conducting regression testing with new patches, security fixes and software updates to the elements of the system for which the customer is responsible.
- Applying patches and security fixes in a timely manner to elements of the system for which the customer is responsible. It should be defined whose responsibility it is to do any patching (i.e. whether it should be the customer or the CSP- depending on the service being delivered).

The customer is responsible for taking reasonable steps to ensure that their processes, systems, devices and activities are not the cause of any perceived vulnerability in the CSP's service(s).

Note that patches and mitigations may affect other systems depending on the service (see change-management).

Worked examples:

Example 27: A government agency implements its email lists using a hosted service running well-known third-party COTS software called MailListEx. The agreement describes a number of commitments to security controls, system configuration elements and vulnerability management procedures to be implemented by the hosted service (a security baseline). The provider also implements internal security controls which are not disclosed. On this basis, the provider makes the following commitments:

- Any vulnerabilities affecting MailListEx which are present in the system and listed on well-known vulnerability databases (e.g. CVE) will be patched within 1 day of patch release;
- Any vulnerabilities above CVSS 7 to be notified via encrypted email to the customer;
- Monthly independent vulnerability testing of the system will be carried out;
- The MailListEx administrator console will be accessible only from a certain IP address via TLS.

At the same time, the agency also monitors vulnerability databases for vulnerabilities affecting MailListEx.

Example 28: A government-run scheme for trading taxi licences uses a cloud based auction service which is offered as a government branded web site. The agreement specifies that the CSP should offer:

- A web based security dashboard reporting, among other information, the number of vulnerability assessments carried out in the last quarter (dates are not given, since they could be used by an attacker to hide vulnerabilities during the assessment).
- Alerts via encrypted email for vulnerabilities detected with CVSS scores of 7 and above, with information on systems affected, timing, etc. The customer-side IT team creates an internal escalation for vulnerability alerts taking into account the affected assets. If the affected assets include the licence database, an escalation procedure is triggered.

In addition, the customer-side IT team specifies a reporting address for vulnerability disclosure. The team also monitors CVE reports relating to the systems.

Example 29: A university department uses a community IaaS service to analyse experimental data from a number of project partners. The service agreement allows for weekly vulnerability testing of customer virtual machines for a period of 15 minutes using a (and where relevant, authenticated) third-party provider.

6. Change-management

This group of parameters is used to monitor and manage critical changes in the security-relevant properties of the system and its configuration.

Parameter definitions:
This group of parameters includes

- Change management process testing frequency and results;
- Change notice time: notice period for changes notified to the customer;
- Change triggers: changes which should be reported to the customer when they occur. Some of the most important include:
    - Loss of certification status (ISO, PCI etc.);
    - Major changes in staff clearance status;
    - Changes or extension of jurisdictions in which data processing occurs;
    - Patches and major system changes which might affect the operation of dependent components not under the provider's control (e.g. changes in authentication systems);
    - Significant changes in security controls and processes used, e.g. encryption key lengths and key management processes.

  NB As a general guide, anything which affects customer certification requirements should be notified to the customer;
- Time to implement security-critical customer change requests e.g. to provision/de-provision/change access privileges.

Risk profile considerations:
Triggers for alerts or incident reporting depend on the nature of the service. For example:

- Services managing highly confidential data should trigger alerts for any changes to key management procedures and encryption algorithms/procedures. Encryption of data is often the responsibility of the customer;
- Services with location requirements (e.g. healthcare services operating in countries where data is required to remain within state boundaries) should trigger alerts for any changes to data storage locations;
- Services with high availability requirements may wish to have frequent reports of change management procedure testing results (since outages can frequently result from change management errors).

Monitoring methodology:
At contract signing time a predefined list should be included, of features which, if changed, require a notice to the customer. A documented change management procedure should be in place, as well as tools to support the logging, reporting and testing of changes.

Considerations for customer/independent testing:
In general, independent testing is not possible in this domain. However certain features of a service can be detected externally. For example, changes in encryption algorithms supported or public security-relevant APIs supported. These can be monitored independently.

Incident and alerting thresholds:
Boolean triggers which require immediate notification should be identified (as opposed to those included in service level reports or dashboard information).

Customer responsibilities:
Customers are responsible for:

- Ensuring that their users, systems, devices and processes are not responsible for delays in provisioning/de-provisioning/changes to system settings, including access privileges.
- The customer should ensure that the agreement provides adequate change reporting to manage critical dependencies on the provider's system which affect overall service operation.

Worked examples:

Example 30: A national healthcare organisation uses a third-party service to perform penetration testing on their systems. In order to meet national guidelines and ISO 27001 audit requirements for third-party service providers, the organisation is required to provide evidence that the third-party has performed background checks on their employees. In order to meet this requirement, they use a penetration-tester that has been certified according to the EXAMPLECERT scheme which requires employee background checks. The agreement states that the supplier is required to notify the customer if their EXAMPLECERT certificate is revoked or not renewed.

Example 31: A national healthcare service uses a cloud storage facility to store patient records. The storage facility automatically encrypts all data at rest and undertakes not to transfer the data to any country which is not EU-adequate (27) (some national regulations do not allow the transfer of the data outside the country). The healthcare provider complies with local laws on encryption of patient data (with algorithms and key lengths specified by a government appointed committee). Among a number of change notification triggers, the agreement with the storage provider therefore states that the provider will immediately notify the customer if their compliance status changes.

Example 32: A PaaS provider publishes a continuously updated roadmap of API version releases, detailing when major changes will be implemented. Customers use this to plan testing schedules.

7. Data isolation

This parameter group covers discrete access to a shared pool of resources by legitimate users for legitimate purposes. Isolation is an essential element of all types of cloud environments- IaaS, PaaS and SaaS, but the mechanisms used to implement isolation, and potentially the parameters used to measure it, will differ considerably. [6]

Data isolation ensures confidentiality, integrity and availability of user data and services between different customers, as well as protection from unauthorised third-party access. Failures in data isolation are essentially a type of vulnerability (see [**Technical compliance and vulnerability management**] for general advice on how to monitor vulnerability management).

Data isolation ensures confidentiality, integrity and availability of user data and services between different customers.

Typical customer isolation concerns include the following:

- IaaS/PaaS: protection of data in memory: can other tenants read and/or modify the blocks of RAM allocated to a particular user process/application instance/virtual machine? A good multi-tenant platform should provide for complete memory isolation between tenants, as well as process and user isolation within the customer account;
- IaaS /PaaS/SaaS: protection of data at rest (on disk/within the database): can other tenants access my files/my objects/my database records?
- IaaS/PaaS/SaaS: protection of data in transit: can other users intercept my data at any layer while it is being transferred across the network? The CSP should offer a sufficient level of network isolation (where this is under their control) between the tenants so that no tenant can see or interfere with data in transit- whether on the wire or within the platform's protocol stack.
- IaaS/PaaS/SaaS: secure deletion: If I dispose of a memory block, or storage space on a block or stream device, is it recycled in a secure fashion before it is provided to other tenants? For example, if I dispose of a virtual disk, and another user accesses the disk without erasing it first, will they see my data?

---

[6] *Performance isolation is another factor which has implications for availability. Performance isolation ensures that individual customer activities do not affect the performance of other tenants' environments on the same platform or environment. We have not recommended the customer monitoring of performance isolation however, since performance isolation failures are equivalent to availability failures from a customer perspective (this is just one of the causes of availability failures). They are also a function of overall demand (when demand for a particular resource is saturated by the existing customer base, the next customer to request a resource will be denied). Finally performance isolation failure is usually very difficult to prove since it typically involves what amounts to a regression analysis over a large number of variables.*

CSPs are responsible for providing isolation, and the customer rarely has visibility into the technology or the configuration that the CSP is using to achieve this. Still, this section defines a number of considerations to test for isolation on an on-going basis.

Data isolation is a functional requirement, and must be present at all times.

Parameter definition:
Data isolation is a functional requirement, and must be present at all times. Therefore, it is a Boolean (yes/no) parameter- customer data is either isolated, or it isn't. Monitoring systems should detect every individual event where customer data is not isolated. An isolation incident may or may not be visible to the customer. The provider may or may not be aware of the incident. Data isolation incidents can have a potentially serious impact and the customer should be alerted as soon as possible after they are detected.

Risk profile considerations:
Data isolation should be prioritised to the extent that the service is processing sensitive data.

Monitoring methodology:
Either the customer or the provider, or both, must ensure that a formal process for monitoring data isolation is in place. Isolation controls, similar to other types of controls, must be tested on a regular basis, especially as provider platforms evolve, undergo functional and configuration changes, or the threat profile changes.

Data isolation is best tested through penetration testing (either by the provider or the customer). Data isolation should be tested in cloud environments as part of regular penetration testing (see [5. Technical compliance and vulnerability management]). For data isolation testing, only test data should be used; Using operational data creates a security risk.

Considerations for customer/independent testing:
Several tests for data isolation can be used as an empirical method to validate customer (or even user-level) isolation in a cloud platform. For example, customers can test memory and storage isolation independently. Where appropriate permission is granted, additional penetration testing can be carried out by a specialised third-party provider, or by the customer. NB. However, independent penetration testing by the customer or a third-party contracted by the customer can only be undertaken according to the bounds acceptable within terms of service and ethical behaviour.

Customer responsibilities:
Customers are responsible for implementing sound architectural choices which support performance and data isolation of systems under their control. For example:

- Use encryption, where feasible;

- Use secure wipe processes in IaaS environments (software-level wipe where data replication/back-up is guaranteed not to exist, or secure deletion of encryption keys for encrypted data);
- Use queues to isolate a component from temporary failures in other components.
- Use independent (of other customers) sources of randomness.

Incident and alerting thresholds:
When testing data isolation, every data isolation incident is severe and must generate an alert.

Worked examples:

Example 33: A tax service uses a private IaaS cloud to provide virtual web servers. As part of regular penetration tests, server memory and storage allocation are tested for traces of residual data.

Example 34: A health service using a SaaS service for document sharing agrees with the CSP to use the services of a third-party penetration testing company to test data isolation (e.g. testing that access control systems are secure). They agree with the provider to run tests via a set of specially provisioned test user accounts. They agree on one day every 3 months when these accounts will be activated. The third-party testing company is certified according to the CREST (ethical security testing) scheme (26).

8. Log management and forensics

This parameter covers access to information about historical events related to the usage of the cloud resource allocated to the customer. Customers may need to obtain information on what data was processed or disposed of and when, where, how and by which of their users, in accordance with their internal control, compliance, audit, or legal and regulatory requirements. Note that **in this section we do not specify which data should be included in logs**, but we assume there is an agreement between the customer and the provider about what will be logged and what logs should be accessible to the customer[7].

Customers may need to obtain information on what data was processed or disposed of and when, where, how and by which of their users.

Here we cover the security and availability of logging and forensics related systems. For example: if the provider is contracted to make available user access logs, this parameter will cover reporting of when they are unavailable.

[7] *Note that different customers may have non-compatible logging requirements, therefore it may not be economically feasible to configure a single set of fundamental logs to satisfy all customer requirements.*

Multi-tenant outsourced services usually cannot give access to raw log data as it contains records of multiple users and thus would compromise the privacy of other customers. Extraction of data attributable to a specific user may be difficult and/or have cost implications; therefore, logs provided are usually 'derived logs,'- i.e. a subset of the logs kept by the provider. The contract/SLA should specify clearly what log information data should be available to the customer.

Logging system availability and accuracy is important for incident response and compliance with certain legal requirements.

Parameter definition:
This group includes the following parameters:

- Availability (as in continuous functioning) of any logs specified in the contractual agreement. The definition of availability may also include:
    - Log access times: how quickly can such logs be provided on request?
    - Availability for write-access (i.e. is the system able to record events?).
- Availability (as in continuous functioning) of non-repudiation systems, e.g. signed time-stamping, WORM (Write Once, Read Many) devices, trusted third-party logging systems etc.
- Log accuracy: does the logging system record events accurately and does it include events which do not relate to my system/usage (when it should not)? Has the system been compromised by an attacker?
- Commitment to respond, in relevant cases, within a certain timeframe, to extraordinary requests for information regarding the use of the customer's system e.g., from law enforcement (where the requested information is not available to the customer). This does not necessarily mean that the CSP will comply with the request, nor that the CSP is responsible for responding to all such requests (e.g. it would not be necessary where the customer can retrieve data themselves).

Risk profile considerations:
Monitoring of log availability is crucial to trace back events and allocate liabilities and responsibilities. The more sensitive the information involved, the more monitoring of log availability is crucial. Log data is also important for incident response so business continuity requirements should be taken into account when reviewing this parameter. Finally, log data is often needed to satisfy corporate data governance and compliance requirements – e.g., Data Protection Law and SOX-like laws in Europe.

Monitoring methodology:
Monitoring of log management systems includes:

- Regular randomised tests of log availability. The results of these tests should be provided to the customer (where applicable and subject to the support agreement with the service provider).
- Verifying the accuracy and availability of event recording and non-repudiation systems using controlled, independently monitored events and examination of the logs to check availability (did the event get logged at all?) accuracy (did it get logged accurately?) and authenticity (is the evidence such as signed time-stamp and other relevant digital evidence available and valid)[8].
- Where extraordinary requests for information are monitored, a record should be kept of the time that the request is filed.

Considerations for customer/independent testing:
The following should be considered with respect to independent testing of log management:

- Periodic tests of log availability may be carried out by the customer, (if dependent on the CSP providing log files, then subject to the support agreement with the service provider).
- Testing of log integrity and availability of non-repudiation systems may be performed independently using controlled events triggered by the customer and by examining the logs to check integrity and authenticity.

Testing of log integrity and availability of non-repudiation systems may be performed independently using controlled events triggered by the customer and examining the logs to check integrity and authenticity.

Incident and alerting thresholds:
Alerting thresholds should be set according to availability and accuracy.

Customer responsibilities:
The customer is responsible for:

- Appropriately defining and communicating logging requirements during the RFP and contract phase.

---

[8] *Note that none of the above tests will detect the insertion of false logs- i.e. logs claiming events which did not happen. This is much more difficult to detect but may, to some extent be monitored by:*

- *Cross-checking with other (independent) sources of evidence- e.g. cross-checking samples of firewall logs with server logs.*
- *Reporting of access control lists and access logs by system operators for logging systems.*

- Logging events which are only available to the customer (e.g. VM events within IaaS). When cloud usage is on an IaaS or PaaS basis and the application belongs to the customer, the customer is responsible for many aspects of logging (although the cloud provider may be offering the infrastructure meeting the appropriate availability and integrity requirements).

Worked examples:

Example 35: An EU agency uses a SaaS service to store and manage field reports on fishing boat inspections. As part of the service, all reports are digitally signed. The SaaS provider agrees to make available a record of login times and source IP addresses and report signature files of all agency users filing reports in the last 3 months, through an automated interface accessible to an authorised, authenticated account available to the audit department and an automated monitoring system. In order to test the availability and correct operation of this logging service, the agency requests the logs of report signatures and timestamps once every 3 months. The agency verifies their availability and authenticity, and cross-checks the logs with its client logs to check for inconsistencies (this is done using an automated script).

Example 36: An IaaS provider allows customers to view logs of various customisable parameters such as network throughput etc. The provider captures and publishes availability statistics for this customer-facing logging system on a public dashboard. The customer SLA commits to an availability level for the customer-facing logging systems of 99%. The availability level is defined according to the guidelines of parameter 1 as: 'The monitoring dashboard at the service portal web page will provide data to all authenticated users from the last month of system operation for an average availability of 99% over a monthly period'.

Example 37: An EU agency hosts its web site using a cluster of virtual machines with an IaaS provider. One day, the CIO receives a phone call informing him that the web site has been defaced and there has been a complaint of a DDoS attack emanating from the server cluster. The CIO thought that the IaaS provider was responsible for keeping logs and thus did not ensure that appropriate logs were kept to investigate the incident and prove that the agency is not responsible for the DDoS attack. Furthermore, the agency site's recovery from the attack and system patching to prevent further attacks is seriously delayed by the lack of logs. This results in long term, serious negative publicity and loss of operations for the agency.

Example 38: An EU agency uses SaaS office software to create reports collaboratively with stakeholders. A stakeholder makes a complaint that the agency has breached its IP rights by making an early draft of a report publicly available for a short period. As part of the dispute, the cloud provider is requested to provide historical logs of access control rules in place on the document. According to the contract, the provider agrees to respond to such requests within 1 week. In this case, they respond within 1 week to say that they are unable to provide such logs in order to protect the confidentiality of other customers.

Checklist guide to the document

How to use the checklist

This checklist is intended to be used as a quick guide to the issues in the document. By no means every question will be relevant to every reader.

- Use this to check you have covered the main issues in the on-going monitoring of customer-side security aspects of cloud contracts.
- Give this to someone in your team who is familiar with the technical issues involved. We have tried to keep the questions as simple as possible, but some issues will need input from IT managers or CISOs. It is assumed that procurement teams will have the support of technical experts and will work cross-functionally to ensure requirements are effectively addressed.
- Each parameter is divided into:
    - What does it mean for your organisation? What should you bear in mind about your own specific requirements before you start thinking about the parameter?
    - How is it defined? What are the variations and ambiguities to consider when looking at how the parameter is defined?
    - How is it measured and reported? How is it measured in practice- this includes some technical considerations?
    - How can you verify and measure it yourself, where possible.
    - What are your responsibilities?

How not to use the checklist:

Do not:

- Attempt to answer every question- many questions will not be relevant to your organisation. The initial section of each parameter asks some questions which should give you some idea of which of the following questions are relevant to you. You can then skip those which are not.
- Give the questionnaire to your provider and ask them to answer it for you- these are questions for the public sector cloud customers, not providers.
- Interpret any questions as requiring fixed answers- the questions are not intended as '*do you xyz*'– '*if not, you should*'. In general, there is no single correct answer: readers should refer to the full document for guidance on alternative responses.
- Interpret examples as definitive. Not all questions are applicable to all types of cloud service.
- Use this out of the context of the rest of the document.

Checklist

1. Service availability

- What does availability of the service mean for your organisation and your risks?
  - When you think of a service outage, which are the most critical aspects of your organisation's mission affected?
  - Which functions of the service or its dependencies are most impacted?
  - How long an outage can each service function tolerate without significant impact? The same average downtime over a month can be spread out differently- one long incident or several shorter ones. If this is critical, is it captured in a mean downtime parameter?
  - Are your availability requirements uniform or are they greater at specific times of the day/week?
  - How important is it to know immediately when the service is down? (Will I receive appropriate alerts or feeds?)
  - How important is it to know in advance about (scheduled) downtime? If so, is this captured in scheduled downtime reporting requirements?
- How is availability defined? (Check this against the results of the previous question.)
  - Which functions and/or users should be covered?
  - According to what criteria are the service-functions considered unavailable?
    - Do you define a minimum period of downtime or a number of failed requests over which availability measurements are made?
    - Is down-time defined in terms of numbers or geographical coverage of affected users or machines?
    - Are there other criteria for defining when the service function is unavailable (what defines a failed request)? For example, returning an error code.
  - What commitments are made by the CSP in terms of average percentage uptime and mean downtime?
  - With what granularity do I want the CSP to commit to availability (1 month/ 1 week/ 1 day)?
  - Are availability criteria different for different sub-periods?
- What metrics and reporting are in place to support monitoring of availability?
  - Are any of the following methods used- user reports, CSP logs, CSP metrics, sample requests, automated monitoring tools?
  - Is there a service dashboard including availability? If so, what functions does it cover?
- How can you monitor availability from the customer perspective?
  - Do you plan to monitor availability independently? (Where availability is critical, this is advisable). If so, what methods will you use to test it?
  - Does your provider supply you with alerts when availability parameters are outside of the pre-set thresholds?

- Can you rely on monitoring data obtained from the CSP or would you rather rely on your own tests and statistics?

- Your responsibilities:
  - If your service delivery depends on several providers, does your system monitoring cover the service end-to-end (including network connectivity) so that you will be able to detect which element of the service delivery is unavailable?
  - If using independent tests, are these acceptable to the provider in terms of type, volume and frequency?
  - Have you checked that any independent tests do not trigger anti-DDoS or CAPTCHA systems?
  - What design decisions affecting availability have been made for the system components under your control?
  - What tests do you perform to test the resilience of the system components under your control?
- Are there penalties defined for unavailability – if so, are they based on monthly average downtime or maximum length of single outages?

2. Incident response

Note that alerts and response to specific kinds of incident are covered in the alerts and monitoring considerations for the other parameters. This parameter covers the monitoring of the overall response capability of the provider.

- What does incident response mean for your organisation and your risks?
  - Which types of incident would have the most impact on your organisation (these should inform the maximum parameters set around incident response)?
  - What are the applicable legal requirements?
  - What would you need from the CSP in the case of such incidents?
    - To respond adequately?
    - To fulfil legal requirements?
    - To resolve any disputes arising from an incident?
- How is incident response defined? (Check this against the results of the previous question).
  - Do you define a maximum response time?
  - Do you classify the severity of incidents?
  - Are there incident reporting channels for your users?
  - What information would you need to be reported to you in the case of provider-detected incidents (e.g. data breach, availability, provisioning, management console)?
  - Do you define a maximum or maximum average incident response time for the provider?

- What metrics and reporting are in place to support monitoring incident response?
    - Does the provider report incident response statistics (e.g. time to respond and severity)?
    - Does the provider carry out regular tests of its incident response procedures? If so, do they report the testing frequency to you?
- What can I do to monitor incident response performance from the customer perspective?
    - Do you keep independent logs of incident reports and response times?
- Your on-going responsibilities:
    - Do you have in place incident detection and response capabilities for the systems/layers you control?
    - How does the provider's incident classification scheme map to your own?
- Are there penalties defined for incident response – if so, what are the thresholds at which they are set (e.g. minimum average response time over a month)?

3. Service elasticity and load tolerance

- What does elasticity mean for your organisation and your risks?
    - How much does the load on various aspects of your service change over different timescale- second/hour/day/week/month/year?
    - How volatile/unpredicatable is the load on your service (it can change but be predictable)?
    - Are denial of service attacks an important consideration?
    - Which types of resource (storage, network, processing etc...) are subject to the most volatility?
    - How critical are the service functions which are subject to the most volatility?
    - Do you require reserve capacity or sudden increases in loading on certain systems in business continuity scenarios?
    - Do you have any requirements to limit resource provisioning (e.g. for financial reasons)?
    - Do you have reserved capacity tested which is sometimes not used?
- How is elasticity defined? (Check this against the results of the previous question).
    - For which resources (if any) should elasticity be monitored?
    - Do you want to be alerted if resource provisioning fails- if so, according to what criteria?
- What metrics and reporting are in place to support monitoring elasticity?
    - Does the provider report on regular load testing?
    - Does the provider offer information on availability of additional capacity (giving the customer visibility of the CSP's ability to accommodate bursts in customer demand)?

- Does the provider regularly test resource limiting features?

- How can you monitor elasticity from the customer perspective?
  - Do you carry out provisioning (e.g. burst) tests within allowed resource limits?
  - Do you carry out regular tests of reserved capacity?
  - Do you keep logs of failed provisioning requests?
  - Do you regularly test resource limiting features?
- Your responsibilities:
  - Have you considered your elasticity requirements in the architectural choices made by your team (e.g. for IaaS building in auto-scaling rules)?
- Are there penalties defined for provisioning failures – if so, what are the thresholds at which they are set?

4. Data lifecycle management

- What does data life-cycle management mean for your organisation and your risks?
  - Which classes of data are processed by the service and how mission-critical are they?
  - How sensitive is your organisation and the specific service use-case to data loss (unrecoverable data loss) and leakage (unauthorised sharing and access)?
  - How important is the continuity of the service for your use-case?
  - How quickly will you need back-ups in the case of data loss (for various service functions)?
  - How recent do back-ups need to be (data recovery point)?
  - How likely is it that you will decide/need to switch to another CSP?
  - What regulatory requirements affect your data life-cycle management?
  - How much do your users care about government access to their data?
- How is data life-cycle management defined? (Check this against the results of the previous question).
  - Does the provider report the frequency of back-up tests?
  - Does the provider specify a maximum age for data restored from back-up?
  - Does the provider specify a minimum rate for data to be restored from back-up?
  - Does the provider specify that information will be provided on data loss or leakage detected?
  - Does the provider specify any maximum and minimum data retention period(s)?
  - Does the provider specify any reporting of government requests for data?
  - Does the provider specify any data durability commitments or reporting (data durability is defined as the percentage of information which will survive a specified period if not backed up)?
- What metrics and reporting are in place to support monitoring data life-cycle management?
  - Does the provider offer logs of back-up operations?
  - Does the provider report back-up test frequency and/or results?
  - Does the provider offer data loss prevention (DLP) functionality and reporting?

- Does the provider report on government requests for data- if so, how (per-user, service-wide)?

- How can you monitor data lifecycle management from the customer perspective?
  - Do you plan to test the availability and performance of back-ups independently (where customer-access is granted), for example
    - Sample back-up-data request?
    - Recovery-point tests?
    - Data deletion tests?
  - Do you plan any other tests of the effectiveness of data lifecycle management (e.g. DLP system tests)?
  - Do you plan to test data portability and export mechanisms, for example:
    - Availability of export functions?
    - Formats and integrity?
    - Export rate?
- Your responsibilities:
  - What design choices have you made which affect back-up and export. For example:
    - Data formats?
    - Redundancy?
    - Data handling practices for software layers under your control (e.g. deletion, subject access, etc.)?
    - Consistency of redundant data stores?
  - Do you have a procedure in place to deal with any regulatory requests for information?
- Are there penalties defined for data lifecycle management – if so, what are the thresholds at which they are set?

5. Technical compliance and vulnerability management:

- What does vulnerability detection and management mean for your organisation and risks?
  - What action, if any, could be taken by your organisation based on reports of vulnerabilities detected in systems controlled by the provider?
  - Which assets are most sensitive to vulnerabilities in the service?
  - What regulatory and compliance requirements apply to your organisation regarding the detection and management of vulnerabilities in the service (e.g., data breach notification)?
- How are technical compliance and vulnerability management defined? (Check this against the results of the previous question).
  - Does the provider define any of the following security baseline information?
    - Security-related configuration and options to be used?
    - System components covered by security controls and configuration elements?

- Schedule and information on software updates and patches to be applied?
- Criteria and procedures for vulnerability discovery and reporting, e.g. public vulnerability lists monitored in relation to COTS systems?

    - Which system components are covered by the above information security baseline and/or vulnerability detection and management processes?
    - Does vulnerability reporting address whether vulnerabilities result from the provider's software configuration or other aspects of the system configuration?

- What metrics and reporting are in place to support monitoring of vulnerability detection and management?
    - Does the provider report on any of the following in relation to the security baseline information:
        - Dynamic information on patches and controls in place?
        - Information on security-relevant configuration changes?
        - Reporting of relevant vulnerabilities and vulnerability trends according to the baseline reporting framework? For example, does the provider classify vulnerabilities by severity (e.g. using CVSS)? If so, is the value of affected assets included in this classification?
    - What reporting channels are in place for responsible disclosure of vulnerabilities by third parties?
- How can you monitor vulnerability detection and management from the customer perspective?
    - Do you plan to perform independent vulnerability testing of any system components (i.e. by your organisation or a third-party, where permitted)? If third-party testers are used, are they certified?
    - Does the agreement specify criteria for the performance of independent vulnerability tests or independent third-party vulnerability reporting? If not, have you verified with the provider if any conditions apply? (See responsibilities.)
    - Do you cross-check with public vulnerability disclosure sources (e.g. CVE) and/or independent abuse reports for issues affecting the provider's systems?
    - Are the appropriate measures in place to ensure confidentiality of vulnerability reports?
- Your responsibilities:
    - Do you have systems in place for technical compliance and vulnerability management for the system elements under your control (see table of responsibilities for IaaS, PaaS, SaaS)?
    - Do your processes comply with the provider's conditions for performing independent vulnerability assessments?
    - In your process for handling vulnerability reports, do you take into account the value of the assets affected (which is usually not available to the provider)?

    - Have you taken measures to ensure that your processes, systems, devices and activities are not the cause of any perceived vulnerability in the cloud provider's service(s)?
    - Do you have a procedure for alerting the provider to any vulnerabilities you detect whose remediation is within the provider's responsibility?
- Are there penalties defined for failures in technical compliance and vulnerability management – if so, how are they triggered?

6. Change management

- What does change management mean for your organisation and your risks?
    - What are the key dependencies of your organisation on specific configuration aspects of the provider which are subject to change, e.g. authentication mechanisms, features related to compliance?
    - What changes would you want the provider to notify you about in advance?
    - Which aspects of system configuration depend on your users, systems, devices, or processes?
- How is change management defined? (Check this against the results of the previous question).
    - Does the provider specify any notice periods for critical changes to system configuration?
    - Does the provider specify notification triggers for key changes?
- What metrics and reporting are in place to support monitoring of change management?
    - Is there a documented change management procedure in place, as well as tools to support the logging and reporting of changes?
    - Does the provider report the frequency of testing of change management procedures?
    - Are records kept of time to implement security-critical customer change requests e.g. to provision/de-provision/change access privileges.
    - Are notification triggers implemented for key events, such as (NB this is a checklist of possible triggers, not a list of must-haves):
        - ■Loss of certification status (ISO, FIPS etc.)?
        - ■Major changes in staff clearance status?
        - ■Changes or extension of jurisdictions in which data processing occurs.
        - ■Patches and major system changes which might affect the operation of dependent components not under the provider's control (e.g. changes in authentication systems)?
        - ■Significant changes in security controls and processes used, e.g. encryption key lengths and key management processes?
        - ■Any change affecting your certification requirements?
- How can you monitor change management from the customer perspective?
    - Do you plan to monitor system changes independently, where possible?
    - Do dependent software or services fail gracefully in the face of service changes?
- Your responsibilities:

- Is there a documented change management procedure in place, as well as tools to support the logging and reporting of changes for systems under your control?
- Have you taken steps to ensure that your users, systems, devices, and processes are not responsible for delays in provisioning/deprovisioning/changing system settings, including access privileges?

- Are there penalties defined for change management related failures – if so, how are they triggered?

7. Data isolation

- What does isolation mean for your organisation and your risks?
  - What sensitive data will be processed by the service? (Determines requirements for data isolation.)
- How is isolation defined? (Check this against the results of the previous question.)
  - Is data isolation included in penetration testing?
- What metrics and reporting are in place to support monitoring of isolation?
  - Which types of data isolation are tested as part of regular penetration tests, for example:
    - Memory?
    - Data at rest?
    - Data in transit?
    - Secure deletion?
- How can you monitor isolation from the customer perspective?
  - Do any penetration tests you carry out test for data isolation- e.g. checking for residual data in newly provisioned resources?
  - Can you detect any dependencies on other externally observable loads on the system, e.g. performance decreases at a particular time of day?
- Your responsibilities:
  - Have you made appropriate isolation-related design choices for elements of the system under your control, for example:
    - Implementing secure wipe processes for sensitive data?
    - Use of queues to isolate components from failures in other components?
    - Encryption for data at rest and in transit?
    - Secure key management?
- Are there penalties defined for isolation – if so, what are the thresholds at which they are set?

8. Log management and forensics

- What do log management and forensics mean for your organisation and your risks?
  - When will log information related to the service be critical to your organisation, for example:
    - Incident reporting?

- Dispute resolution?
- Law enforcement requests?
- Service level monitoring?

- What logging performance and availability levels do you require in the above situations?
- What type of evidence is required for logs to be useful in the above situations (i.e. how important are non-repudiation systems and other sources of evidence)?
- How likely is it that your organisation will be subject to a law-enforcement or freedom of information request involving data stored in the service?

- How are log management and forensics defined? (Check this against the results of the previous question.)
  - Does the agreement specify an availability commitment for logging systems? If so, does this include:
    - Access times for read-only access?
    - Availability for write-access?
    - Availability of non-repudiation systems e.g. time-stamping, third-party logging etc.?
  - Does the agreement include a commitment to respond, within a certain timeframe, to extraordinary requests for information regarding the use of the customer's system?
- What metrics and reporting are in place to support monitoring of log management and forensics?
  - Are logs tested frequently for availability?
  - Does the provider test the log accuracy and correct functioning of non-repudiation systems (e.g. using sample events)?
  - If so, are the frequency and/or results of these tests reported? (This may not be appropriate, depending on the service model.)
- How can you monitor log management and forensics from the customer perspective?
  - Do you perform cross-checks with your own event-logging systems (e.g. firewall logs)?
- Your responsibilities:
  - Do you log relevant events in the systems under your control?
  - Do you test the logging systems under your control regularly?
- Are there penalties defined for log management and forensics – if so, what are the thresholds at which they are set?

ANNEX Contractual considerations from ENISA 2009 Risk Assessment

1. **Data protection:** attention should be paid to choosing a processor that provides sufficient technical security measures and organisational measures governing the processing to be carried out, and ensuring compliance with those measures
2. **Data security:** attention should be paid to mandatory data security measures that potentially cause either the cloud provider or the customer to be subject to regulatory and judicial measures if the contract does not address these obligations.
3. **Data transfer:** attention should be paid to what information is provided to the customer regarding how data is transferred within the cloud provider's proprietary cloud, outside that cloud, and within and outside the European Economic Area.
4. **Law enforcement access:** each country has unique restrictions on, and requirements providing for, law enforcement access to data. The customer should pay attention to information available from the provider about the jurisdictions in which data may be stored and processed and evaluate any risks resulting from the jurisdictions which may apply.
5. **Confidentiality and non-disclosure:** the duties and obligations related to this issue should be reviewed.
6. **Intellectual property:** in the case of IaaS and PaaS, intellectual property, including original works created using the cloud infrastructure, may be stored. The cloud customer should ensure that the contract respects their rights to any intellectual property or original works as far as possible without compromising the quality of service offered (e.g. back-ups may be a necessary part of offering a good service level).
7. **Risk allocation and limitation of liability:** when reviewing their respective contract obligations, the parties should underscore those obligations that present significant risk to them by including monetary remediation clauses, or obligations to indemnify, for the other party's breach of that contract obligation. Furthermore, any standard clauses covering limitations of liability should be evaluated carefully.
8. **Change of control:** transparency concerning the cloud provider's continuing ability to honour their contract obligations in the case of a change of control, as well as any possibility to rescind the contract.

Glossary of acronyms

Acronym	Full Text
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CESG	Communications-Electronics Security Group
CHECK	UK CESG IT Health Check Service
CIO	Chief Information Officer
CISO	Chief Information Security Officer
COTS	Commercial off-the-shelf
CREST	Council of Registered Ethical Security Testers
CSP	Cloud Service Provider
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DDoS	Distributed Denial of Service
DLP	Data Loss Prevention
FedRAMP	Federal Risk and Authorisation Management Programme
FISMA	Federal Information Security Management Act (US)
FIPS	Federal Information Processing Standards (US)
IaaS	Infrastructure as a Service
IETF	Internet Engineering Task Force
ISAE	International Standard on Assurance Engagements
ISCM	Information Security Continuous Monitoring
ISO	International Organisation for Standardization
ITIL	Information Technology Infrastructure Library
MRT	Mean Recovery Time
MTBF	Mean Time Between Failures
NIST	National Institute of Standards and Technology (US)
OS	Operating System
PaaS	Platform as a Service
PCI	Payment Card Industry security standards council
RAID	Redundant Array of Independent Disks
RFC	Request For Comments (standard format of IETF documents)
RfP	Request for Proposals
RIPA	Regulation of Investigatory Powers Act (UK)
RPO	Recovery Point Objective
RTO	Recovery Time Objective
SSAE	Statements on Standards for Attestation Engagements
SaaS	Software as a Service
SLA	Service Level Agreement
TLS	Transport Layer Security
VM	Virtual Machine
WORM	Write once, read many (non-repudiation technology)

Bibliography

1. Interxion Europe-wide survey of current and intended cloud usage and attitudes towards cloud computing. [Online] http://www.interxion.com/cloud-insight/index.html.

2. Neelie Kroes Vice-President of the European Commission responsible for the Digital Agenda Setting up the European Cloud Partnership World Economic Forum Davos, Switzerland, 26th January 2012 . [Online] http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/12/38&format=HTML&aged=0&language=EN&guiLanguage=en.

3. Cloud Computing: Benefits, Risks and Recommendations for Information Security. [Online] https://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment.

4. ENISA Cloud Computing Information Assurance Framework. [Online] 2009. http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-information-assurance-framework.

5. **ENISA.** Security and Resilience in Governmental Clouds. [Online] 2011. http://www.enisa.europa.eu/activities/risk-management/emerging-and-future-risk/deliverables/security-and-resilience-in-governmental-clouds/.

6. —. Survey and analysis of security parameters in cloud SLAs across the European public sector. [Online] 2011. http://www.enisa.europa.eu/activities/application-security/test/survey-and-analysis-of-security-parameters-in-cloud-slas-across-the-european-public-sector.

7. ISO 2700x series. [Online] http://www.27000.org/.

8. SSAE 16 Auditing Standard. [Online] 2011. http://www.ssae-16.com/.

9. International Standard on Assurance Engagements (ISAE) 3402. [Online] http://www.ifac.org/sites/default/files/downloads/b014-2010-iaasb-handbook-isae-3402.pdf.

10. Common Assurance Maturity Model. [Online] http://common-assurance.com/.

11. Cloud Security Alliance Cloud Controls Matrix. [Online] https://cloudsecurityalliance.org/research/ccm/.

12. Guidelines on information security controls for the use of cloud computing services. [Online] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43757.

13. NIST Definition of Cloud Computing - NIST SP 800-145. [Online] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

14. **NIST.** Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations. [Online] 2011. http://csrc.nist.gov/publications/nistpubs/800-137/SP800-137-Final.pdf.

15. **Government, US.** Federal Risk and Authorization Management Programme - Concept of Operations (Includes a section on continuous monitoring). [Online] 2012. http://www.gsa.gov/graphics/staffoffices/FedRAMP_CONOPS.pdf.

16. Annual FISMA Reporting: Chief Information Officer Questions. [Online] 2009. http://www.whitehouse.gov/sites/default/files/omb/assets/memoranda_fy2009/cio_questions.pdf.

17. **Budget, US Office of Management and.** Fiscal Year 2010 Report to Congress on the Implementation of the FISMA. [Online] 2010. http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/FY10_FISMA.pdf.

18. **ENISA.** Resilience Metrics and Measurements: Technical Report. [Online] 2011. http://www.enisa.europa.eu/activities/res/other-areas/metrics/reports/metrics-tech-report.

19. **IETF.** RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels. [Online] 1997. http://www.rfc-editor.org/rfc/rfc2119.txt.

20. ITIL - incident definition. [Online] http://www.itlibrary.org/index.php?page=Incident_Management.

21. ITIL Availability Management. [Online] 2006. http://www.cmg.org/measureit/issues/mit33/m_33_1.html.

22. Common Vulnerability Scoring System. [Online] http://www.first.org/cvss.

23. Bugtraq mailing list. [Online] http://www.securityfocus.com/archive/1.

24. Ful disclosure mailing list. [Online] http://seclists.org/fulldisclosure/.

25. CHECK - IT Health Check Service. [Online] http://www.cesg.gov.uk/servicecatalogue/CHECK/Pages/WhatisCHECK.aspx.

26. Council of Registered Ethical Security Testers. [Online] http://www.crest-approved.org/.

27. Commission decisions on the adequacy of the protection of personal data in third countries. [Online] http://ec.europa.eu/justice/policies/privacy/thridcountries/index_en.htm.

28. CVSS Calculator. [Online] http://nvd.nist.gov/cvss.cfm?calculator.

29. Google transparency report. [Online] http://www.google.com/transparencyreport/governmentrequests/.

30. Fiscal Year 2010 Report to Congress on the Implementation of the FISMA. [Online] http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/FY10_FISMA.pdf.

31. CHECK. [Online] http://govcertuk.cesg.gov.uk/products_services/iacs/check/index.shtml.

32. Fedramp continuous monitoring. [Online] http://www.gsa.gov/graphics/staffoffices/FedRAMP_CONOPS.pdf.

33. European IT decision-makers and influencers give their views on cloud computing. [Online] http://www.interxion.com/cloud-insight/index.html.

34. Resilience Metrics and Measurements: Technical Report . [Online] http://www.enisa.europa.eu/activities/res/other-areas/metrics/reports/metrics-tech-report.

35. Survey and analysis of security parameters in cloud SLAs across the European public sector. [Online] http://www.enisa.europa.eu/activities/application-security/test/survey-and-analysis-of-security-parameters-in-cloud-slas-across-the-european-public-sector.

36. Survey and analysis of security parameters in cloud SLAs across the European public sector . [Online] http://www.enisa.europa.eu/activities/application-security/test/survey-and-analysis-of-security-parameters-in-cloud-slas-across-the-european-public-sector.

37. Security and Resilience in Governmental Clouds . [Online] http://www.enisa.europa.eu/activities/risk-management/emerging-and-future-risk/deliverables/security-and-resilience-in-governmental-clouds/.

38. Council of Registered Ethical Security Testers. [Online] http://www.crest-approved.org/.

39. ENISA Cloud Computing Information Assurance Framework. [Online] http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-information-assurance-framework.

40. Cloud Computing: Benefits, Risks and Recommendations for Information Security. [Online] https://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment.

P.O. Box 1309, 71001 Heraklion, Greece
www.enisa.europa.eu