CC in the Cloud Guidance for Cloud Evaluations

Preface

The CC in the Cloud Technical Work Group (CCitC) is developing an Essential Security Requirements (ESR) document that captures the fundamental requirements for evaluating IT products applicable to CC which operate in Cloud environments. Such cloud environments may or may not conform to hybrid, public, or private cloud topologies and the associated Cloud Service Providers. This effort is not meant to replace solution or service oriented frameworks or certification processes. The initial draft of this document contains material that was created by the TWG as a framework to utilize Common Criteria methodologies for products operating in a cloud environment.

Objective

This document defines cloud evaluations and certification terminology and describes appropriate advice.

The main intended audience of this document are evaluation sponsors and manufacturers, but it is also related to evaluators, certifiers and end users of this type of products.

Cloud Service and Product Presentation and Definitions

Glossary

The following definitions are used throughout the document. It is important that each term be clearly understood in order that guidance documentation for the evaluation process be put in context.

Software as a Service (SaaS)

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings. [nist_cloud]

The following PPs are examples which might be extended with CCitC methodolgy to cover the above use case: cPP_App_SW, cPP_DBMS, PP_MDM

For example, if the cPP for Application Software were to be used as a baseline the cloud extensions may be applied to the existing TOE Boundary and TOE Platform given in the following diagram:

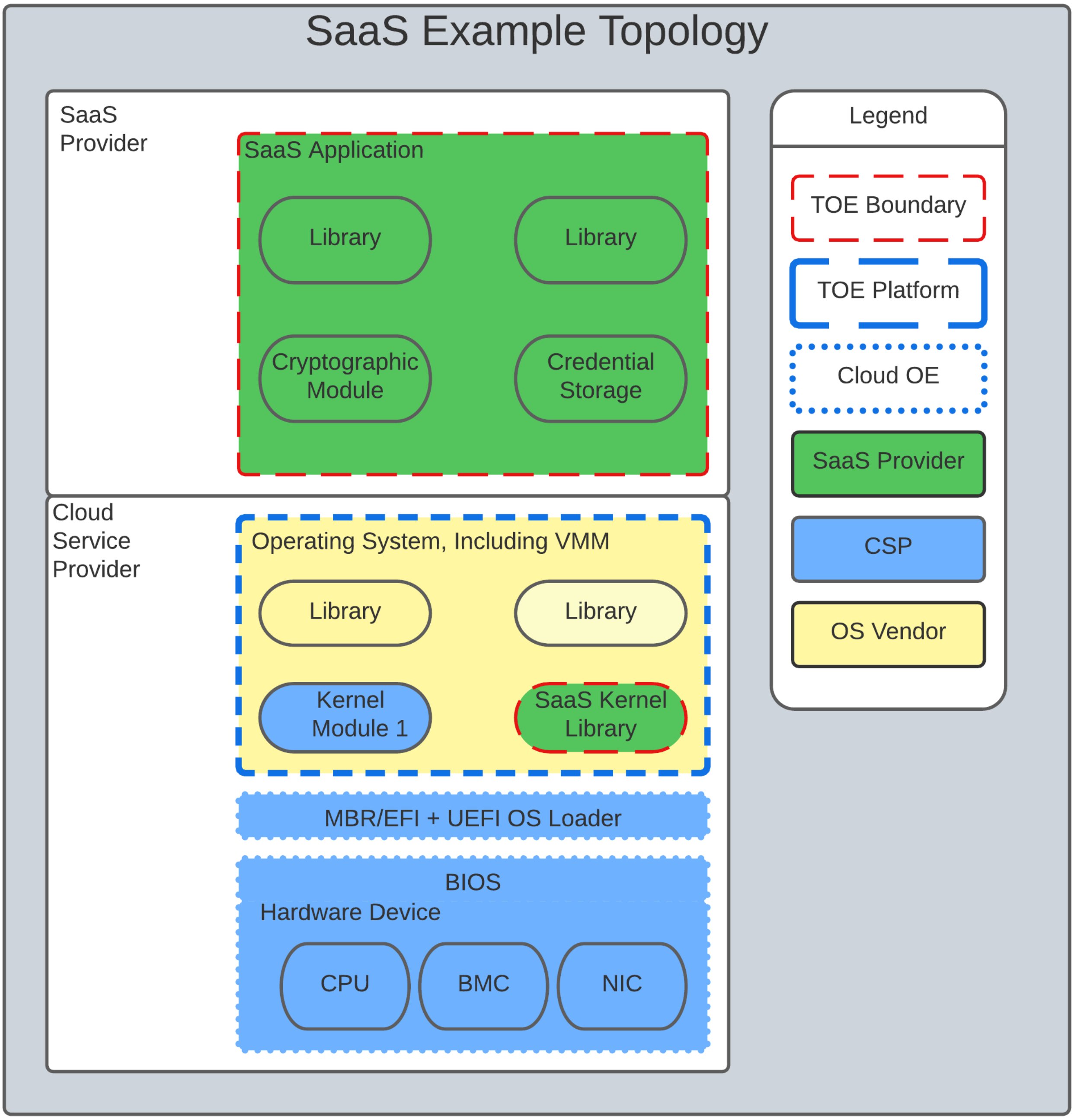

Figure 1. SaaS Example

In this example, the SaaS Application provided by the SaaS provider relies on a TOE platform from an OS Vendor which is hosted by the Cloud Service Provider on the CSPs hardware. In this Cloud evaluation scenario, additional TSS and Assurance Activities could be prescribed to expand the evaluated configuration in a Cloud Operating Environment.

Platform as a Service (PaaS)

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. [nist_cloud]

The following PPs are examples which might be extended with CCitC methodolgy to cover the above use case: GP_OS_PP, cPP_ND

For example, if the Protection Profile for General Purpose Operating System were to be used as a baseline the cloud extensions may be applied to the existing TOE Boundary and Cloud Cloud Operating Environment given in the following diagram:

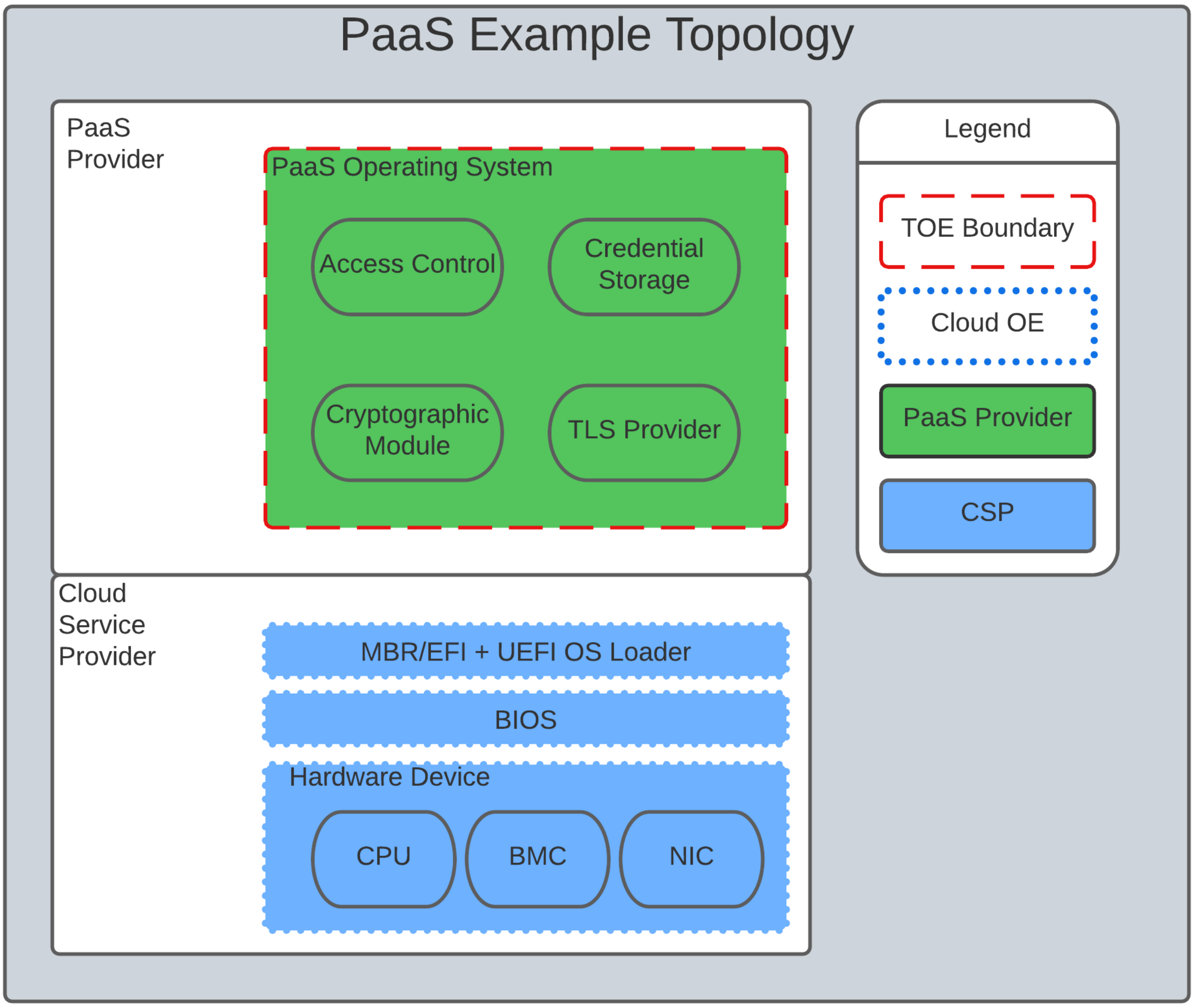

Figure 2. Paas Example

In this example, the PaaS OS provided by the PaaS provider relies on a Cloud Operating Environment which is hosted by the Cloud Service Provider on the CSPs hardware. In this Cloud evaluation scenario, additional TSS and Assurance Activities could be prescribed to expand the evaluated configuration in a Cloud Operating Environment.

Infastructure as a Service (Iaas)

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls) [nist_cloud]

The following PPs are examples which might be extended with CCitC methodolgy to cover the above use case: PP_BASE_VIRTUALIZATION

For example, if the Protection Profile for Virtualization were to be used as a baseline the cloud extensions may be applied to the existing TOE Boundary and Cloud Cloud Operating Environment given in the following diagram:

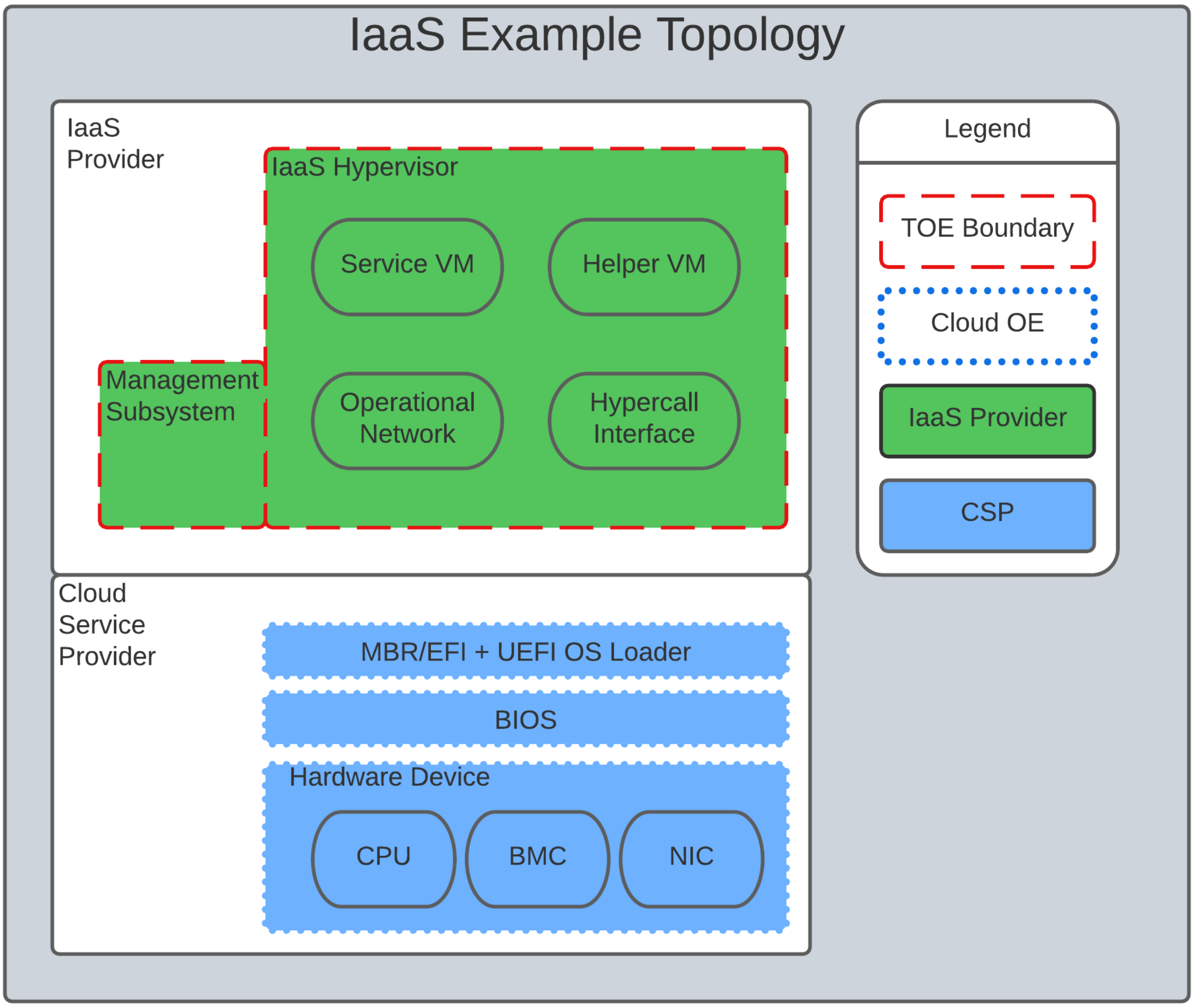

Figure 3. IaaS Example

In this example, the IaaS hypervisor provided by the IaaS provider relies on a Cloud Operating Environment which is hosted by the Cloud Service Provider on the CSPs hardware. In this Cloud evaluation scenario, additional TSS and Assurance Activities could be prescribed to expand the evaluated configuration in a Cloud Operating Environment.

On-Premises (On-Prem)

Please log definition as github issue. These should be contextual with CCiTC. ==== Cloud Service Provider (CSP) Please log definition as github issue. ==== Trusted Cloud Service Provider (TCSP) Will define as a group. ==== Cloud Authorization Scheme Please log definition as github issue.

Architecture

Life-cycle Presentation

Dev-Ops / Agile

It may be pertinent to discuss the primary pillars of cloud service deployments here using Dev Ops Methodology at a high level. This will later be used to support use cases most likely to leverage these practices such as SaaS.

Waterfall

In contrast to Agile methodology, the majority of PaaS or IaaS use cases are more likely to embrace a waterfall methodology for large systems such as public cloud deployments. This information will tie into further concepts below.

Contributors Roles in Product Evaluations

Roles Clarification

TOE Developer

Pull from JIL template

Cloud Service Provider (CSP)

Will define role as TC.

Sponsor

Pull from JIL template

Evaluator

Pull from JIL template

Certification Body (CB)

Pull from JIL template

Cloud Authorization Scheme

Will define role as TC.

End User

Pull from JIL template

Contributors Involvement

This section would be useful for explaining the relationships possible between the TOE Vendor, CSP, and Trusted Platform.

Theoretical Planning for a Cloud Evaluation

This section in the JIL guidance essentially outlines the way in which CC subprocess can be worked in parallel. It could be eliminated from Cloud Guidance as this is already well defined and understood.

Cloud Sub-Processes

Introduction

This section in the JIL guidance is meant to help vendors "anticipate their development capability to comply to the requirements of CC". For the CCiTC purposes, this should provide guidance to PP authors and Evaluators how to augment CC deliverables for cloud evaluations.

Development Environment Sub-Process

It may be necessary for the iTC to provide additional assurance strategies for PP authors to incorporate more development environment review. The goal of which is to support Dev Ops practices and facilitate cloud evaluations that allow the end user some assurance apart from fixed version evaluations.

For example if a TOE vendor is leveraging a cloud providers container images to deplay a software product, the Development Environment sub-process should require additional collateral to demonstrate that the cloud provider is providing assurance to the TOE vendor for the image that it is appropriately signed and/or updated.

FEDRamp PaaS OS images can be assumed to receive continous vulnerability patching to maintain authorization and that collateral can be extended to CCiTC.

CNAS-4: CI/CD pipeline & software supply chain flaws

Examples:

- Insufficient authentication on CI/CD pipeline systems
- Use of untrusted images
- Use of stale images
- Insecure communication channels to registries
- Overly-permissive registry access
- Using a single environment to run CI/CD tasks for projects requiring different levels of security

Security Target (TSS) Sub-Process

This section shall discuss the expected changes to a Security Target for cloud evaluations. This also may be better suited to address SFR changes needed in a PP. For example, changes need to address SFRs that deal with credential management:

For example, for all Assurance Activities that extend TOE Summary Specification (TSS) requirements in a ST will need to be modified such that Cloud Operating Environments are captured.

Due to the fungible nature of compute resourcing in a Cloud Environment, traditional methods of credential storage to a local operating environment are not viable. Cloud IAM, Token, and Key management are often adapted to Cloud Products and these changes must be reflected in the associated TSS requirements.

CNAS-5: Insecure secrets storage

Examples:

- Orchestrator secrets stored unencrypted
- API keys or passwords stored unencrypted inside containers
- Hardcoded application secrets
- Poorly encrypted secrets (e.g. use of obsolete encryption methods, use of encoding instead of encryption, etc.)
- Mounting of storage containing sensitive information

SFRs deailing with protected communications:

CNAS-6: Over-permissive or insecure network policies

Examples:

- Over-permissive pod to pod communication allowed
- Internal microservices exposed to the public Internet
- No network segmentation defined
- End-to-end communications not encrypted
- Network traffic to unknown or potentially malicious domains not monitored and blocked

SFRs dealing with auditing:

Guidance Documentation Sub-Process

This section shall discuss the increased requirements of product configuration in cloud environments.

It is important to distinguish here that not all expected elements of a traditional AGD document can be translated for Cloud Environments. It may be necessary to exclude or supplement these guidance requirments depending on the topology of the product and the cloud service provider. In some circumstances, the Cloud Provider is the only entity that may fulfill these guidance requirements to ensure that the TOE is deployed in the tested configuration. If the CSP is a Trusted Provider, this exclusion may be minimized.

For example, auditing an on-prem resources versus an equivalent resource deployed to in a cloud environment presents a number of additional considerations. When developing guidance requirements, PP Authors must consider the following common pitfalls for auditing servcies in Cloud Environments and tailor appropriately as needed for the product technology type and Cloud topology:

- No container or host process activity monitoring
- No network communications monitoring among microservices
- No resource consumption monitoring to ensure availability of critical resources
- Lack of monitoring on orchestration configuration propagation and stale configs

CC Guide Modifications:

- Installation Guidance
    - Typically CC Guidance containts instructions on how to configure the TOE exactly as tested. Due to the ethereal nature of cloud platforms, this is often untennable for a variety of reasons. As such, CC guidance must be extended to ensure that the fundamental controls to deploy the TOE in a cloud environment securely are captured.The following items will focus on particular areas of concern.
- Crytpo Config

        - D@RE
            - KMS/HMS Config
        - DiT
- Setting Time / Time Zone for Clound Tenancy
- Audit config
- Cloud Dependencies
    - Platform Configuration
        - This should explain how your cloud platform management plane is set up to allow for the TOE (esxi config/kubernetes config/etc)
            - Container Orchestration settings
        - Containerization Settings
            - Privledge Container Settings
            - Network bridge settings
        - Platform Isolation
    - Env. Variables
    - Network Resource Configuration
        - Network Isolation
    - Data Storage Resource Configuration
        - DB or Storage Isolation
    - Token generation
- Uninstallation/Removal Guidance
    - Destruction of Secrets
    - Data santization

This section may need a table to map the list items into a table for SaaS / PaaS / IaaS.

CNAS-1: Insecure cloud, container or orchestration configuration

Examples:

- Publicly open cloud storage buckets
- Imrpoper permissions set on cloud storage buckets
- Container runs as root
- Container shares resources with the host (network interface, etc.)
- Insecure Infrastructure-as-Code (IaC) configuration

CNAS-3: Improper authentication & authorization

Examples:

- Unauthenticated API access on a microservice
- Over-permissive cloud IAM role
- Lack of orchestrator node trust rules (e.g. unauthorized hosts joining the cluster)
- Unauthenticated orchestrator console access
- Unauthrized or overly-permissive orchestrator access

CNAS-10: Ineffective logging & monitoring (e.g. runtime activity)

Examples:

- No container or host process activity monitoring
- No network communications monitoring among microservices
- No resource consumption monitoring to ensure availability of critical resources
- Lack of monitoring on orchestration configuration propagation and stale configs

Development / Tests Sub-Process

This section shall discuss augmentations needed for assurance activities that are targetting cloud evaluations.

Penetration Testing Methodology

This section shall discuss modifications to AVA activities for cloud evaluations.

CNAS-2: Injection flaws

Examples:

- SQL injection
- XXE
- NoSQL injection
- OS command injection
- Serverless event data injection

List of potentional vulnerabilities

Since a cloud stack will inherently contain many vulnerabilities, it would be useful for the iTC to provide guidance on risk managment practices to minimize these factors.

CNAS-7: Using components with known vulnerabilities

Examples:

- Vulnerable 3rd party open source packages
- Vulnerable versions of application components
- Use of known vulnerable container images

CNAS-8: Improper assets management

Examples:

- Undocumented microservices & APIs
- Obsolete & unmanaged cloud resources

CNAS-9: Inadequate ‘compute’ resource quota limits

Examples:

- Resource-unbound containers
- Over-permissive request quota set on APIs

Defining penetration tests

The JIL Guidance uses this section on how to compose the penetration testing coverage needed. This may need to be expanded given the content above or removed entirely from the CCiTC guidance.

List of attacks and strategies

The JIL Guidance uses this section to allow for attacker potential mitigations. The iTC will need to provide some language here to adapt for cloud evaluations.

References