Security Architecture and Operating Model

`2021.1`

In the digital age, cyber attacks are inevitable. At MiCT, we are taking a "zero trust", "minimal infrastructure" approach to managing risk and information security.

This document describes our guiding principles and aspirations in managing risk and the building blocks of our security model.

Policy Statements

MiCT policy requires that:

(a) MiCT's security program and operations should be designed and implemented with the following objectives and best practices:

- data-centric, cloud-first
- assume compromise therefore never trust, always verify
- apply controls using least-privilege and defense-in-depth principles
- avoid single point of compromise
- automate whenever possible, the simpler the better, less is more
- prompt self management and reward good behaviors

(b) Security shall remain a top priority in all aspects of MiCT's business operations and product development.

Controls and Procedures

MiCT Security Principles

(1) Data-centric model; zero-trust architecture "Zero Trust" is a data-centric security design that puts micro-perimeters around specific data or assets so that more granular rules can be enforced. It remedies the deficiencies with perimeter-centric strategies and the legacy devices and technologies used to implement them. It does this by promoting "never trust, always verify" as its guiding principle. This differs substantially from conventional security models which operate on the basis of "trust but verify."

In particular, with Zero Trust there is no default trust for any entity — including users, devices, applications, and packets—regardless of what it is and its location on or relative to the corporate network. In addition, verifying that authorized entities are always doing only what they're allowed to do is no longer optional; it's now mandatory.

!!! Summary

```
* No internal network. (Almost) 100% cloud.
* Fully segregated with Granular policy enforcements.
* Individually secured devices. No production access by default.
```

(2) "Air-Gapped" environments meet short-lived processes We extend the zero-trust security model with a "Minimal Infrastructure" approach, where we use "Anything-as-a-Service" whenever possible, to harness the full power of the cloud. Cloud services allow us to contain and control access at a much more granular level, compared to operating on-premise infrastructure. Via access to the extensive APIs provided by the cloud services, we would be able to more easily integrate and automate security operations. Additionally, minimizing infrastructure significantly reduces always-on attack surfaces. Services that are not used are turned off, instead of being idly available which opens itself up to attacks. Together with Zero Trust, this security model and architecture enables a high degree of flexibility for end-user computing while maintaining the highest level of security assurance.

!!! Summary

```
* No direct administrative or broad network connectivity into production.
* Processes are short-lived and killed after use.
* Minimal persistent attack surface making it virtually impenetrable.
```

(3) Least-privilege temporary access Cyber attacks are inevitable. When it comes to preparing for potential attacks, MiCT security operations take the approach that assumes a compromise can happen at any time, to any device, with little to no indicators. This is also an extension of the "zero trust" model. When building security operations, we carefully perform risk analysis and threat model, to identify potential single point of compromise and to avoid having the "keys to the kingdom".

In other words, compromise of any single system or user or credential, should not easily lead to a broad or full compromise of the entire infrastructure or operations. For example, if an attacker gains access to a admin credential (e.g. Active Directory domain), it should not directly lead to the compromise of all systems and data in the environment.

!!! Summary

```
* Need-based access control for both employees and computing services.
* Access to critical systems and resources are closed by default, granted on demand.
* Protected by strong multi-factor authentication.
* No "keys to the kingdom"; no single points of compromise.
* "Secrets" (such as SSH Keys) must remain secret at all times.
```

(4) Immutable builds and deploys The MiCT platform leverages a microservice architecture. This means that the system has been decomposed into numerous small components that can be built and deployed individually. Before these components get deployed to our *production* environments, we thoroughly test and validate the changes in our *lower* environments which are completely isolated from production. This allows us to test upcoming changes while ensuring there is no impact to our customers.

As a particular build of a component progresses through our environments, it is important that the build does not change thus we ensure that each build is immutable. Once an *immutable build* has been validated in our *lower* (non-production) environments, we then deploy it to our *production* environment where the change will be available to MiCT customers and end-users.

Changes to our infrastructure (database schema changes, storage buckets, load balances, DNS entries, etc.) are also described in our source code and deployed to our environments just like the applications. This architectural approach to managing infrastructure is referred to as *infrastructure as code* and is a key requirement for fully automated deployments with minimal human touch.

!!! Summary

```
* Infrastructure as code with active protection.
* Automated security scans and full traceability from code commit to production.
* "Hands-free" deployment ensures each build is free from human error or malicious contamina
```

(5) End-to-end data protection and privacy It is of the utmost importance that MiCT provides for confidentiality (privacy), integrity and availability of its customer's data. Your data is protected with end-to-end encryption, combined with strong access control and key management. We also prohibit our internal employees to access customer data directly in production. So your data remains safe and private at all times. We will never use or share your data without your prior consent.

We are proud to offer our customers data storage peace of mind with a money-back guarantee. We guarantee your private data stored on our platform is always safe and protected from cyberattacks such as ransomware, and we will reimburse you for certain losses of such data due to unauthorized activity in eligible accounts that resulted through no fault of your own.

!!! Summary

```
* Data is safe both at rest and in transit, using strong encryption, access control and key
* No internal user access is allowed to customer data in production.
```

(6) Strong yet flexible user access We all know by now that "Passw0rd" makes a terrible password. Access control is so important we must get it right. That's why we leverage tried-and-true technology such as SAML, OAuth, multi-factor authentication, and fine-grained authorization to provide strong yet intuitive access options, both for our internal staff to access business resources and for our customers to access MiCT platform and services.

!!! Summary

```
* OAuth 2.0, OpenID Connect, SAML for customer authentication and single sign-on.
* Multi-factor authentication.
* Fine-grain attribute-based or role-based authorization.
```

(7) Watch everything, even the watchers You can't protect what you can't see.

As the famous strategist, Sun Tzu, once said, "Know thy self, know thy enemy. A thousand battles, a thousand victories." It all starts with knowing ourselves. This applies to the infrastructure, environments, operations, users, systems, resources, and most importantly, data. It is important to inventory all assets, document all operations, identify all weaknesses, and visualize/understand all events.

This includes conducting various risk analysis, threat modeling, vulnerability assessments, application scanning, and penetration testing. Not only that, this requires security operations to keep an eye on everything, and someone should also "watch the watchers".

At first, this would require significant manual effort and may seem impossible to keep up-to-date. Our goal is to automate security operations, so that this can be achieved programmatically as our operations evolve to become more complex.

Additionally, MiCT security team will actively monitor threat intelligence in the community, with feeds and information sharing platform such as NH-ISAC to stay abreast of the attacker activities and methodologies.

!!! Summary

```
* All environments are monitored; All events are logged; All alerts are analyzed; All assets
* No privileged access without prior approval or full auditing.
* We deploy monitoring redundancy to "watch the watchers".
```

(8) Centralized and automated operations As much as possible, MiCT security will translate policy and compliance requirements into reusable code for easy implementation and maintenance. This allows us to truly be able to enforce policy and compliance in a fast and scalable way, rather than relying solely on written policies and intermittent manual audits. For example, endpoint device policies may be translated into Chef InSpec code and compliance may be enforced through the agent. Access Control policies for production environments are translated into AWS IAM JSON policies and implemented via Terraform code.

Automation makes it truly possible to centralize security operations, including not only event aggregation and correlation, but also the orchestration and management of previously siloed security controls and remediation efforts.

!!! Summary

```
* API-driven cloud-native security fabric that
    - centrally monitors security events,
    - visualizes risk management,
    - automates compliance audits, and
    - orchestrates near real-time remediation.
```

(9) Usable security Security benefits from transparency, and should operate as an open-book. This allows the entire organization to take responsibility for and accountability of adopting security best practices. Similar to code reviews and pull requests in the development process, MiCT security team makes security standards and practices available to all employees for feedback prior to adoption.

We emphasize on the usability and practicality of security. A security solution or process is not effective, if it is not being used, no matter how good it may be. Having impractical security would only generate noise, provide a false sense of security, and incur unnecessary cost. Nothing is perfect, but we embrace an agile mindset to test and try, and to continuously improve.

!!! Summary

```
* All employees receive security awareness training not annually, but monthly.
* Simple policies, processes, and procedures.
* No "Shadow IT".
* DevSecOps with common goals and an integrated team.
* Processes that encourage self management and reward good behavior.
```

(10) Regulatory compliant and hacker verified Security != Compliance. We cannot have one without the other.

!!! Summary

```
* Regulatory Compliant;
* Independently assessed and certified;
* Hacker verified.
```

Security Architecture

MiCT developed a security architecture on top of its three main infrastructure environments - Cloud (AWS), DevOps, and workforce collaboration / end-user computing.

Architecture Diagrams Detailed architecture diagrams of the in-scope networks, endpoints, applications as well as the security operations are developed and maintained by JupiterOne.

Cloud Architecture

Cloud Native

- Designed for the cloud using true multi-tenant architecture
- Auto scaling across multiple data centers in multiple regions around the world

- MiCT services deployed inside private subnets of Virtual Private Cloud (VPC)
- Comprehensive security and compliance via AWS certifications
- Ongoing security testing by AWS and AWS customers

Customer Benefits

- Infrastructure is tailored to our customer's goals and usage patterns
- "Shared use" model reduces cost
- Nearly infinite compute and data capacity via AWS cloud provider
- Customers can focus on solving business problems and not worry about infrastructure
- Automatic backup and recovery
- Continuous improvements via change control process
- Faster adoption of new technology

Evolution of Cloud Computing

1. Baremetal
    - A computer in someone else's data center
2. Virtual Machine
    - A portion of a computer in someone else's data center
    - In AWS, a Virtual Machine is created from Amazon Machine Image (AMI)
3. Container
    - A package of essential application libraries and code but not the core OS libraries - Simpler to scale a docker image because - No duplication of core OS processes (networking, filesystem, etc) - Typically a Docker container
4. Function
    - Just the application code that runs in a pre-built container

MiCT strives to leverage functions as the primary building blocks for our platform because:

- functions deploy more quickly than containers and virtual machines
- AWS automatically scales Lambda functions based on the number of incoming invocations
- they are short-lived processes which minimizes attack surface

Metrics, Measurements and Continuous Monitoring

A set of metrics / KPIs have been defined to assist in the measuring, reporting and optimizing the security program and the controls in place.

A security scorecard is produced every with updates to key metrics of the MiCT information security program, to measure its adoption and effectiveness.

The reports and scorecards are maintained by and can be accessed at JupiterOne.

Quality of Service

MiCT strives to provide a high quality of service to all of its customers. This is accomplished through a security architecture that encompasses all of MiCT's operations and provides high data confidentiality, integrity, and availability.

An overview of MiCT's architecture can be found in Security Architecture. MiCT uses a highly scalable cloud architecture to provide system quality at all times.

All systems are monitored and measured in real time as described in Application Service Event Recovery.

MiCT uses DevOps methodology as described in Software Development Process to ensure a smooth delivery process of all systems and applications.

Status for external facing, customer applications and systems is published at .