Configuration and Change Management

`2021.1`

MiCT standardizes and automates configuration management through the use of automation scripts as well as documentation of all changes to production systems and networks. Automation tools such as Chef, Salt, and Terraform automatically configure all MiCT systems according to established and tested policies, and are used as part of our Disaster Recovery plan and process.

Policy Statements

MiCT policy requires that:

(a) All production changes, including but not limited to software deployment, feature toggle enablement, network infrastructure changes, and access control authorization updates, must be invoked through approved change management process.

(b) Each production change must maintain complete traceability to fully document the request, including requestor, date/time of change, actions taken and results.

(c) Each production change must be fully tested prior to implementation.

(d) Each production change must include a rollback plan to back out the change in the event of failure.

(e) Each production change must include proper approval.

- The approvers are determined based on the type of change.
- Approvers must be someone other than the author/executor of the change.
- Approvals may be automatically granted if certain criteria is met. The auto-approval criteria must be pre-approved by the Security Officer and fully documented and validated for each request.

Controls and Procedures

Configuration Management Processes

1. Configuration management is automated using industry-recognized tools like Chef, Salt and Terraform to enforce secure configuration standards.
2. All changes to production systems, network devices, and firewalls are reviewed and approved by Security team before they are implemented to assure they comply with business and security requirements.
3. All changes to production systems are tested before they are implemented in production.
4. Implementation of approved changes are only performed by authorized personnel.

5. Tooling is used to generate an up to date system inventory.
    - All systems are categorized and labeled by their corresponding environment, such as *dev*, *test*, and *prod*.
    - All systems are classified and labeled based on the data they store or process, according to MiCT data classification model.
    - The Security team maintains automation which monitors all changes to IT assets, generates inventory lists, using automatic IT assets discovery, and services provided by each cloud provider.
    - IT assets database is used to generate the diagrams and asset lists required by the Risk Assessment phase of MiCT's Risk Management procedures
    - MiCT Change Management process ensures that all asset inventory created by automation is reconciled against real changes to production systems. This process includes periodic manual audits and approvals.
    - During each change implementation, the change is reviewed and verified by the target asset owner as needed.
6. MiCT uses the Security Technical Implementation Guides (STIGs) published by the Defense Information Systems Agency as a baseline for hardening systems.
    - Windows-based systems use a baseline Active Directory group policy configuration in conjunction with the DISA STIGs.
    - Linux-based systems use Red Hat Enterprise Linux STIG as a guideline for implementation.
    - EC2 instances in AWS are provisioned using only hardened and approved Amazon Machine Images (AMIs).
    - Docker containers are launched using only approved Docker images that have been through security testing.
7. All IT assets in MiCT have time synchronized to a single authoritative source.
    - On-premise systems are configured to point to an internal NTP server.
    - The internal NTP server and all AWS instances are pointing to the same set of ntp.org servers.
8. All frontend functionality (e.g. user dashboards and portals) is separated from backend (e.g. database and app servers) systems by being deployed on separate servers or containers.
9. All software and systems are required to complete full-scale testing before being promoted to production.
10. All code changes are reviewed to assure software code quality, while in development, to proactively detect potential security issues using pull-

requests and static code analysis tools. More details can be found in the *Software Release / Code Promotion* section.

Configuration Monitoring and Auditing

All infrastructure and system configurations, including all software-defined sources, are centrally aggregated to a configuration management database (CMDB) – JupiterOne.

Configuration auditing rules are created according to established baseline, approved configuration standards and control policies. Deviations, misconfigurations, or configuration drifts are detected by these rules and alerted to the security team.

Production Systems Provisioning

1. Before provisioning any systems, a request must be created and approved in the Github Production Change Management (PRODCM) project.
    - Github access requires authenticated users.
    - Security grants access to the Github PRODCM project following the procedures covered in the Access Establishment and Modification section.
2. The security team must approve the provisioning request before any new system can be provisioned, unless a pre-approved automation process is followed.
3. Once provisioning has been approved, the implementer must configure the new system according to the standard baseline chosen for the system's role.
4. If the system will be used to store sensitive information, the implementer must ensure the volume containing this sensitive data is encrypted.
5. Sensitive data in motion must always be encrypted.
6. A security analysis is conducted once the system has been provisioned. This can be achieved either via automated configuration/vulnerability scans or manual inspection by the security team. Verifications include, but is not limited to:
    - Removal of default users used during provisioning.
    - Network configuration for system.
    - Data volume encryption settings.
    - Intrusion detection and virus scanning software installed.
7. The new system is fully promoted into production upon successful verification against corresponding MiCT standards and change request approvals.

User Endpoint Security Controls and Configuration

1. Employee laptops, including Windows, Mac, and Linux systems, are configured either
    - Manually by IT or the device owner; or
    - Automatically using a configuration management tool or equivalent scripts.
2. The following security controls are applied at the minimum:
    - Disk encryption
    - Unique user accounts and strong passwords
    - Approved NTP servers
    - Approved security agents
    - Locking after 2 mins of inactivity
    - Auto-update of security patches
3. The security configurations on all end-user systems are inspected by Security through either a manual periodic review or an automated compliance auditing tool.

Server Hardening Guidelines and Processes

1. **Linux System Hardening:** Linux systems have their baseline security configuration applied via automation tools. These tools cover:
    - Ensuring that the machine is up-to-date with security patches and is configured to apply patches in accordance with our policies.
    - Stopping and disabling any unnecessary OS services.
    - Apply applicable DISA STIGs to OS and applications.
    - Configuring 15-minute session inactivity timeouts for SSH sessions.
    - Installing and configuring the virus scanner.
    - Installing and configuring the NTP daemon, including ensuring that modifying system time cannot be performed by unprivileged users.
    - Configuring disk volumes for providers that do not have native support for encrypted data volumes, including ensuring that encryption keys are protected from unauthorized access.
    - Configuring authentication to the centralized Directory Services servers.
    - Configuring audit logging as described in the Auditing Policy section.
2. **Windows System Hardening:** Windows systems have their baseline security configuration applied via the combination of Group Policy settings and/or automation scripts. These baseline settings cover:
    - Joining the Windows Domain Controller and applying the Active Directory Group Policy configuration (for AD-managed systems only).
    - Ensuring that the machine is up-to-date with security patches and is configured to apply patches in accordance with our policies.

- Apply applicable DISA STIGs to OS and applications.
- Stopping and disabling any unnecessary OS services.
- Configuring session inactivity timeouts.
- Installing and configuring security protection agents such as anti-virus scanner.
- Configuring transport encryption according to the requirements described in the Mobile Device Security and Disposable Media Management section.
- Configuring the system clock to point to approved NTP servers and ensuring that modifying system time cannot be performed by unprivileged users.
- Configuring audit logging as described in the Auditing Policy section.

3. Any additional configuration changes applied to hardened Windows systems must be clearly documented by the implementer and reviewed by the Security team.

Configuration and Provisioning of Management Systems

1. Provisioning management systems such as configuration management servers, remote access infrastructure, directory services, or monitoring systems follows the same procedure as provisioning a production system.
2. Critical infrastructure roles applied to new systems must be clearly documented by the implementer in the change request.

Configuration and Management of Network Controls

All network devices and controls on a sensitive network are configured such that:

- Vendor provided default configurations are modified securely, including
  - default encryption keys,
  - default SNMP community strings, if applicable,
  - default passwords/passphrases, and
  - other security-related vendor defaults, if applicable.
- Encryption keys and passwords are changed anytime anyone with knowledge of the keys or passwords leaves the company or changes positions.
- Traffic filtering (e.g. firewall rules) and inspection (e.g. Network IDS/IPS or AWS VPC flow logs) are enabled.
- An up-to-date network diagram is maintained.

In AWS, network controls are implemented using Virtual Private Clouds (VPCs) and Security Groups. The configurations are managed as code and stored in approved repos. All changes to the configuration follow the defined code review, change management and production deployment approval process.

Provisioning AWS Accounts

AWS Account Structure / Organization MiCT maintains a single Organization in AWS, maintained in a top-level AWS account (master). Sub-accounts are connected that each hosts separate workloads and resources in its own sandboxed environment. The master account itself handles aggregated billing for all connected sub-accounts but does not host any workload, service or resource, with the exception of DNS records for MiCT root domain, using AWS Route53 service. DNS records for subdomains are maintained in the corresponding sub-accounts.

Access to each account is funneled through our designated SSO provider, which establishes a trust relationship to a set of predefined roles in the master account. Once authenticated, a user then leverages AWS IAM Assume Role capability to switch to a sub-account to access services and resources.

The account and network structure looks like the following:

```
SSO/IdP   MiCT-master
                billing and root DNS records only

              MiCT-dev
                   VPC
                       Subnets
                            Security-Groups
                                 EC2 instances
                                      Docker containers

              MiCT-test
                   VPC
                        Subnets
                             Security-Groups
                                  EC2 instances
                                       Docker containers

              MiCT-infra
                   VPC
                        Subnets
                             Security-Groups
                                  EC2 instances
                                       Docker containers

              MiCT-prod-us1
                   VPC
                        Subnets
                             Security-Groups
                                  EC2 instances
```

`Docker containers`

`MiCT-prod-us2`

...

Infrastructure-as-Code MiCT AWS environments and infrastructure are managed as code. Provisioning is accomplished using a set of automation scripts and Terraform code. Each new environment is created as a sub-account connected to `MiCT-master`. The creation and provisioning of a new account follows the instructions documented in the Bootstrap a new AWS environment page of the development wiki.

Automated change management for deploys to AWS

The MiCT Continuous Delivery Pipeline automates creation and validation of change requests. This is done in a 3-step process:

1. **Create/Validate Change Request Ticket**

   https://localhost is used for continuous delivery (build and deploy), and we employ https://localhost-Github automation such that:

   - Whenever deployment to a controlled environment (e.g. production accounts and infrastructure account) is requested, the https://localhost job will check for an approved PRODCM ticket, or create a new ticket if not found.
   - The automation code will attempt to automatically populate the required data for the PRODCM ticket (see list above).
   - If the data cannot be automatically populated, https://localhost may pause the job and prompt the user for manual input.
   - Job will be paused until the request is approved or canceled (rejected). Before continuing to deployment, https://localhost will validate the change request's build job identifier, build number and source code branch.

2. **Detect Change Details and Obtain Approval**

   A separate `change-management-bot` is implemented to provide additional details to assist/automate the ticket approval process:

   - Whenever a PRODCM ticket is created, the bot is triggered via a Github webhook.
   - The bot is configured to examine the following:
     - Look for all the code changes since the last approved PRODCM ticket.
     - Check that all code commits have been approved by a reviewer other than the author, except for a version bump.

- Ensure that security scanning has been completed for this build and no blocking issue is found.

!!! attention

```
The following practices will fail this validation and result in
manual processing, therefore should be avoided:

- squashing commits on PR merges
- commits after PR approval without re-approval
```

- Details of the analysis are posted to the PRODCM Github ticket.
- When all the required checks pass validation, the bot recommends approval. The cm-bot may be configured to automatically approve the ticket if all of the required conditions above (and future ones) are met. Additionally, a manual review / approval is always required in the following conditions:
  - This is the first prod deploy with no prior approval history
  - A related CM ticket / deploy of the same project is pending
- If human approvals are needed, the required approvers will review the details and approve/decline accordingly.
- Random inspections of automatically approved tickets are performed by the security team monthly to ensure the automation functions properly.

!!! important

```
1. Note that the above flow does not catch weaknesses in design, and
therefore does not replace the need for threat modeling and security
review in the design phase.
2. Additional requirements may be added later as the process continues
to mature.
```

3. **Detect Risky Changes, Deploy and Close**

   https://localhost job proceeds only with an approved and validated PRODCM ticket.

   - During production deploys, a `terraform plan` is always performed first to detect risky changes.
   - Examples of security-related or risky changes include:
     - Change to "policy" attribute of resource (aws_s3_bucket.policy, aws_kms_key.policy)
     - Change to IAM policy, role, user or group
     - Attach/detach policy
     - Change/delete to security group

  - Anything is deleted (in prod, deletes should be unusual so they should be manually reviewed)
- If risky changes are detected, the deploy is paused and the PRODCM ticket is updated to require manual review before continuing.
- Once a deploy is completed, the PRODCM ticket is automatically resolved and closed.

Patch Management Procedures

Local Systems

MiCT uses automated tooling to ensure systems are up-to-date with the latest security patches.

- On local Linux and Windows systems, the unattended-upgrades tool is used to apply security patches in phases.
  - High Risk security patches are automatically applied as they are released
  - Monthly system patching for regular applications are applied as needed.
  - Snapshotting of a system will take place before an update is applied.
  - Once the update is deemed stable the snapshot will be removed.
  - In case of failure of the update the snapshot will be rolled back.
  - If the staging systems function properly after the two-week testing period, the security team will promote that snapshot into the mirror used by all production systems. These patches will be applied to all production systems during the next nightly patch run.
  - The patching process may be expedited by the Security team
  - On Windows systems, the baseline Group Policy setting configures Windows Update to implement the patching policy.

Cloud Resources

MiCT follows a "cattle-vs-pets" methodology to keep the resources in the cloud environments immutable and up-to-date with security patches.

- AWS Elastic Container Service (ECS) is used to dynamically manage container resources based on demand.
- Engineering team builds security-approved AMI from the latest AWS optimized Amazon Machine Image (AMI) to include required security agent.
- The security agents installed on the security-approved AMIs continuously scan for and report new vulnerabilities.
- The custom AMIs are automatically rebuilt from the latest AWS AMIs weekly to include the latest security patches.

User Endpoints

MiCT requires auto-update for security patches to be enabled for all user endpoints, including laptops and workstations.

- The auto-update configuration and update status on all end-user systems are inspected by Security through either manual periodic audits or automated compliance auditing agents installed on the endpoints.

Production Deploy / Code Promotion Processes

In order to promote changes into Production, a valid and approved Change Request (CR) is required. It can be created in the Change Management System/Portal which implements the MiCT Change Management workflow, using the Production Change Management (PRODCM) Github project to manage changes and approvals.

- At least two approvals are required for each PRODCM ticket. By default, the approvers are
  - Security Lead and
  - Engineering Lead.
- Additional approver(s) may be added depending on the impacted component(s). For example,
  - the IT Manager is added as an approver for IT/network changes; and
  - the DevOps Lead is added as an approver for changes to `aws-MiCT-infra` account in AWS.
- Each PRODCM ticket requires the following information at a minimum:
  - Summary of the change
  - Component(s) impacted
  - Justification
  - Rollback plan
- Additional details are required for a code deploy, including:
  - Build job name
  - Build ID and/or number
  - Deploy action (e.g. plan, apply)
  - Deploy branch (e.g. master)
  - Target environment (e.g. `aws-MiCT-infra`, `aws-MiCT-prod-us`, `datacenter-hq`)
  - Links to pull requests and/or Github issues
  - Security scan status and results

Emergency Change

In the event of an emergency, the person or team on call is notified. This may include a combination or Development, IT, and Security.

If an emergency change must be made, such as patching of a zero-day security vulnerability or recovering from a system downtime, and that the standard change management process cannot be followed due to time constraint or personnel availability or other unforeseen issues, the change can be made by:

- **Notification:** The Engineering Lead, Security Lead, and/or IT Lead must be notified by email, Slack, or phone call prior to the change . Depending on the nature of the emergency, the leads may choose to inform members of the executive team.
- **Access and Execution:** Manually access of the production system or manual deploy of software, using one of the following access mechanisms as defined in Access Control policy and procedures:
  1. Support/Troubleshooting access
  2. Root account or root user access
  3. Local system access (for on-premise environment)
- **Post-emergency Documentation:** A PRODCM ticket should be created within 24 hours following the emergency change. The ticket should contains all details related to the change, including:
  - Reason for emergency change
  - Method of emergency access used
  - Steps and details of the change that was made
  - Sign-off/approvals must be obtained per the type of change as defined by the standard CM process
- **Prevention and Improvement:** The change must be fully reviewed by Security and Engineering together with the person/team responsible for the change. Any process improvement and/or preventative measures should be documented and an implementation plan should be developed.