

# AWS Control Tower Security

## MANAGEMENT

AWS Control Tower orchestrates multi-account governance by deploying a landing zone with guardrails (preventive, detective, proactive), Account Factory, and centralized logging. Compromising the management account or disabling controls grants unrestricted access across the entire organization.

**CRITICAL**

Risk Level

**Prev/  
Det/Pro**

Control Types

**Landing  
Zone**

Governance

**Mgmt  
Account**

Top Target

## Service Overview

### Landing Zone & Controls

Pre-configured secure multi-account environment. Preventive controls (SCPs), detective controls (Config rules), and proactive controls (CloudFormation Hooks). Mandatory controls cannot be disabled.

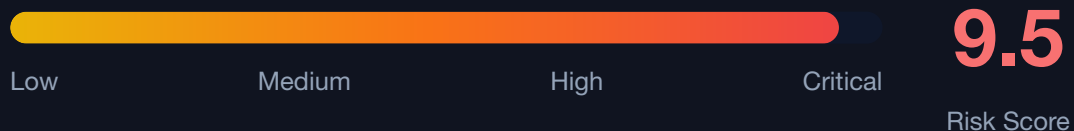
**Attack note:** The management account is exempt from all preventive controls (SCPs). Any identity in this account can bypass every guardrail.

### Account Factory & Key Roles

Standardized account provisioning via Service Catalog. Creates AWSControlTowerExecution role in every member account with AdministratorAccess, trusting the management account root.

**Attack note:** AWSControlTowerExecution in every member account has AdministratorAccess and trusts the management account root with no conditions by default.

## Security Risk Assessment



Compromise of the management account means full admin access to every enrolled member account via AWSControlTowerExecution. Disabling controls removes guardrails organization-wide. Drift introduced by an attacker can disable security detection without obvious alerts.

## ✂ Attack Vectors

### Management Account Compromise

- Assume AWSControlTowerExecution in any member account
- Exploit weakly-scoped IAM policies for role assumption
- Compromise AWSControlTowerAdmin role
- Target Account Factory / AFT pipeline credentials
- Social-engineer org administrators

### Control & Landing Zone Manipulation

- Disable preventive controls (SCPs) on OUs
- Disable detective controls to stop compliance monitoring
- Introduce landing zone drift by modifying SCPs
- Delete/modify AWSControlTowerCloudTrailRole
- Abuse delegated administrator privileges

## ⚠ Misconfigurations

### Dangerous Defaults

- AWSControlTowerExecution trusts mgmt account root with no conditions
- No Permissions Boundaries on management account identities
- Only mandatory controls enabled
- Account Factory provisions without additional SCPs
- Audit account Lambda roles have cross-account admin

### Governance Gaps

- No alerting for control disable/enable operations
- Landing zone drift not monitored
- AFT pipeline credentials stored without rotation
- No SCP protecting Control Tower roles in member accounts
- Management account used for daily workloads

## Enumeration

List Landing Zones

```
aws controltower list-landing-zones
```

List Enabled Controls on an OU

```
aws controltower list-enabled-controls \
  --target-identifier
arn:aws:organizations::123456789012:ou/o-abc/ou-abc-xyz
```

List Drifted Controls

```
aws controltower list-enabled-controls \
  --target-identifier
OU_ARN \
  --filter
'{"driftStatuses":
["DRIFTED"]}'
```

List All Baselines

```
aws controltower list-baselines
```

List Organization Accounts

```
aws organizations list-accounts \
  --query 'Accounts[*].
[Id,Name,Status]' --output
table
```

## Privilege Escalation

### From Member to Full Org Access

- Assume AWSControlTowerExecution in any member account
- Audit account role chain to AdministratorExecutionRole
- Register compromised account as delegated admin
- Exploit management account identity with sts:AssumeRole \*

### From Management Account

- Disable preventive controls to remove SCP restrictions
- Disable detective controls to blind compliance
- Modify landing zone manifest to weaken security
- Create new accounts with attacker-controlled settings

#### Key insight:

AWSControlTowerExecution has AdministratorAccess in every member account. A single sts:AssumeRole with Resource: \* in the management account is enough to pivot to all accounts.

## Persistence

### Maintaining Access

- AWSControlTowerExecution role persists across account lifecycle
- Add conditions bypass in member account trust policies
- Create accounts via Account Factory with backdoor settings
- Register as delegated administrator for sensitive services
- Store AFT pipeline credentials for future access

### Evading Detection

- Introduce drift that disables detective controls
- Modify CloudTrail role to disrupt centralized logging
- Move accounts between OUs to bypass controls
- Disable auto-enrollment for new accounts
- Use management account (exempt from SCPs)

## Detection

### Critical CloudTrail Events

- DisableControl / EnableControl - control changes
- UpdateLandingZone / ResetLandingZone - LZ modifications
- AssumeRole on AWSControlTowerExecution
- CreateManagedAccount - new account creation
- DisableBaseline - baseline removal

### Drift Indicators

- SCP modifications outside Control Tower
- Accounts moved between OUs manually
- Control Tower roles modified in member accounts
- Landing zone status showing DRIFTED
- CloudTrail configuration changes



## Exploitation Commands

Assume AWSControlTowerExecution

```
aws sts assume-role \  
  --role-arn arn:aws:iam:::role/  
AWSControlTowerExecution \  
  --role-session-name ct-pivot
```

Disable a Control on an OU

```
aws controltower disable-control \  
  --control-identifier arn:aws:controltower:us-east-1::control/  
CONTROL_ID \  
  --target-identifier OU_ARN
```

List SCPs on an OU

```
aws organizations list-policies-for-target \  
  --target-id ou-xxxx-xxxxxxxx \  
  --filter SERVICE_CONTROL_POLICY
```

Get Landing Zone Drift Status

```
aws controltower get-landing-zone \  
  --landing-zone-identifier LZ_ARN
```

## Policy Examples

### **X Dangerous - Wildcard AssumeRole in Mgmt Account**

```
{
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource": "*"
}
```

Allows assumption of AWSControlTowerExecution in every member account -- full org compromise

### **✓ Secure - Permissions Boundary Blocking CT Role**

```
{
  "Effect": "Deny",
  "Action": "sts:AssumeRole",
  "Resource": "arn:aws:iam::*:role/AWSControlTowerExecution"
}
```

Applied as Permissions Boundary to block unauthorized pivoting to member accounts

### **✓ Secure - SCP Protecting CT Roles**

```
{
  "Effect": "Deny",
  "Action": ["iam:DeleteRole", "iam:PutRolePolicy",
    "iam:UpdateAssumeRolePolicy"],
  "Resource": ["arn:aws:iam::*:role/
    AWSControlTowerExecution", "arn:aws:iam::*:role/aws-
    controltower-*"]
}
```

Prevents member account identities from modifying or deleting Control Tower roles

## Defense Recommendations



### Isolate the Management Account

Run zero production workloads. Restrict to Control Tower administration, Organizations management, and billing only.



### Apply Permissions Boundaries

Deny sts:AssumeRole against AWSControlTowerExecution on all management account identities.



### Add Conditions to CT Execution Trust

Modify trust policy on AWSControlTowerExecution in member accounts to require principal tags or IP ranges.



### Enable All Strongly Recommended Controls

Do not rely only on mandatory controls. Enable all strongly recommended preventive and detective controls.



### Monitor Control Tower API Calls

Alert on DisableControl, EnableControl, UpdateLandingZone, and AWSControlTowerExecution role assumptions.



### Protect CT Roles via SCP

Deploy SCP denying modification/deletion of AWSControlTowerExecution and aws-controltower-\* roles by member accounts.

Always obtain proper authorization before testing