Vantalect

AWS SECURITY, COMPLIANCE & GOVERNANCE · SAMPLE DELIVERABLE

Incident Runbooks and Handover

Week 6 of 6: Hand over · prepared for StrataForge Robotics, Inc.

Prepared by Charles Sieg · charles.sieg@vantalect.com · vantalect.com

August 21, 2026 · AWS Security, Compliance & Governance Audit: The Hardening (six weeks)

SAMPLE DELIVERABLE

This is a sample deliverable. StrataForge Robotics, Inc. is a fictional company: every account, resource, finding, figure, and person in this report is simulated to show the form and depth of a Vantalect engagement. AWS prices are list prices from the AWS Price List API on October 13, 2026.

Contents

Summary 2
The handover workshop 5
The runbooks 6
RB-01: OTA signing key used by an unexpected principal 7
RB-02: Robot fleet authorization failures or message-volume anomaly 8
RB-03: Root or break-glass sign-in 9
RB-04: Credentials used from outside AWS 10
RB-05: Access key exposed 11
RB-06: Logging or detection turned off 12
RB-07: Something became public 13
RB-08: Kubernetes cluster compromise 14
RB-09: Compromised instances in a group 16
RB-10: Critical control failure on a production account 17
How the runbooks are tested 17
The remaining roadmap 18
By owner 22
What matters most, and in what order 22
The guardrails StrataForge now owns 23
Keeping the score where it is 25
The monthly review 25
The exception process 27
The new-account baseline 27
The quarterly re-scan 28

Summary

This is what StrataForge now owns. In six weeks the Vantalect scan score (71 controls) went from 32.4% to 91.1% (86.3% without the suppressions the work earned), every critical finding is closed or accepted in writing, and the second invoice is issued. This report is the handover: **10 incident**

runbooks in full, the agenda of the 2-hour handover workshop, the **18 findings still open or partly remediated, each with an owner and a date**, the guardrails your platform team now owns, and the routine that keeps the score where it is.

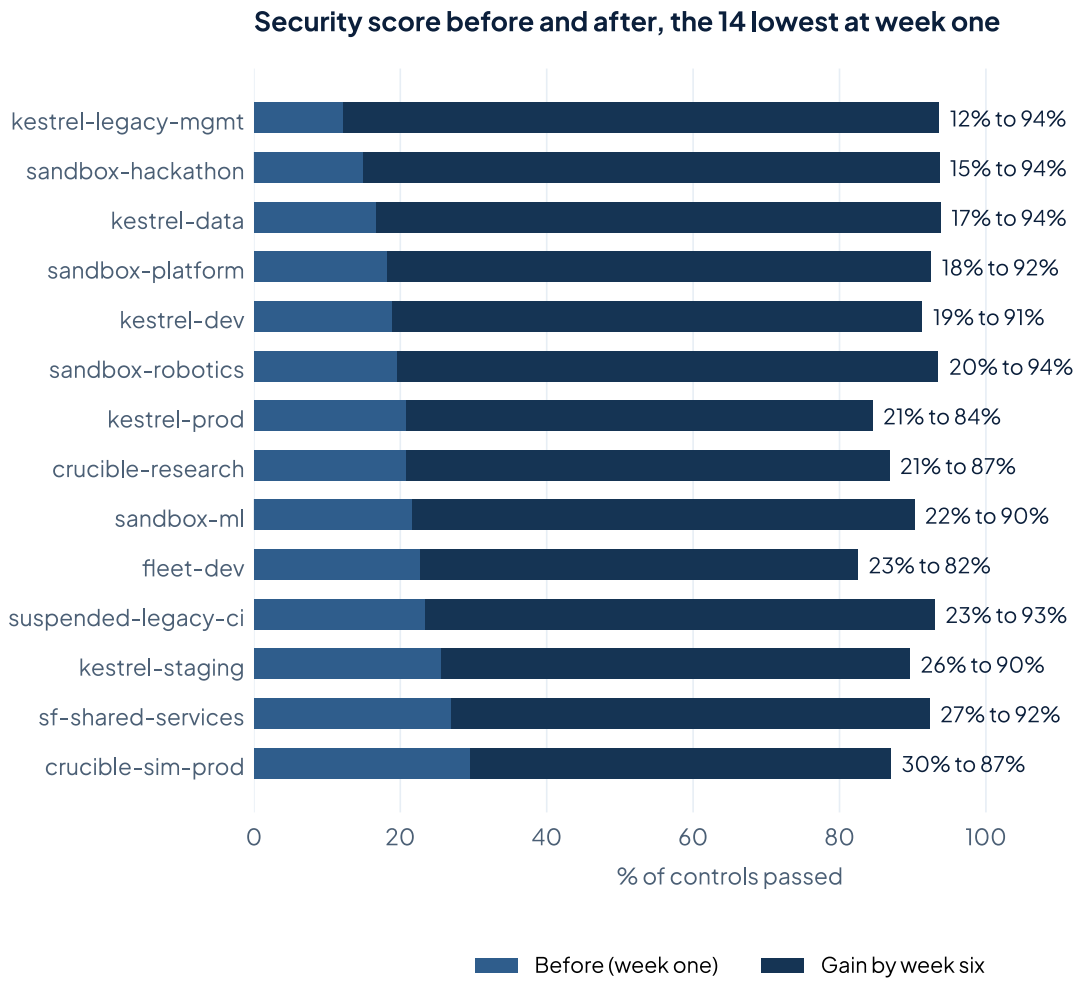
The one thing to know: **the score does not hold itself.** A new account, a new workload, or a new engineer adds failing controls faster than anyone expects, and the guardrails only stop the failures they were written for. What holds it is a monthly review that takes Elena Vasquez's platform team and Nadia Petrova's security operations team about an hour, a quarterly test of the runbooks and the break-glass path, and exceptions that expire. All three are in Keeping the score where it is.

The state of the register at the end of week 6.

Handover at a glance	Result
Vantalect scan score, both standards, week one to week six	32.4% to 91.1% (86.3% without the new suppressions)
Findings in the register	54
Closed	34
Accepted in writing	2
Partially remediated	8
Open	10
Incident runbooks	10
Remaining remediation effort	717 hours
Next date a finding is due	September 4, 2026 (SF-SEC-048)
Last date a finding is due	January 29, 2027 (SF-SEC-031)
Risk acceptances that expire	December 15, 2026 (SF-SEC-002); August 14, 2027 (SF-SEC-049)

Vantalect's access ends with the engagement. On the day of the workshop, your engineers confirm that the read-only audit role is gone from every account, including the management account, where it was a separate stack. This check lists any use of the role after the removal; run it in the management account and in a sample of member accounts, and it should return nothing. Replace `<removal-time>` with the time the last delete finished.

```bash
aws cloudformation describe-stacks --stack-name vantalect-audit-readonly   # in the management
account: must fail with "does not exist"
aws cloudtrail lookup-events \
  --lookup-attributes AttributeKey=EventName,AttributeValue=AssumeRole \
  --start-time <removal-time> --region us-east-1 \
  --query 'Events[?contains(CloudTrailEvent, `vantalect-audit-readonly`)].[EventTime,Username]'
```



The 14 lowest-scoring accounts at week one, with the score they started from and the gain by week six.

The handover workshop

Deliverable 15 is the workshop and its recording. It runs two hours on Zoom, is recorded, and is attended by the people who will do the work: Marcus Oyelaran, Elena Vasquez, and the owners of the findings and runbooks below. The recording is part of the work product, so a new engineer in a year can watch the part that is theirs.

The agenda of the 2-hour handover workshop.

Time	Topic	Led by	What you leave with
0:00 to 0:10	The result, and what is now yours	Marcus Oyelaran and Charles Sieg	The re-scan, the gate, and the one-page list of what Vantalect no longer touches
0:10 to 0:45	The 8 high-urgency runbooks (RB-01, RB-02, RB-03, RB-04, RB-05, RB-06, RB-08, RB-09), with a live test: a sample finding becomes a page	Nadia Petrova	Runbooks each owner has read aloud, and the page time of the live test
0:45 to 0:55	The 2 runbooks that wait for business hours or page only for production (RB-07, RB-10)	Nadia Petrova and Elena Vasquez	Which of them page at night and which wait for business hours
0:55 to 1:15	Guardrails: changing a service control policy safely, the exception process, and the new-account baseline	Elena Vasquez	A tested rollback for each policy, and the exception register
1:15 to 1:30	Identity: joiners, movers, and leavers; the quarterly break-glass test; the factory gateway keys	Grace Liu	The monthly access review, ready to run
1:30 to 1:50	The remaining roadmap: owners, dates, and blockers	Sam Okafor and Elena Vasquez	Every date confirmed or moved on the record
1:50 to 2:00	The monthly review: who chairs, when it meets, and the first agenda	Marcus Oyelaran	A meeting on the calendar

Three rules for the session. Nothing is shown that the owner has not seen before, so nobody is surprised in front of a colleague. The live test uses a sample finding, never a real one. And every date in the roadmap is confirmed by its owner out loud, because a date nobody has said aloud is a hope.

The runbooks

Deliverable 13. There is one runbook for each alert most likely to page StrataForge's on-call. Each has a trigger, a PagerDuty service, an owner, and the first 15 minutes of response, and each lives in `platform-landing-zone/runbooks/` next to the alarm that triggers it, so changing the alarm and changing the procedure happen in one pull request. The first 15 minutes are about stopping harm and preserving evidence; the investigation comes after. Two rules apply to every one: **contain first and ask questions second**, and **never delete evidence**.

The 10 runbooks and who they page.

ID	Runbook	Pages	Urgency
RB-01	OTA signing key used by an unexpected principal	Fleet OTA	high urgency
RB-02	Robot fleet authorization failures or message-volume anomaly	Fleet Cloud	high urgency
RB-03	Root or break-glass sign-in	Security	high urgency
RB-04	Credentials used from outside AWS	Security	high urgency
RB-05	Access key exposed	Identity	high urgency
RB-06	Logging or detection turned off	Security	high urgency
RB-07	Something became public	Security	low urgency; high for production accounts
RB-08	Kubernetes cluster compromise	Fleet Cloud	high urgency
RB-09	Compromised instances in a group	Security	high urgency
RB-10	Critical control failure on a production account	Platform	low urgency, business hours

RB-01: OTA signing key used by an unexpected principal

RB-01 at a glance.

Item	Detail
Trigger	EventBridge: kms:Sign on alias/ota-firmware-signing-prod by any principal other than ota-release-signer
Pages	Fleet OTA (high urgency)
Owner	Sam Okafor, Director, Fleet Cloud Engineering
Linked finding types	`ota-signing-user`, `ota-test-key-policy`
Linked findings	SF-SEC-001; SF-SEC-021
Lives in	`platform-landing-zone/runbooks/ota-signing-key.md`

The first 15 minutes.

1. Pause OTA distribution: apply the pre-approved bucket policy that denies s3:PutObject under releases/ (one command in the runbook).
2. Cancel any IoT job created in the last hour that targets firmware not in the release register.
3. Identify the principal and its session from the CloudTrail event; revoke its sessions and disable its credentials.
4. Page the fleet engineering lead and the CISO; start the incident record in Pavilion.
5. Compare every object written to releases/ in the last 24 hours with the release register hashes.

The pause is one pre-approved policy. Save the current bucket policy first, apply the pause, and restore the saved policy when the release engineer and the CISO agree it is safe. The pause denies every writer to `releases/`, the publisher role included, which is the point.

```bash
aws s3api get-bucket-policy --bucket <ota-distribution-bucket> --query Policy --output text >
ota-policy-before-pause.json
aws s3api put-bucket-policy --bucket <ota-distribution-bucket> --policy file://ota-pause-
releases.json
```

```JSON
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PauseReleases",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::<ota-distribution-bucket>/releases/*"
    }
  ]
}
```

`ota-pause-releases.json` must carry the bucket's existing statements as well, because `put-bucket-policy` replaces the whole policy; the saved copy above is where you get them.

Who decides what. Sam Okafor owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-02: Robot fleet authorization failures or message-volume anomaly

RB-02 at a glance.

Item	Detail
Trigger	AWS IoT Device Defender Detect violation on a robot thing group (authorization failures or messages sent)
Pages	Fleet Cloud (high urgency)
Owner	Sam Okafor, Director, Fleet Cloud Engineering
Linked finding types	`iot-policy-scope`, `iot-shared-certs`
Linked findings	SF-SEC-002; SF-SEC-030
Lives in	`platform-landing-zone/runbooks/robot-fleet-anomaly.md`

The first 15 minutes.

1. Open the violation and list the devices and their customer sites.
2. If one certificate is involved, move its robot to the quarantine thing group, whose policy is an explicit Deny on every IoT action. A policy that only allows less removes nothing, because IoT policies add up and an Allow elsewhere on the certificate still applies.

3. If more than 20 devices at one site are involved, call the site contact before deactivating anything: a shared certificate deactivated mid-shift stops a production line. Once the site agrees, deactivate it with update-certificate --new-status INACTIVE.
4. Check whether the certificate belongs to the legacy firmware group covered by the accepted risk.
5. Record the decision and the customer notification in the incident record.

```bash
aws iot list-thing-groups-for-thing --thing-name <thing-name>
aws iot add-thing-to-thing-group --thing-group-name <quarantine-thing-group> --thing-name
<thing-name>
# Only after the site agrees: stop the certificate from connecting at all.
aws iot update-certificate --certificate-id <certificate-id> --new-status INACTIVE
```

The quarantine group's policy is an explicit Deny on every IoT action, because IoT policies add up and a group that allows less would take nothing away. Deactivate the certificate only when you know the robot is not on a production line. A shared certificate deactivated mid-shift stops the line, which is why the third step in the runbook is a phone call and not a command.

Who decides what. Sam Okafor owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-03: Root or break-glass sign-in

RB-03 at a glance.

Item	Detail
Trigger	EventBridge: ConsoleLogin by the management account root user or breakglass-01/02, or any sts:AssumeRoot call
Pages	Security (high urgency)
Owner	Nadia Petrova, Manager, Security Operations
Linked finding types	`break-glass`, `root-credentials`
Linked findings	SF-SEC-040; SF-SEC-010, SF-SEC-029
Lives in	`platform-landing-zone/runbooks/break-glass.md`

The first 15 minutes.

1. Call the person named in the break-glass log for today; no entry means treat it as hostile.

2. If hostile: from a second administrator, remove the break-glass users' console access and rotate their passwords.
3. If the trigger is the management account's root user, or an sts:AssumeRoot call nobody can account for: find the calling principal in the event, revoke its sessions with an inline deny on aws:TokenIssueTime, and disable its credentials. An AssumeRoot session lasts at most 15 minutes, so contain the principal that can start the next one. For the management root, reset its password with the hardware-MFA holder and confirm it has no access keys.
4. Review every API call in the session from CloudTrail Lake.
5. Notify the CISO within 15 minutes, whatever the outcome.

```bash
aws cloudtrail start-query --query-statement \
  "SELECT eventTime, eventName, eventSource, sourceIPAddress, userAgent FROM <event-data-store-id> WHERE userIdentity.arn LIKE '%<root-or-breakglass-arn-fragment>%' AND eventTime > '<session-start-time>' ORDER BY eventTime"
aws cloudtrail get-query-results --query-id <query-id>
```

Who decides what. Nadia Petrova owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-04: Credentials used from outside AWS

RB-04 at a glance.

Item	Detail
Trigger	GuardDuty: UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS
Pages	Security (high urgency)
Owner	Nadia Petrova, Manager, Security Operations
Linked finding types	`imdsv2`, `workload-admin-roles`
Linked findings	SF-SEC-014; SF-SEC-017, SF-SEC-046
Lives in	`platform-landing-zone/runbooks/credential-exfiltration.md`

The first 15 minutes.

1. Identify the instance role and the instance from the finding.

2. Revoke the role's active sessions (aws:TokenIssueTime condition) and isolate the instance with the forensics security group.
3. Snapshot the instance's volumes into sf-forensics before anything is terminated.
4. Check the instance for IMDSv1 and for an application SSRF path.

Revoking sessions is an inline deny on everything issued before now. It does not delete the role, so the instance keeps working under new credentials after the compromise is contained.

```bash
aws iam put-role-policy --role-name <instance-role-name> --policy-name AWSRevokeOlderSessions \
  --policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Deny","Action":
["*"],"Resource":["*"],"Condition":{"DateLessThan":{"aws:TokenIssueTime":"<current-utc-
time>"}}}]}'
aws ec2 modify-instance-attribute --instance-id <instance-id> --groups <forensics-security-
group-id>
```

Who decides what. Nadia Petrova owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-05: Access key exposed

RB-05 at a glance.

Item	Detail
Trigger	AWS Health event AWS_RISK_CREDENTIALS_EXPOSED, or a secret-scanning alert from GitHub
Pages	Identity (high urgency)
Owner	Grace Liu, Manager, Identity and Access
Linked finding types	`access-key-retirement`, `lambda-env-secrets`
Linked findings	SF-SEC-012; SF-SEC-015
Lives in	`platform-landing-zone/runbooks/exposed-key.md`

The first 15 minutes.

1. Deactivate the key; AWS may already have attached its quarantine policy.
2. Find the owner from the key's IAM user tags and the retirement register.
3. Review the key's CloudTrail activity since the exposure time.

4. Replace the key with a role. For the remaining factory gateways, PlatformIdentityAdmin issues the new key (the sf-protect-account-defaults policy denies iam:CreateAccessKey to everyone else), and it is recorded in the retirement register.

```bash
aws iam update-access-key --user-name <user-name> --access-key-id <access-key-id> --status
Inactive
aws iam list-user-tags --user-name <user-name>
aws cloudtrail lookup-events --lookup-attributes
AttributeKey=AccessKeyId,AttributeValue=<access-key-id> \
  --start-time <exposure-time> --max-results 50
```

Deactivate before you delete. A deactivated key can be turned back on if the owner proves a pipeline depended on it; a deleted one cannot.

Who decides what. Grace Liu owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-06: Logging or detection turned off

RB-06 at a glance.

Item	Detail
Trigger	GuardDuty: Stealth:IAMUser/CloudTrailLoggingDisabled, or an SCP denial on GuardDuty, Security Hub CSPM, AWS Config, or CloudTrail changes
Pages	Security (high urgency)
Owner	Nadia Petrova, Manager, Security Operations
Linked finding types	`cloudtrail-log-archive`, `guardduty-org`, `scp-guardrails`
Linked findings	SF-SEC-025; SF-SEC-023; SF-SEC-026
Lives in	`platform-landing-zone/runbooks/detection-disabled.md`

The first 15 minutes.

1. Confirm sf-org-trail-v2 is logging (get-trail-status) and that its bucket's Object Lock configuration is unchanged.
2. Identify the principal; an SCP denial by an unexpected principal is an attempted compromise.
3. Re-enable anything the security automation role finds disabled.

4. Escalate to the CISO.

```bash
aws cloudtrail get-trail-status --name sf-org-trail-v2 --region us-east-1
aws s3api get-object-lock-configuration --bucket sf-log-archive-org-trail-locked-us-east-1
aws guardduty list-detectors --region <region>
aws securityhub describe-hub --region <region>
```

Who decides what. Nadia Petrova owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-07: Something became public

RB-07 at a glance.

Item	Detail
Trigger	IAM Access Analyzer finding with public access, or Security Hub CSPM S3.2, RDS.1, or EC2.19 failing
Pages	Security (low urgency; high for production accounts)
Owner	Nadia Petrova, Manager, Security Operations
Linked finding types	`s3-public-bucket`, `ebs-snapshot-public`, `rds-snapshot-public`, `admin-ports-internet`, `notebook-exposed`
Linked findings	SF-SEC-006; SF-SEC-005; SF-SEC-007; SF-SEC-009, SF-SEC-027, SF-SEC-028, SF-SEC-041; SF-SEC-004
Lives in	`platform-landing-zone/runbooks/public-exposure.md`

The first 15 minutes.

1. Remove the public access first (bucket policy, snapshot attribute, or security group rule); ask questions second.
2. Check S3 server access logs and CloudTrail data events for downloads while it was public.
3. Find the change that made it public and its pull request or console session.
4. If customer data was exposed, start the privacy incident procedure.

```bash
# S3 bucket: block everything first, then look.
aws s3api put-public-access-block --bucket <bucket-name> \
  --public-access-block-configuration
BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true
# EBS snapshot: remove the public group.
aws ec2 modify-snapshot-attribute --snapshot-id <snapshot-id> --attribute
createVolumePermission --operation-type remove --group-names all
# RDS snapshot: remove the public restore value.
aws rds modify-db-snapshot-attribute --db-snapshot-identifier <snapshot-id> --attribute-name
restore --values-to-remove all
# Security group: remove the open rule.
aws ec2 revoke-security-group-ingress --group-id <security-group-id> --protocol tcp --port
<port> --cidr 0.0.0.0/0
```

Who decides what. Nadia Petrova owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-08: Kubernetes cluster compromise

RB-08 at a glance.

Item	Detail
Trigger	GuardDuty: AttackSequence:EKS/CompromisedCluster or a Runtime Monitoring finding on an EKS node
Pages	Fleet Cloud (high urgency)
Owner	Sam Okafor, Director, Fleet Cloud Engineering
Linked finding types	`eks-endpoint-access`, `eks-operations`
Linked findings	SF-SEC-016; SF-SEC-054
Lives in	`platform-landing-zone/runbooks/eks-compromise.md`

The first 15 minutes.

1. Identify the namespace, pod, and service account from the finding.
2. Cordon the node and apply the deny-all network policy to the namespace.
3. Cut the workload's access: delete its Pod Identity association, and for an IRSA role edit the trust policy so the pod's service account can no longer assume it and add an inline deny on sessions issued before now. Deleting an access entry stops only people and nodes, not an IRSA pod.

4. Capture the node's volumes into sf-forensics.

```bash
kubectl cordon <node-name>
kubectl apply -f deny-all.yaml
aws eks delete-access-entry --cluster-name <cluster-name> --principal-arn <workload-role-arn>
# people and nodes only
aws eks delete-pod-identity-association --cluster-name <cluster-name> --association-id
<association-id>   # Pod Identity workloads
# IRSA workloads: edit the role's trust policy (remove the pod's service account), then deny
older sessions.
aws iam put-role-policy --role-name <workload-role-name> --policy-name AWSRevokeOlderSessions \
  --policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Deny","Action":
["*"],"Resource":["*"],"Condition":{"DateLessThan":{"aws:TokenIssueTime":"<current-utc-
time>"}}}]}'
```

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: deny-all
  namespace: <namespace>
spec:
  podSelector: {}
  policyTypes:
    - Ingress
    - Egress
```

The deny-all policy only works where the cluster's network policy engine is enabled. Confirm it on each production cluster before you need it, not during an incident.

Who decides what. Sam Okafor owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-09: Compromised instances in a group

RB-09 at a glance.

Item	Detail
Trigger	GuardDuty: AttackSequence:EC2/CompromisedInstanceGroup or CryptoCurrency:EC2/BitcoinTool.B!DNS
Pages	Security (high urgency)
Owner	Nadia Petrova, Manager, Security Operations
Linked finding types	`notebook-exposed`, `ci-controller-exposed`, `admin-ports-internet`
Linked findings	SF-SEC-004; SF-SEC-008; SF-SEC-009, SF-SEC-027, SF-SEC-028, SF-SEC-041
Lives in	`platform-landing-zone/runbooks/compromised-instances.md`

The first 15 minutes.

1. Isolate the instances with the forensics security group; suspend their Auto Scaling processes.
2. Snapshot the volumes into sf-forensics.
3. Revoke the instance role's sessions.
4. Look for the entry point: open ports, IMDSv1, an exposed notebook or CI controller.

```bash
aws ec2 modify-instance-attribute --instance-id <instance-id> --groups <forensics-security-group-id>
aws autoscaling suspend-processes --auto-scaling-group-name <asg-name>
aws ec2 create-snapshot --volume-id <volume-id> --description "incident <incident-id>"
aws ec2 modify-snapshot-attribute --snapshot-id <snapshot-id> --attribute createVolumePermission \
  --operation-type add --user-ids <forensics-account-id>
```

An encrypted snapshot also needs its KMS key shared with the forensics account before that account can copy it.

Who decides what. Nadia Petrova owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

RB-10: Critical control failure on a production account

RB-10 at a glance.

Item	Detail
Trigger	Security Hub CSPM: a CRITICAL or HIGH failed control in a production account (routed from the us-east-1 aggregation Region)
Pages	Platform (low urgency, business hours)
Owner	Elena Vasquez, Vice President, Cloud Platform
Linked finding types	`securityhub-org`, `config-org`
Linked findings	SF-SEC-038; SF-SEC-022
Lives in	`platform-landing-zone/runbooks/control-failure.md`

The first 15 minutes.

1. Open the control's remediation entry in the library.
2. Find the change that caused the failure (AWS Config timeline for the resource).
3. Fix it through the owning repository's pull request; suppress only with a documented reason.
4. Record the fix in the findings register.

```bash
aws configservice get-resource-config-history --resource-type <resource-type> --resource-id
<resource-id> --limit 5
```

Who decides what. Elena Vasquez owns this runbook: keeps it current, runs its quarterly test, and signs off any change to its alarm. The on-call engineer decides containment. The decision to notify a customer or the CISO is in the steps above, not left to judgment in the moment.

How the runbooks are tested

A runbook that has not been run is a guess. Each is tested every quarter, and the result is kept with the runbook.

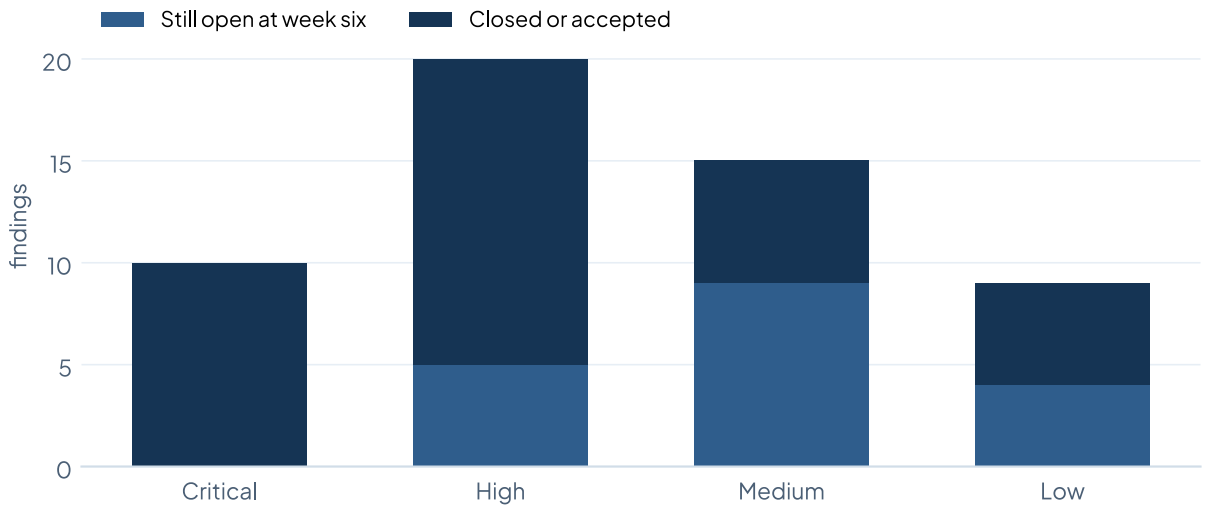
The quarterly runbook tests.

Test	How	What to record
A GuardDuty-sourced page	`aws guardduty create-sample-findings --detector-id <detector-id> --finding-types <finding-type>` in a sandbox account	Time from the call to the page
A Security Hub CSPM page	`aws securityhub batch-import-findings` with a test CRITICAL finding	Time from the call to the page
The break-glass page	Someone signs in as `breakglass-01` with the sealed hardware key	Time to the page; the key returned to the envelope
The trail alarm	`StopLogging` on a scratch trail in the management account, where the alarm lives	Time to the page; trail restarted
Device Defender	A test violation on a scratch thing group	Time to the page, and that the quarantine group works
A tabletop	The on-call reads one runbook aloud for a scenario nobody has seen	Steps that were unclear, fixed in the next pull request

The remaining roadmap

18 findings are not closed: 8 partially remediated and 10 open. Together they are about **717 hours** of work, each with an owner and a date, none critical, and none of it depends on Vantalect. We sequenced them by risk and then by dependency; the dates are the owners', set in the workshop. Open a ticket for each in your own system; the register is the source of truth for scope and evidence.

Findings at week six



The findings still open at week six by severity, against those closed or accepted.

Every open and partially remediated finding, soonest due first, with what remains, its owner, its repository, the estimated effort, and its date.

ID	Severity	Status	Finding	What remains	Owner	Repository	HoursDue
SF-SEC-048	Low	open	1,922 workload roles have not been used in 90 days or more	All of it; not started in the engagement	Grace Liu	`platform-landing-zone`	16Sep 4
SF-SEC-015	High	partially remediated	144 Lambda functions have secret-named environment variables	Move the fleet-dev functions' secrets to Secrets Manager	Sam Okafor	`fleet-infra`	32Sep 30
SF-SEC-016	High	partially remediated	All 16 EKS cluster API endpoints accept connections from 0.0.0.0/0	Make the non-production endpoints private once CI runners reach them through the Transit Gateway	Sam Okafor	`fleet-infra`	38Sep 30
SF-SEC-030	High	partially remediated	23 robot certificates are shared by 1,840 robots at 3 customer sites	Re-provision the robots at the Bremen and Ota sites during their maintenance windows	Sam Okafor	`fleet-infra`	32Sep 30
SF-SEC-014	High	partially remediated	1,631 instances still allow IMDSv1	Switch the instances that still make IMDSv1 calls after their agents and SDKs are upgraded	Elena Vasquez	`platform-landing-zone`	56Oct 30
SF-SEC-033	Medium	open	556 running instances are not managed by Systems Manager; 2,380 run AMIs over a year old	All of it; not started in the engagement	Elena Vasquez	`platform-landing-zone`	20Oct 30
SF-SEC-034	Medium	open	39 internet-facing ALBs have no AWS WAF; 84 ALBs do not redirect HTTP to HTTPS	All of it; not started in the engagement	Sam Okafor	`fleet-infra`	16Oct 30
SF-SEC-043	Medium	partially remediated	750 default VPCs, most in Regions nobody uses, have open default security groups and network ACLs	Replace the open default network ACLs in the working Regions' VPCs	Elena Vasquez	`platform-landing-zone`	16Oct 30
SF-SEC-045	Medium	open	31 databases keep one day of backups; 112 resilience checks fail	All of it; not started in the engagement	Sam Okafor	`fleet-infra`	12Oct 30
SF-SEC-054	Low	partially remediated	EKS control-plane logging is off on 15 of 16 clusters, and every cluster still uses the aws-auth ConfigMap	Move every cluster to access entries and upgrade 1.31 and 1.32 clusters	Sam Okafor	`fleet-infra`	24Nov 13

ID	Severity	Status	Finding	What remains	Owner	Repository	HoursDue
SF-SEC-012	High	partially remediated	1,009 active access keys are older than 90 days; 400 are over three years old	Replace the factory gateway users' keys with IAM Roles Anywhere during factory maintenance windows	Grace Liu	`platform-landing-zone`	63Nov 20
SF-SEC-035	Medium	open	22 database instances store data unencrypted	All of it; not started in the engagement	Sam Okafor	`fleet-infra`	74Nov 20
SF-SEC-036	Medium	open	9 ElastiCache replication groups lack encryption at rest or in transit	All of it; not started in the engagement	Sam Okafor	`fleet-infra`	33Nov 20
SF-SEC-037	Medium	open	5 EFS file systems are not encrypted at rest	All of it; not started in the engagement	Hiro Tanaka	`crucible-infra`	12Nov 20
SF-SEC-032	Medium	partially remediated	EBS encryption by default is off in 762 account-Regions; 1,245 volumes are unencrypted	Migrate the remaining unencrypted volumes through encrypted snapshot copies	Elena Vasquez	`platform-landing-zone`	78Dec 18
SF-SEC-046	Low	open	130 customer managed policies use service-wide wildcards such as s3:*	All of it; not started in the engagement	Elena Vasquez	`fleet-infra`	40Dec 18
SF-SEC-047	Low	open	2,850 of 14,464 resources have no Team tag, so no owner can be found for them	All of it; not started in the engagement	Elena Vasquez	`platform-landing-zone`	12Dec 18
SF-SEC-031	Medium	open	1,247 Lambda functions run on deprecated runtimes	All of it; not started in the engagement	Sam Okafor	`fleet-infra`	143Jan 29

By owner

The remaining work by owner.

Owner	Findings	HoursEarliest due
Grace Liu, Manager, Identity and Access	2	79September 4, 2026
Elena Vasquez, Vice President, Cloud Platform	6	222October 30, 2026
Sam Okafor, Director, Fleet Cloud Engineering	9	404September 30, 2026
Hiro Tanaka, Director, Crucible Platform	1	12November 20, 2026

What matters most, and in what order

1. **`SF-SEC-014`: IMDSv1.** It is the finding most likely to turn a small application bug into a credential theft, and the instances that still allow it are the ones whose agents and SDKs have not been upgraded. Do these as the owners upgrade; do not set `http_tokens` to `required` on an instance until its owner has confirmed nothing calls IMDSv1.
2. **`SF-SEC-012`: the factory gateway keys.** Replace with IAM Roles Anywhere inside each factory's maintenance window. The date is set by the factory calendar, not ours.
3. **`SF-SEC-016`: public EKS endpoints.** Non-production clusters go private once CI runners reach them through the Transit Gateway. Production clusters are already private.
4. **`SF-SEC-030`: shared robot certificates.** Re-provision at the customer sites in their maintenance windows. This is a customer-calendar finding, like the accepted critical.
5. **The Medium findings with encryption in them** (`SF-SEC-035`, `SF-SEC-036`, `SF-SEC-037`, `SF-SEC-032`): each needs a snapshot, an encrypted copy, and a cutover, and each has a customer-visible outage if it is done carelessly. Do them in the maintenance window of the system, not before.
6. **Everything with a Systems Manager or patching shape** (`SF-SEC-033`, `SF-SEC-031`): get the unmanaged instances under Systems Manager first; everything else about patching depends on it.

Two dates are not in the table because they are not findings. December 15, 2026: the acceptance for SF-SEC-002 expires, so the last robot must be on firmware 4.2 or the signers must decide again, in writing. And November 26, 2026: EKS upgrades the control plane of any cluster still on Kubernetes 1.31 automatically, with no rollback, so the 1.31 clusters must be upgraded by their owners before then (`SF-SEC-054`).

The guardrails StrataForge now owns

The platform team owns these. They are in `platform-landing-zone`, changed only by pull request, reviewed by someone who did not write the change, and deployed by StackSets and the Organizations tooling. Vantalect has no access to any of it after the engagement ends.

The organization policies StrataForge now owns. The policy text is in the Audit Report Appendix (SF-SEC-026, SF-SEC-018, SF-SEC-052, SF-SEC-014, and SF-SEC-005), and the week-3 report says what each one protects.

Guardrail	Kind	Where it applies	What it does
`sf-protect-security-services`	Service control policy	Root	Denies: Stopping or deleting CloudTrail trails and event selectors, AWS Config recorders, GuardDuty, Security Hub CSPM, IAM Access Analyzer, and Inspector, except the platform and security roles
`sf-deny-member-root`	Service control policy	Root	Denies: Every action by a member account's root user, except in a root session started with sts:AssumeRoot (the condition `Null: {aws:AssumedRoot: true}`), which is how root tasks run
`sf-protect-account-defaults`	Service control policy	Root	Denies: Turning off EBS default encryption; changing S3, snapshot, or AMI Block Public Access, RDS snapshot sharing, or KMS key rotation outside the platform roles; and creating IAM users, access keys, or console passwords outside the PlatformIdentityAdmin role
`sf-require-workload-boundary`	Service control policy	Root	Denies: Creating a role or user, or setting a permissions boundary, without the sf-workload-boundary boundary, and editing, deleting, or removing the boundary itself, outside the platform and identity administrator roles
`sf-deny-unused-regions`	Service control policy	Workloads, Sandbox, Suspended	Denies: Every action in the 11 unused Regions, except global services, the security services, and the platform roles
`sf-rcp-enforce-tls`	Resource control policy	Root	Denies: Requests to S3, KMS, SQS, Secrets Manager, and STS without TLS (aws:SecureTransport false)
`sf-ec2-imdsv2-default`	EC2 declarative policy	Root	Sets: Instance metadata defaults for new launches: IMDSv2 required, hop limit 2

Guardrail	Kind	Where it applies	What it does
`sf-ec2-block-public-sharing`	EC2 declarative policy	Root	Sets: Snapshot block public access (block all sharing) and AMI block public access (block new sharing)

The account defaults every member account, existing and new, receives from StackSets; `sf-management` gets the same defaults from a self-managed stack, because a service-managed StackSet never deploys there. The templates are in the Audit Report Appendix (SF-SEC-050 and the findings it names).

Default	Kind	Where it applies	What it sets
`account-baseline`	StackSet default	every member account; sf-management gets a self-managed stack	S3 account-level Block Public Access, alternate SECURITY contact, AWSSupportAccess role (the password policy is set by CLI, because CloudFormation has no resource for it)
`ebs-default-encryption`	StackSet default	every member account, 17 Regions; sf-management gets a self-managed stack	EBS encryption by default with a customer managed key, alias/sf-ebs, that grants the Auto Scaling role and the forensics account (a snapshot under the AWS managed aws/ebs key cannot be shared)
`security-alarms`	StackSet default	every member account, 17 Regions; sf-management gets a self-managed stack	EventBridge rules for root and break-glass sign-in, trail changes, and kms:Sign by anyone but the OTA signer, to PagerDuty
`vpc-baseline`	StackSet default	every VPC in a member account; sf-management gets a self-managed stack	Flow logs to the log archive; default security groups with no rules

How to change one without an outage:

1. **Write the change as a pull request** that shows the full policy document, not a diff of a diff.
2. **Test it in the Sandbox OU first.** Attach it there, run the platform pipelines and one ordinary administrator workflow for a business day, and read CloudTrail for denied calls.
3. **Widen to non-production, then the root.** Wait 24 hours with no denied call that a person or pipeline needed before each step.
4. **Know the exits.** An exception role is a named role in the policy's condition, with a written reason and an end date. A rollback is detaching the policy from the OU it was attached to last. Nobody edits a live policy in the console.
5. **Remember the management account.** Service control policies and resource control policies do not apply to it, and service-managed StackSets never deploy to it. Protect it with its own self-managed stacks, the alarms, and the hardware-key root, not with a deny; test every policy in a sandbox account, not in `sf-management`.

Also yours: 2 risk acceptances, and the clock on each.

The risk acceptances in force and when each ends.

Finding	Severity	Accepted by	Expires	Who renews it
SF-SEC-002	Critical	Marcus Oyelaran and Priya Raman	December 15, 2026	The signers, in writing, with the current count in front of them
SF-SEC-049	Low	Marcus Oyelaran	August 14, 2027	The signers, in writing, with the current count in front of them

Keeping the score where it is

The monthly review

One hour, the first Tuesday of the month, chaired by Nadia Petrova, with Elena Vasquez, Grace Liu, Sam Okafor, and Marcus Oyelaran or a delegate. The standing agenda:

The standing agenda of the monthly security review.

Item	What to look at	A signal worth acting on
The score	Security Hub CSPM's own score by account, against last month's (it covers every enabled control, so it will not match the Vantalect scan score of 71 controls)	Any account down more than a point
New failures	Failed CRITICAL and HIGH controls first seen this month, by account and by the team that owns it	More than a handful, or any in a production account
Coverage	Every account on every detection service in every Region, from the delegated administrators	A service that is not on in a new account or Region
Exceptions	The exception register, and every acceptance that expires within 60 days	An exception nobody can explain; an acceptance with no renewal plan
The roadmap	Every finding above, with its owner's date	A date that moved twice
Credentials	The credential report: IAM users, keys older than 90 days, users with no matching person	Any user nobody can match to a person
Unused roles	Roles unused for 90 days (`SF-SEC-048`)	Growth since last month
Runbooks	The quarterly test results, once a quarter	A page time that is not what the runbook says
Cost	The detection stack against its monthly forecast	An increase nobody planned

To read the score without waiting for a console, keep a saved insight in the Security Hub CSPM aggregation Region and read it in the review:

```bash
aws securityhub create-insight --region us-east-1 --name sf-failed-critical-high-by-account \
  --group-by-attribute AwsAccountId \
  --filters '{"ComplianceStatus":[{"Value":"FAILED","Comparison":"EQUALS"}],"RecordState":
[{"Value":"ACTIVE","Comparison":"EQUALS"}],"SeverityLabel":
[{"Value":"CRITICAL","Comparison":"EQUALS"},{"Value":"HIGH","Comparison":"EQUALS"}]}'
aws securityhub get-insight-results --region us-east-1 --insight-arn <insight-arn>
```

The exception process

Exceptions are written before they take effect, time-boxed, narrow, and counted. Exactly who approves each kind and where it lives is in the week-3 report; the monthly review is where they are read. The rule that keeps the process honest is the last one: an exception nobody can explain is removed, and what breaks tells you who needed it.

The new-account baseline

A new account is the commonest source of new failures. The guardrails are built so that a new account gets nearly all of the baseline with no one remembering to ask: place the account in the right OU and it inherits the policies; enroll it in Control Tower and the StackSets deploy; the delegated administrators enable the detection services. This is the checklist for the day it happens, with the verification for each.

What a new account gets, and how to confirm it.

Step	How	How to verify
Create it through Control Tower Account Factory, in the OU that matches its job	A pull request to `platform-landing-zone`	The account appears in the OU, enrolled, with the Control Tower baseline SUCCEEDED
The service control policies, the resource control policy, and the declarative policy apply	Inherited from the OU and the root; nothing to do	`aws organizations list-policies-for-target --target-id <account-id> --filter SERVICE_CONTROL_POLICY`
The StackSet defaults deploy	`account-baseline`, `ebs-default-encryption`, `security-alarms`, and `vpc-baseline` use automatic deployment	`aws cloudformation list-stack-instances --stack-set-name <stack-set-name>` shows CURRENT for the new account
GuardDuty, Security Hub CSPM, Config, Access Analyzer, and Inspector turn on	Auto-enable from the delegated administrators in every Region (for Inspector, auto-enable reaches new accounts only; run the administrator's `enable` call for any account that joined before it)	The coverage report in the monthly review shows the account on every service in every Region
Root credentials are removed	Centralized root access management	An audit root session with the `IAMAuditRootUserCredentials` task policy shows none
People reach it through Identity Center	A permission-set assignment in `platform-landing-zone`	The Identity Center sign-in log; no IAM users in the credential report
It has an owner	Team and Owner tags, set at creation	The tagging coverage report (`SF-SEC-047`)

If any row fails to verify, the account is not ready for a workload. A failed row is a bug in the baseline, and the fix goes in `platform-landing-zone`, not in the account.

The quarterly re-scan

Run the same scan every quarter against the whole organization, compare it with the week-six result of 91.1% (86.3% before the suppressions the work earned), and file every drop as a finding with an owner. The week-six report states how the scan was run and what was suppressed, so that

the comparison stays honest. If the score falls by more than a point between reviews, find out that week why. The cost of the review is an hour a month; the cost of not doing it is the next engagement.

© 2026 Vantalect, LLC. Sample deliverable; StrataForge Robotics is fictional.

AWS, Amazon Web Services, AWS Graviton, and the AWS service names are trademarks of Amazon.com, Inc. or its affiliates. Kubernetes is a registered trademark of The Linux Foundation. GitHub is a trademark of GitHub, Inc. NVIDIA and Isaac Sim are trademarks of NVIDIA Corporation. SAP and SAP S/4HANA are trademarks of SAP SE. Okta is a trademark of Okta, Inc. PagerDuty is a trademark of PagerDuty, Inc. SOC 2 is a registered trademark of the American Institute of Certified Public Accountants. CIS Benchmarks is a trademark of the Center for Internet Security, Inc. Vantalect is not affiliated with or endorsed by any of them. StrataForge Robotics is fictional.