> **Lab-built. Not exercised.** This pack is adapted from a lab-built runbook. The runbook, its KQL queries and this tabletop have **not** been run against a production tenant, a live incident, or a team. Nothing here reports detection results, timings or false-positive rates. The scenario is fictional. Treat this as a starting point and test it in your own environment.

Companion to the open-source runbook entra-account-compromise-ir and the post Order of Operations: The First Hour of a Compromised Cloud Account.

Contents

1. How to use this pack
2. Facilitator guide
3. Scenario setup
4. Inject timeline (one page)
5. Injects 1-6
6. Role cards
7. Discussion questions
8. Readiness score and debrief sheet
9. What NOT to claim after running this
10. Sources

1. How to use this pack

Item	Detail
Length	60 minutes: 5 min brief, 45 min injects, 10 min hotwash
Group size	3-6 people plus a facilitator. A scribe is strongly recommended
Audience	Small IT or security teams that run Microsoft Entra ID (Microsoft 365)
Format	Discussion-based. Nobody touches a live tenant during the exercise
You need	This pack printed (one role card per player), a clock, a whiteboard or shared doc for the timeline, and your own runbook or on-call list if you have one
Goal	Practise the containment order - **disable → revoke sessions → reset password → hunt leftovers** - and surface who can actually do each step today
Output	A short list of gaps, each with an owner and a date (section 8)

The one idea this exercise tests. Each containment step leaves a gap that the next one closes. Revoking sessions doesn't help if the attacker still knows the password. Resetting the password alone doesn't end every session that already exists. Disabling the account stops new sign-ins, but tokens that were already issued can keep working until they expire. And none of the three removes what the attacker left behind (inbox rules, OAuth consent, new MFA methods). That's the "order of operations" from the post.

2. Facilitator guide

Before the session (15-30 minutes of prep)

- ⬛ Read the runbook's Contain and Eradicate sections (docs/runbook.md).
- ⬛ Decide whether to use the fictional company below or swap in your own org's names (keep the attacker IPs in the RFC 5737 documentation ranges).
- ⬛ Find out, privately, who on the team currently holds which Entra admin roles. You'll compare that with what players say in Injects 2-4.
- ⬛ Print role cards (section 6) and one debrief sheet (section 8).
- ⬛ Pick a scribe. The scribe records decisions, times and gaps. Not blame.

Running it

1. **Brief (5 min).** Read the scenario setup aloud. State the ground rules:
    - It's a discussion. Say what you would do and who would do it.
    - "I don't know" and "I don't have access to that" are useful answers. They become action items.
    - We're testing the plan, not the people. Nobody is scored as an individual.
2. **Injects (45 min).** Read each inject aloud at its time mark. Give players the prompts. Move on when time is up even if the discussion isn't finished; note open items.

3. **Hotwash (10 min).** Walk the debrief sheet. End with owners and dates.

Facilitator tips

- Keep the clock visible. Time pressure is part of the lesson.
- If players jump straight to "reset the password," ask: *what happens to sessions that already exist?*
- If players say "revoke sessions" first, ask: *what stops the attacker from signing straight back in?*
- If players say "we're done" after the reset, play Inject 5.
- Push for names, not job titles: "Who, by name, is on call at 06:15 on a weekday?"
- Don't argue product licensing at the table. Note "need to check our license" as an action item.

The intended "aha" moments (from the repo's facilitator notes)

- Revoking sessions doesn't kill access tokens that were already issued straight away, including tokens held by an OAuth app the user consented to.
- An attacker-added MFA method survives a password reset unless someone removes it.

3. Scenario setup (read aloud)

> **Fictional.** Fabrikam is Microsoft's sample company name. Every person, address, IP and time below is made up. IPs are from the RFC 5737 documentation ranges.

- **Company:** Fabrikam, Inc., a small business using Microsoft 365 and Microsoft Entra ID. The IT team is small. There is no 24/7 SOC.
- **Target account:** `j.rivera@fabrikam.example` accounts payable specialist. Not an admin.
- **Attacker IP:** 203.0.113.45
- **Time:** A weekday morning. The first alert lands at 06:10 local.
- **Assume for this exercise:** sign-in and audit logs are being sent to a Log Analytics workspace, so the runbook's queries are available. (If that isn't true for your org, that's your first action item. See the retention note in Inject 6.)

4. Inject timeline (one page)

Clock	Inject	Containment step it drives	Runbook ref	Query ref
0:00-0:05	Brief	-	-	-
0:05	**1. "Approve to continue"**: MFA denial burst, then a success	Triage → decision to contain	§2 Triage	03, 11
0:13	**2. The help desk call**	**Step 1: Disable the account**	§3.1	-
0:21	**3. Still signed in?**	**Step 2: Revoke sessions / refresh tokens**	§3.2, timing caveats	11, 12
0:29	**4. The password**	**Step 3: Reset password, force change**	§3.3	12
0:37	**5. What they left behind**	**Step 4: Hunt leftovers** (inbox rules, OAuth consent, MFA methods)	§4.1-4.3, 4.8	06, 07, 08, 10
0:45	**6. Pressure to restore**	Verify, then recover	§3 Verify, §5	12
0:50-1:00	Hotwash	Debrief sheet	§6 Lessons learned	-

Containment order this pack teaches:

```
1. DISABLE          →  2. REVOKE SESSIONS    →  3. RESET PASSWORD    →  4. HUNT
   stop new              stop existing             kill the stolen         inbox
   sign-ins              sign-ins renewing         credential              grant
                                                                           device

                        ↓ then VERIFY (query 12). Still active? Go back to
```

The runbook also lists optional containment steps that this 60-minute version only touches in discussion: mark the user compromised in ID Protection (needs Entra ID P2), block clear attacker infrastructure with a Conditional Access policy (excluding break-glass accounts), and disable the user's devices.

5. Injects

Inject 1 (0:05) - "Approve to continue"

Read aloud: An alert fires. `j.rivera` has 9 MFA denials between 06:02 and 06:10 from 203.0.113.45. There is one successful sign-in from the same IP at 06:11.

Prompts

- What do you check first? Who checks it?
- One success after a run of denials: is that enough to contain without reaching the user first? Who has the authority to decide?
- How do you contact the user? (Hint: not by email to the mailbox you think is compromised.)

Facilitator notes

- Runbook triage rule: *if you can't rule out compromise within your agreed time, treat it as real and contain. Containment can be reversed. Data loss can't.* Does the team have an "agreed time"? If not, that's an action item.
- Query 03 looks for MFA denial bursts and success after denial; query 11 pulls one user's sign-in timeline by IP.
- ATT&CK: T1621 (Multi-Factor Authentication Request Generation), then T1078.004 (Valid Accounts: Cloud Accounts).

Inject 2 (0:13) - The help desk call → Step 1: Disable

Read aloud: J. Rivera calls the help desk: "My phone kept buzzing, so I hit approve to make it stop."

Prompts

- What does the help desk say and do? What must it **not** do? (For example, reset MFA over the phone without verifying the caller.)
- Who disables the account, right now, by name? Where do they do it?
- Does that person hold the role needed? (Microsoft Learn: disabling a non-admin user needs User Administrator; an admin account needs Privileged Authentication Administrator.)

Facilitator notes

- Disabling the account blocks new sign-ins. It does not by itself end tokens already issued.
- If nobody at the table can name who holds the role, record it. Pre-approving responder roles is runbook §0 Prepare.

Inject 3 (0:21) - Still signed in? → Step 2: Revoke sessions

Read aloud: The account was disabled at 06:20. At 06:24 the sign-in timeline still shows non-interactive activity for `j.rivera` from 203.0.113.45.

Prompts

- Why could activity continue after the account is disabled?
- Who revokes sessions and refresh tokens? (Entra admin center: user > **Revoke sessions**; or Graph `revokeSignInSessions` / `Revoke-MgUserSignInSession`.)
- How long do you expect it to take before the attacker is fully out?

Facilitator notes (from Microsoft Learn)

- Revoking sign-in sessions invalidates the user's refresh tokens (and browser session cookies). Microsoft notes there may be a small delay of a few minutes.
- Access tokens already issued stay valid until they expire: by default a random 60-90 minutes (75 on average), unless the app supports Continuous Access Evaluation (CAE). CAE-capable services such as Exchange Online, SharePoint Online and Teams can act in near real time on events like account disablement, password reset, and an admin revoking refresh tokens.
- Sessions issued by an app itself (not Entra) are controlled by that app. Entra can't directly revoke them.
- This is why the team does all three steps, not one.

Inject 4 (0:29) - The password → Step 3: Reset

Read aloud: Someone asks: "The account's disabled and sessions are revoked. Do we still need to reset the password? And the user is asking for a

new one now."

Prompts

- Yes or no on the reset, and why?
- How do you set the new password so the user, not the attacker, gets it? (Force change at next sign-in; hand it over out of band.)
- Is this account cloud-only or synced from on-premises AD? What changes if it's synced?

Facilitator notes

- The attacker still knows the old password. If the account is re-enabled before a reset, they can sign straight back in.
- For hybrid accounts, Microsoft Learn says to disable the user in on-premises AD and reset the password **twice** there (to mitigate pass-the-hash risk where replication is delayed).
- Verify: query 12 looks for the containment actions in the audit log (for example `Disable account` or `Update user` `Update StsRefreshTokenValidFrom Timestamp`, `Reset password` or `Reset password (by admin)`) and for any successful sign-ins after the containment time. If sign-ins still succeed, go back to Step 1.

Inject 5 (0:37) - What they left behind → Step 4: Hunt leftovers

Read aloud: Three findings come back from the logs:

1. A new inbox rule created from 203.0.113.45 at 06:14. It moves messages containing "invoice" or "payment" to the RSS Subscriptions folder.
2. A `Consent to application` event at 06:20 by `j.rivera` for an app named "Fabrikam Docs Viewer" with delegated `Mail.Read` and `offline_access`.
3. `User registered security info` at 06:16 from 203.0.113.45.

Prompts

- What is each one for, from the attacker's point of view?
- Which evidence do you capture **before** you remove anything?
- Does disabling, revoking and resetting stop the app? Why or why not?
- Disable or delete the app? (Microsoft's app-consent playbook says disable, because a deleted app can come back if another user consents.)
- How would you check whether any **other** users have the same rule, the same app, or sign-ins from the same IP?
- Who needs to be told outside IT (finance, vendors)?

Facilitator notes

- Inbox rule: hides payment-related replies (ATT&CK T1564.008). Query 06.
- OAuth consent: app keeps reading mail; tokens already issued stay valid until they expire (T1528). Query 07.
- New MFA method: survives a password reset unless removed (T1556.006). Query 08.

- Runbook §4.8: search all users for the same IPs, app IDs and rule patterns. Each new account goes back to Step 1.

Inject 6 (0:45) - Pressure to restore

Read aloud: The accounts payable manager needs J. Rivera working before the noon payment run. Separately, someone asks: "When did this actually start? Could it have been weeks ago?"

Prompts

- Which checks must pass before you re-enable the account?
- How does the user re-register MFA safely?
- How far back can you actually look in your logs today?

Facilitator notes

- Runbook §5: confirm eradication (re-run queries 06, 07, 08, 10), confirm identity out of band before re-enabling, have the user re-register MFA (phishing-resistant where available), remove temporary access, then watch more closely for an agreed period.
- Retention (Microsoft Learn): Entra keeps sign-in and audit logs for **7 days on Entra ID Free** and **30 days on P1/P2** unless exported. Upgrading doesn't bring back data that has already expired. The Microsoft 365 unified audit log is separate (Purview Audit Standard: 180 days).

6. Role cards

Cut along the lines. One card per player. Players should stay in role.

ROLE CARD: IT Lead / Incident Lead

You own: the decision to contain, the order of steps, and who does what. **Your first question:** "Can we rule out compromise in the time we agreed? If not, we contain." **You must decide:**

- When to declare an incident and open a ticket and timeline (times in UTC)
- Who disables, revokes and resets, and in what order
- When to escalate (for example: the user holds an admin role, or tenant-level settings like federation changed)
- What "done" means before the account is re-enabled

Watch out for: doing the right steps in the wrong order; declaring victory after the password reset.

ROLE CARD: Help Desk

You own: the first conversation with the user, and identity checks. **Your first question:** "How do I know this caller is really J. Rivera?" **You must decide:**

- How you verify the caller before changing anything
- How you reach the user out of band (phone or chat you already trust, **not** the possibly compromised mailbox)
- How you hand over a new temporary password safely

You must NOT: reset MFA or give out a new password over the phone to an unverified caller. Don't promise the user a time to be back online until the IT Lead says so.

ROLE CARD: Manager (business owner)

You own: the business impact and the people outside IT. **Your first question:** "What can't wait, and what can?" **You must decide:**

- Whether you accept the user being locked out for now (agreeing that doubt means contain)
- Who in finance and which vendors need a heads-up about possible payment fraud
- Who approves extra spend or outside help if the team needs it

Pressure you'll bring (Inject 6): "J. Rivera needs to run payments by noon."

ROLE CARD (optional): SOC / Security Analyst

You own: the logs and the queries. **Your first question:** "What does the sign-in timeline show, by IP?" **You must decide:**

- Which query you run for each inject (03, 11 for triage; 12 to verify; 06, 07, 08, 10 for leftovers)
- What evidence to export before anything gets removed
- Whether the same IP, app or rule pattern shows up on other users

Know your limits: how far back your logs go, and whether the queries have been tested in your workspace. (In the source repo, they haven't.)

ROLE CARD: Scribe (recommended)

You own: the timeline and the gap list. Record every decision with a time, who made it, and why. Mark each "I don't know" or "I don't have access" as a gap. Hand the list to the facilitator for the debrief sheet.

7. Discussion questions

Use during the hotwash or for a follow-up session.

1. Could we have disabled the account within 15 minutes of Inject 1? What slowed us down?
2. Did everyone agree on the containment order? Where did we disagree?
3. Did anyone need access or a role they don't have today? Who approves it, and can that happen before the next incident?
4. Have we agreed, in advance and with leadership, that disabling an account on suspicion is acceptable, including a senior person's account?
5. How do we contact a user whose mailbox may be compromised?
6. How far back can we see sign-in and audit logs today? Is that enough to find a compromise that started weeks ago?
7. Would a Conditional Access setting have prevented this or made it harder (for example, number matching or phishing-resistant MFA, or blocking user consent to unverified apps)?
8. How would we find other accounts hit by the same campaign?
9. Which detection would have fired first in **our** tenant today? Have we actually tested it?
10. What do we change first: the runbook, the queries, our policies, or our on-call list? Who owns each change, and by when?

8. Readiness score and debrief sheet

> This scores the plan and the access, not the people. (The source repo's facilitator notes say: don't score the team; record gaps as action items.) Use it to rank what to fix first.

Date: _____ Facilitator: _____ Scribe: _____ Participants (roles only): _____

A. Containment readiness (0 = no, 1 = partly, 2 = yes)

#	Question	0 / 1 / 2	Notes
1	We had an agreed rule for when doubt means contain		
2	We named a person (not a job title) who could disable the account		
3	That person holds the needed role today		
4	We put disable → revoke → reset in the right order without prompting		
5	We knew revocation isn't instant and why (token lifetime, CAE)		
6	We knew whether the account is cloud-only or synced, and what changes		
7	We had a way to verify containment (audit log + no new sign-ins)		
8	We looked for inbox rules, OAuth consents and new MFA methods		
9	We captured evidence before removing anything		
10	We checked other users for the same IP, app or rule		
11	We contacted the user out of band		
12	We know how far back our sign-in and audit logs go		
	Total (out of 24)		

There is no pass mark. Any 0 is a gap worth an owner and a date.

B. Timeline (from the scribe)

Time (exercise clock)	Decision / action	Who	Notes

C. Action items

#	Gap	Fix	Owner (name)	Due date
1				
2				
3				
4				
5				

D. One sentence for leadership

"If one of our users' accounts were compromised this afternoon, we _____."

9. What NOT to claim after running this

Running a discussion exercise is useful. It doesn't prove much on its own. After the session, **don't** say:

- **"We tested our incident response."** You discussed it. Nothing was executed in a tenant.
- **"Our detections work."** The queries in the source repo have not been run against real data. A tabletop doesn't test them either.
- **"We can contain an account in X minutes."** You haven't measured it. Time a real drill in a lab or test tenant first.
- **"Revoking sessions kicks the attacker out immediately."** Microsoft documents a delay of a few minutes, and access tokens already issued can stay valid until they expire unless CAE applies.
- **"Resetting the password fixes it."** Not on its own. Sessions, OAuth grants and attacker-added MFA methods can survive it.
- **"We're compliant with NIST SP 800-61."** The runbook maps its steps to SP 800-61 phases. A tabletop doesn't establish compliance with anything.
- **"This is Mike Sosa's / Microsoft's production playbook."** It's a lab-built starting point, adapted from an open-source repo. It is not Microsoft guidance and has not been run in a production tenant.

OK to say: "We ran a 60-minute discussion exercise on Entra ID account compromise, found [N] gaps, and assigned owners and dates."

10. Sources

Mike's public work

- Runbook, queries and original 90-minute tabletop: https://github.com/mikedsosa/entra-account-compromise-ir
  - Runbook: https://github.com/mikedsosa/entra-account-compromise-ir/blob/main/docs/runbook.md
  - Tabletop (Fabrikam): https://github.com/mikedsosa/entra-account-compromise-ir/blob/main/docs/tabletop-fabrikam.md
  - Sources list: https://github.com/mikedsosa/entra-account-compromise-ir/blob/main/SOURCES.md
- Post: Order of Operations: The First Hour of a Compromised Cloud Account: https://mikedsosa.com/blog/order-of-operations/

Microsoft Learn (re-read 2026-11-05)

- Revoke user access (disable, revoke, hybrid reset twice, roles, app session tokens): https://learn.microsoft.com/en-us/entra/identity/users/users-revoke-access
- revokeSignInSessions (refresh tokens and browser session cookies; "a small delay of a few minutes"; `User.RevokeSessions.All`): https://learn.microsoft.com/en-us/graph/api/user-revokesigninsessions?view=graph-rest-1.0
- Access token lifetime (random 60-90 minutes, 75 on average): https://learn.microsoft.com/en-us/entra/identity-platform/access-tokens

- Continuous access evaluation (critical events; Exchange Online, SharePoint Online, Teams): https://learn.microsoft.com/en-us/entra/identity/conditional-access/concept-continuous-access-evaluation
- Data retention (7 days Free, 30 days P1/P2; not retroactive): https://learn.microsoft.com/en-us/entra/identity/monitoring-health/reference-reports-data-retention
- Purview Audit (Standard) 180-day retention: https://learn.microsoft.com/en-us/purview/audit-solutions-overview
- Audit activity names used in verification: https://learn.microsoft.com/en-us/entra/identity/monitoring-health/reference-audit-activities
- App consent incident response playbook (disable rather than delete): https://learn.microsoft.com/en-us/security/operations/incident-response-playbook-app-consent
- confirmCompromised (Entra ID P2): https://learn.microsoft.com/en-us/graph/api/riskyuser-confirmcompromised?view=graph-rest-1.0

NIST / MITRE

- NIST SP 800-61 Rev. 2: https://csrc.nist.gov/pubs/sp/800/61/r2/final
- NIST SP 800-61 Rev. 3 (supersedes Rev. 2): https://csrc.nist.gov/pubs/sp/800/61/r3/final
- T1621: https://attack.mitre.org/techniques/T1621/
- T1078.004: https://attack.mitre.org/techniques/T1078/004/
- T1564.008: https://attack.mitre.org/techniques/T1564/008/

- T1528: https://attack.mitre.org/techniques/T1528/
- T1556.006: https://attack.mitre.org/techniques/T1556/006/

Other

- RFC 5737 (IPv4 documentation address ranges): https://www.rfc-editor.org/rfc/rfc5737

Fictional scenario. Lab-built material, not exercised, not run against a production tenant. Views are my own.