EXECUTIVE BRIEF

Backup, Recovery and Incident Response

Prepared for ownership and executive team | Environment: **5 offices, 170+ users, 300 endpoints**

> *Redacted work sample. Client name, hostnames, addresses and personnel removed. Everything else stands as delivered.*

Where things stand

You've been encrypted twice in three months, so I'm not going to spend much time on whether it could happen a third time. What matters more is what it would cost you if it did, and right now the honest answer is quite a lot - every office, some number of days nobody could predict, and no way to tell a patient or a regulator what was actually taken.

That isn't bad luck. The backup arrangement had never been tested, and the access model lets one compromised workstation reach everything. Both are fixable, and neither costs anything like what a third event would.

How often we take a copy

This is the decision that drives most of the rest, and it's yours rather than IT's.

How often we copy your data decides how much work your staff have to redo afterward. Copying overnight means you lose a day. Copying every half hour means you lose an hour. Across 170 people that difference works out to roughly a day of billable work, every time something happens.

What I'd suggest is weekly full backups, daily incrementals, and versioned copies every 30 to 60 minutes. I've priced all three so you can see what the shorter interval actually buys you.

Why the backups didn't help

Two things, and both are common enough that I wouldn't read much into them.

The copies sat behind the same credentials as the live systems, so when somebody took an administrator account they got the backups along with it. At least one copy needs to live somewhere the production environment can't reach.

The other is that nobody had ever restored from them. Until you've actually done a restore you don't really know what you've got, so we now test quarterly against a separate system, bring back a whole server rather than a single file, and write down how long it took. When I say recovery runs about eleven hours, that comes from doing it rather than from a vendor's datasheet.

How long they were inside

Attackers usually sit in a network for weeks before they trigger anything, partly so the encryption has time to reach the backups as well.

So if you're keeping seven days of history and somebody was present for a month, every copy you hold is already affected. Retention needs to run longer than they're willing to wait, and that's most likely why the second event went the way it did.

If it happens again

The order matters more than the speed here. Restoring before we'd found the source is what led to the second event, so the sequence below is deliberate.

Contain first. Lock the affected shares at network, share and file level before anyone starts investigating. A share that's still writable while we're looking at it is still being encrypted.

Work out what recovery is even possible before picking a strategy, which means checking backup history early rather than at the end.

Find the carrier. Encrypted files stop matching their original format internally, and that tells us both what got hit and where it came in. Without that step the restore tends to bring the problem back with it.

Remove, restore, verify. We destroy infected content rather than trying to clean it, restore from the last copy that predates the infection, then scan what we've restored before it goes back into service. If the infection is older than the restore point, recovery just reintroduces it, and this is the step that usually gets skipped.

Bring access back a layer at a time, which in your case means over about 72 hours. There'll be pressure to switch everything on at once, but staging it means anything we missed stays in one tier instead of reaching all five offices.

And write it down as you go. Insurers and regulators both ask afterward, and nobody's memory holds up well under that.

The part that costs more than the downtime

Locked files are disruptive. Files that left the building are something else, and that's usually where the real cost sits.

So while recovery is running we look separately at whether data actually went anywhere - sign-in records, what was opened and copied, mail forwarding rules, outbound traffic over the previous few weeks.

For a practice holding patient records, that answer decides whether you're cleaning up or notifying people, and those are very different in cost, timeline and legal exposure. You'll have it in writing the same day either way.

What we need from you

A decision on the copy interval. Thirty minutes or overnight, both priced.

Somebody named who can authorize a shutdown. One person needs to be able to say "cut access to all five offices" at two in the morning without calling a meeting, and they should know that's their call before they ever need to make it.

Annual staff training, with quarterly reinforcement. Every incident here started with somebody clicking something. What's in this document limits the damage; training is what reduces how often it happens.

Sign-off before changes. Nothing gets altered - access, systems, credentials - without a recorded approval from you, even in the middle of an incident. That record is what lets you reconstruct your own decisions later when somebody asks.

Where you stand now

Containment, source identification, staged restoration and this plan are all done. You've got a tested recovery path and a written procedure, where three months ago there was neither.

What's left is the interval decision and the training cadence. Both are yours to make, and both cost considerably less than the last two months did.

Technical detail - retention schedules, access model, segmentation design and restore test results - is in the accompanying operational appendix.

OPERATIONAL APPENDIX

Backup, Recovery and Incident Response

Companion to the Executive Brief | For IT and operations | **5 offices, 170+ users, 300 endpoints**

> *Redacted work sample. Hostnames, addresses, IP ranges, vendor account numbers and personnel removed. Schedules, thresholds and procedures stand as delivered.*

This carries the detail behind the executive summary. It's written for whoever runs the environment day to day, including whoever inherits it next.

1. Backup schedule and retention

Tier	Frequency	Retains	Recovery point
Full	Weekly, Sunday overnight	8 weeks	7 days
Incremental	Daily, overnight	30 days	24 hours
Version copies	Every 30-60 min, business hours	14 days	30-60 minutes
Monthly archive	Month end	12 months	30 days

Retention needs to run longer than an attacker typically sits in the network. I'd treat thirty days of incrementals as a starting point rather than a target, since a seven-day window against a month of dwell time means every copy you've kept is already affected.

The version-copy interval is really a cost decision. I'd price 30-minute and 60-minute separately and put both in front of the business. What you're buying is storage and I/O overhead, and what comes back is staff hours nobody has to redo.

2. Isolation

These are the ones I wouldn't compromise on.

- At least one copy immutable or offline, somewhere production can't reach using production credentials.
- Backup infrastructure authenticating separately from the production domain. If taking a domain admin account also lets somebody delete backups, you don't really have a recovery tier.
- Backup admin accounts kept distinct from production admin accounts, with MFA, and not used for day-to-day work.
- Console access restricted by source, logged, and reviewed monthly.

3. Restore testing

Quarterly at minimum, and written down.

- Restore to a non-production target. Don't test by overwriting live data.
- Test a full system restore rather than pulling back a single file. File-level restores don't tell you much about whether you could actually recover.
- Measure and record how long it took. Any recovery-time commitment you make should come from a restore you've watched, not from a vendor's figure.

- Check the application works afterward, not just that the files are present. A database that won't mount hasn't really been restored.
- Report results to the business owner, including the tests that failed.

For each test, record the date, source copy, target, elapsed time, what you verified, what failed, and what changed as a result.

4. Access model

Least privilege by default, with administrative rights granted per task and per window rather than held standing.

- Named admin accounts, separate from daily-use accounts, never shared.
- MFA on all administrative and remote access, without a bypass list.
- Permissions through security groups only. Direct folder permissions stop being auditable within about two years, and they're usually why nobody can answer "who has access to this?"
- Quarterly access review covering staff, contractors and vendors. Contractor access gets an expiry date when it's created.
- A break-glass account held by the business rather than by IT or an outside provider. Sealed, documented, tested annually.

For joiners, movers and leavers: same-day de-provisioning, with evidence rather than assertion. The ticket doesn't close until the disabled account, revoked sessions, reclaimed license and mailbox disposition are all recorded.

5. Segmentation

- File shares segmented by function and site, so one compromised workstation can't move across the whole data estate.
- Cross-site access granted explicitly where there's a business need, rather than inherited by default.
- Workstation-to-workstation traffic restricted. In this environment there's no reason endpoints need to reach each other directly.
- Backup network segment kept off the user VLANs.

6. Incident response, operational detail

6.1 Containment

Lock at all three levels before anyone starts investigating: network, share and file. A share that's still writable while you're looking at it is still being encrypted.

Record the time containment went on. It bounds the damage window, and every estimate after that depends on it.

6.2 Recovery position

Work this out before choosing a strategy:

- The last known-good copy and its timestamp.
- Whether that copy is still within retention or has aged out.
- Whether it's reachable and uncompromised.

- Estimated restore duration, taken from the most recent test.

6.3 Finding the infection and the carrier

Look through file history for files whose headers no longer match their declared format. Encrypted content fails header validation whatever the extension says.

That gives you both the affected set and, ordering by timestamp, the earliest affected file, which points at the carrier and the entry point. Skip it and the restore tends to bring the infection back with it.

Correlate against endpoint detection logs, authentication events, mail gateway logs for the preceding 30 days, and any file-share audit trail.

6.4 Eradication and restore

- Remove infected content completely rather than cleaning it in place.
- Restore from the last copy predating the earliest affected file, rather than simply the most recent one.
- Scan the restored data before it goes back into service. If the infection is older than the restore point, recovery reintroduces it, and this is the step that gets skipped most often.
- Scan the other shares to confirm nothing moved laterally.

6.5 Staged re-enablement

Bring access back one tier at a time over a defined window, which here means about 72 hours.

The order I'd suggest is core infrastructure, then per-site file access, then remote access, then external integrations. Hold each tier long enough to watch it before opening the next.

Expect pressure to turn everything on at once. Staging it means anything missed stays in one tier instead of reaching all five offices.

7. Data compromise assessment

Run this alongside recovery rather than after it. What you're establishing is whether data actually left, as distinct from being encrypted in place.

- Authentication and sign-in logs - sources, times, unusual geography, impossible travel.
- File access history: what was opened, copied or moved, and when.
- Mail rules and forwarding, which is a common persistence and exfiltration path and easy to overlook.
- Outbound traffic volume for the 30 days before the event.
- Cloud storage sync logs where they apply.

That's what decides whether the organization is cleaning up or disclosing, and in a regulated environment the difference carries statutory obligations and fixed timelines. Findings go to named stakeholders the same day, in writing, whichever way they fall.

8. Change control

No change to infrastructure, permissions, credentials, configuration or data without a recorded approval, including during an active incident.

Log the proposed change, the reason, the risk, who approved it, the date, and the outcome.

This is the first thing to go under time pressure, which is also when it matters most. The log is what lets the organization reconstruct its own decisions afterward for insurers, regulators, and whoever holds this environment next.

9. Review cadence

Activity	Frequency
Restore test, full system, non-production	Quarterly
Access review - staff, contractor, vendor	Quarterly
Backup isolation verification	Quarterly
Break-glass account test	Annually
Staff security training, formal	Annually
Staff reinforcement, short-form	Quarterly
This document, reviewed and reissued	Annually, or after any incident