The Small Agency Password Playbook (2026 Edition)

Practical workflows, templates, and real fixes for managing client credentials

This playbook assumes one thing:
You are running an agency with imperfect systems, limited time, and real client pressure.

The goal is **not perfect security**.
The goal is **clear control using tools you already have**.

Phase 1: Stabilise Immediate Risk (Week 1)

1. Passwords Shared in Slack or Email

What usually happens (short + real)

Client credentials arrive by email or WhatsApp, get pasted into Slack for speed, and then quietly become permanent. No one deletes them because work moves on.

Why this keeps happening

- Slack feels faster than “process”
- There’s no single owner for access
- Temporary fixes become permanent by default

Red flags

- Passwords in Slack history

- Screenshots floating around
- Multiple people saving the same login

What to do instead (with Google Sheets)

Immediate containment flow

```
Client sends credentials
        ↓
Ops adds to ONE private Google Sheet
        ↓
Temporary access granted
        ↓
Slack/email message deleted
```

🔹 Template 1: Client Access Sheet (Temporary)

Create one Google Sheet per client

Use this as the first containment step for any agency.

Client Access Sheet

When to use

- **Client sends credentials**
- **Team needs access fast**
- **You want control without plaintext passwords**

Ops tip
Create one copy per client and restrict edit access to Ops/Owner.

Sheet name

```
Client Name – Credential Access (Internal)
```

Columns

```
Tool / Platform
Login URL
Username / Email
Password Location (NOT password)
Who Has Access
Purpose
Access Start Date
Access End Date
Last Reviewed
```

Password Location examples

- “Ops-managed secure vault”
- “Client-managed”
- “Encrypted storage (internal)”

👉 This **immediately removes plaintext passwords** from chats and inboxes.

Slack message template (copy-paste)

```
I’ve added the login to the client access sheet.
Please don’t copy or reshare.
Access is limited to this task and will be reviewed after.
```

Email reply to client (simple, non-technical)

```
Thanks for sharing.
We’ve stored this securely in our internal access sheet and limited visibility
to only the team members working on your account.
```

---

2. Shared Admin Accounts

What usually happens

Clients hand over one admin login, and everyone uses it because creating users feels like overhead.

Why agencies keep doing this

- Clients don’t guide access setup
- Tools feel confusing
- Deadlines win over cleanup

Risk indicators

- One password breaks multiple workflows
- No idea who made changes
- Hard to answer client questions

What to do instead (decision-based)

Access decision flow

```
Does the tool support multiple users?
                ↓
Yes → Create individual users
No  → Restrict shared credential visibility
```

If individual users ARE supported

- Create one login per person
- Assign the lowest required role
- Remove access immediately when work ends

Real example

- Ads specialist → Google Ads account access
- Designer → CMS editor
- Intern → No production access

If shared login is unavoidable

Template 2: Shared Credential Control Sheet

Columns

```
Tool Name
Shared Username
Who Can Access
Why They Need Access
Access Start Date
Access End Date
Owner (Ops / Lead)
```

Rule
If there's no end date, access should not be granted.

Some teams move from time-bound Google Sheet access to secure, expiring access links that don't expose the password at all. Tools like All Pass Hub support this once teams are ready to centralize.

Phase 2: Structure What You Already Use (Weeks 2–3)

3. Using Google Sheets Without Creating Silent Risk

What usually happens

One giant spreadsheet stores everything. Everyone can edit. Old credentials never get removed.

Why agencies keep doing this

- Sheets are familiar
- Zero setup
- “Temporary” becomes long-term

When Sheets start failing

- Multiple clients
- Contractors rotating
- No clarity on ownership

What to do instead (minimum safe setup)

Sheet structure (mandatory)

One Google Sheet per client
Never one master list.

Required columns

```
Service Name
Login URL
Username / Email
```

```
Password Location
Access Level
Primary Owner
Last Reviewed Date
```

❌ Never store the actual password
✅ Store **where** it lives and **who controls it**

Quarterly review workflow

```
Calendar reminder
        ↓
Sort by "Last Reviewed Date"
        ↓
Remove unused access
        ↓
Update review date
```

This takes **30 minutes per quarter**, not hours.

---

4. Organise Credentials by Client (Not by Tool)

What usually happens

Credentials are grouped by tools, which makes exits and audits painful.

Why agencies do this

- Tools feel tangible
- Clients feel abstract

Problems caused

- Accidental cross-client access

- Slow offboarding
- Confusion during onboarding

What to do instead

Recommended client-first structure

```
Client Name
 ├─ Website & Hosting
 ├─ Marketing Tools
 ├─ Analytics & Reporting
 ├─ Billing (Restricted)
```

Operational benefit
When a freelancer exits, you remove **one client's access**, not 15 credentials.

As agencies grow, role-based access becomes easier to manage when credentials live in a central vault rather than individual sheets. This is where password managers designed for teams (such as APH) remove manual overhead.

Phase 3: Sharing, Rotation, and Reviews (Ongoing)

5. Secure Sharing Without Follow-ups

What usually happens

Access is shared "temporarily" but never reviewed again.

Why

- No expiry defaults

- Ops forgets
- No tracking

What to do instead

Access-sharing rule

Every access must have:

- Purpose
- Start date
- End date

Internal Slack message template

```
Granting access to:
Client:
Tool:
Purpose:
Access expires on:
```

If you can’t fill all four, don’t grant access yet.

---

6. Password Rotation That People Actually Follow

What usually happens

Calendar-based rules are ignored.

Why

- Too disruptive
- Hard to track
- Low perceived value

Practical rotation triggers

Rotate credentials when:

- Someone leaves
- A project ends
- Client relationship changes
- Something feels off

Rotation checklist (paste into Sheet)

```
Password updated
Access confirmed
Old access removed
Review date updated
```

---

7. The 30-Minute Quarterly Access Review

What usually happens

Reviews only happen after a scare.

Why agencies skip them

- Feels optional

- No structure

Template 3: Quarterly Access Review Sheet

Columns

```
Client
Tool
Current Users
Should Still Have Access? (Y/N)
Action Required
Reviewed By
Review Date
```

Calendar reminder text

```
Quarterly Access Review – remove stale access before it becomes a problem
```

Manual quarterly reviews work well early on. More mature teams often rely on automatic access logs and item-level visibility to reduce review effort.

Preparing for Client Security Questions (2026-Ready)

Clients increasingly ask:

- Who can access our systems?
- How fast can access be revoked?
- Where are credentials stored?

Simple, honest answer framework

```
We store credentials centrally,
limit access by role and purpose,
and review access regularly.
```

No jargon. No over-explaining.

What “Good Enough” Looks Like

You’re in a good place if:

- Passwords are not in Slack or email
- Access has owners and expiry
- Offboarding is predictable
- Reviews are routine

That’s real agency security.

Final Note for Ops Managers

Most agencies don’t fail at security because they don’t care.
They fail because **systems never replaced improvisation**.

Reduce the number of places passwords live.
Reduce the number of people who can see them.
Add review points that fit real work.

That’s how you stay secure without slowing down.

Optional (Low Priority, Natural Progression)

As agencies mature, many move from Sheets to a central credential vault that:

- Enforces access by default
- Tracks usage automatically
- Removes manual reviews