RFID Fuzzing

The Short Version

Fuzzing is a way of throwing traffic at a component to induce an error state or exploit a weakness in the system.

Please only fuzz the low frequency system. The HF readers can be hard to replace if they break. Note that I have replicated an issue found in a different brand of controller. This vulnerability is not found in the EWS controllers used.

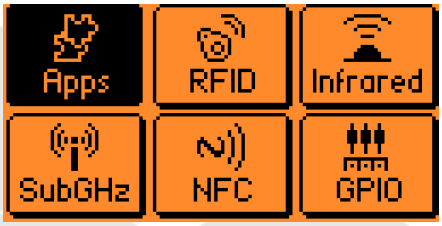
The Long Version

Fuzzing is where you **throw traffic at a system** to either induce an **error state** or to find a **testing / default value** left in the system. We're going to use the latter here. Bear in mind **fuzzing can fry readers** or cause a **system to enter an error state** requiring resetting.

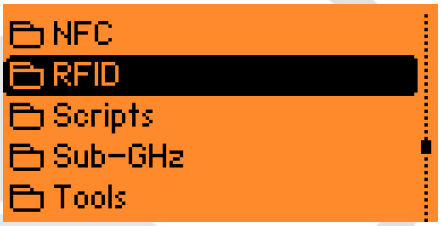
On a Flipper, you'll need a particular application called RFID Fuzzer. If you don't have it, ask Phil for a Flipper.

Go to apps → RFID → RFID Fuzzer → Default Values → TD to 0.6ms → hold over **left hand reader ONLY** → start and let it run

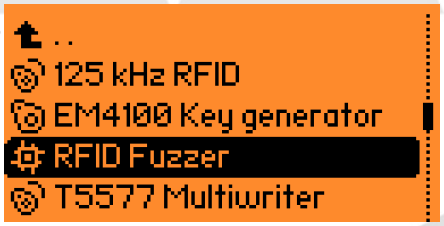
1



2



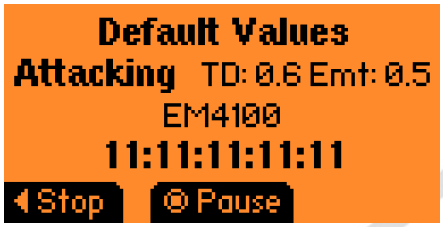
3



4

5

6



Enquiries@StyxSecurity.org

We don't knock.

What's Happening?

RFID Fuzzing

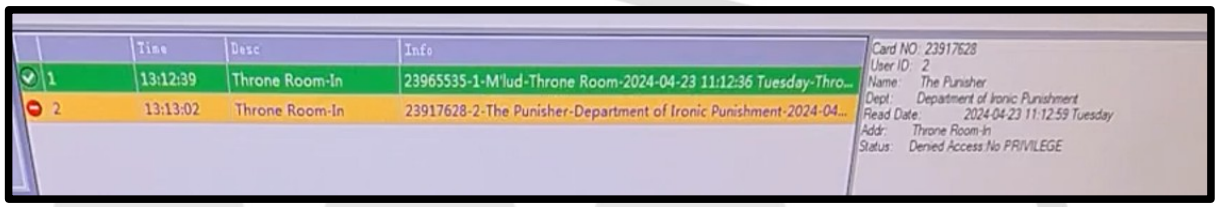
Note that I have replicated an issue found in a different brand of controller. This vulnerability is not found in the EWS controllers used here.

When the successful value is sent to the reader, the back end software sees "1" (not "00000001" as you'd expect from this system).

The door opens, but this card number is not valid. The cards come with 8 digit numbers and, atop that, this number will not appear in the database even when you pull all valid credentials off the reader!

We can also see this reader is attached to door 1. The same method doesn't work on door 2, but changing the card number to "2" will work on door 2. So this is something inbuilt but is entirely invisible to the end user.

What's amazing is this back door disappeared when the server side software ended the trial period and I fully registered it…



Above: example of a real card being presented. The Punisher is trying to access the Throne Room.

Above: real time monitoring during the fuzzing. You can see the card number is 1 with no user ID (impossible to manually enter), name or department.

Enquiries@StyxSecurity.org

We don't knock.

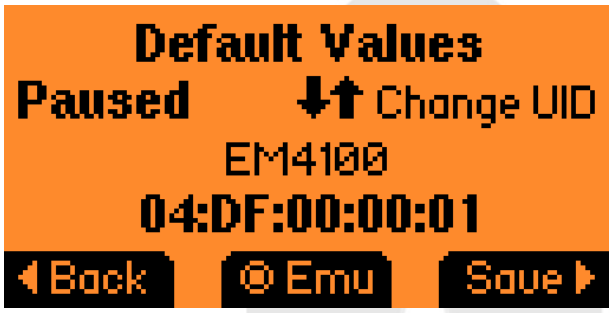
Vexing Hexing

RFID Fuzzing

Note that I have replicated an issue found in a different brand of controller. This vulnerability is not found in the EWS controllers used here.

How to take the successful fuzzing exploit and use it to open the other door:

If we stop the fuzzing attack when the door opens, it'll land us on the hex data that was sent. In case you missed it, it's the one to the right:



Now we have figured out that user number "1" works on door 1, we can create a hex value on an EM4100 card for door 2.

This one is a partial guide, so you can figure the details out for yourself.

The hex data for user 1 was "**04:DF:00:00:01**"

You can probably figure out what you need to change that to for user number 2.

We also note that the card type is **EM4100**.

Steps:

1) You'll need to go into the RFID app.
2) Create a new card / tag.
3) The card type will be EM4100.
4) Set the hex data to the value you've worked out will be correct.
5) Save it.
6) Find the tag you saved and emulate it.
7) Present it to the LF readers and see what happens.

If you're struggling, ask Phil for help.