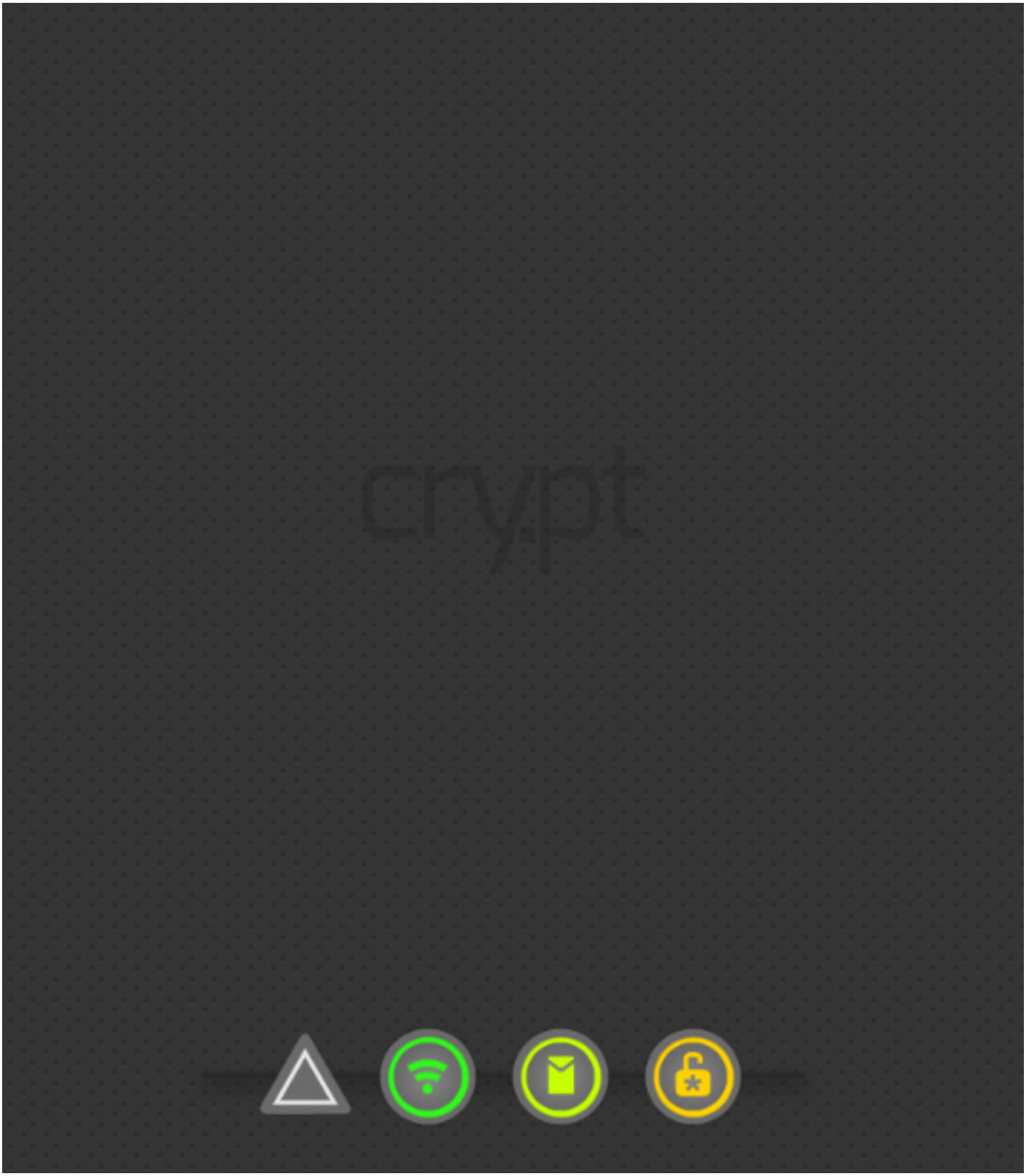
Crypt

User Test Document


crypt

The User Test

The user test started with the Crypt “desktop” open. The testers were presented with two tasks they had to complete:

1. Accept a mission.
2. Complete the mission.

The reason for the general nature of these tasks is to make sure they don’t lead the testers in any specific directions, and to see how little instructions could be given to users in the final tutorial.

It was explained to the testers that their questions would not be answered during the completion of these tasks. Throughout the test, testers were encouraged to share their thought process and troubles. They were asked questions when it seemed they were stuck, or displayed unexpected behavior.

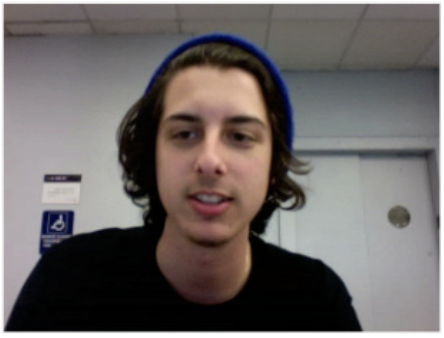
Further questions were asked after completion of the tasks (or after giving up).

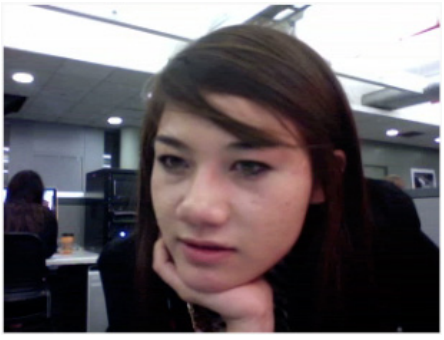
After this, users were shown a video of their test session and asked more questions about their behavior at certain points.

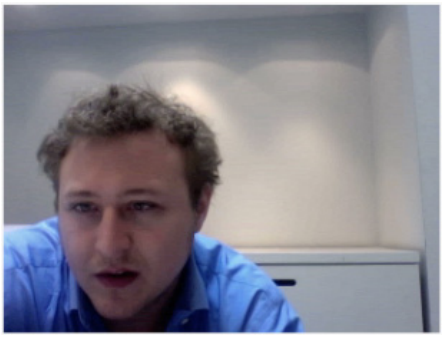
Tests were recorded using *Silverback* software.

The Users

Users were picked out of the game's target audience, in a spectrum between moderately technical and highly technical users. All users were between 19-30 years of age. 10 users were tested.







Moderately technical

5

Highly technical

5

Failed to complete mission

4

Completed mission

6

Fastest completion

1:50

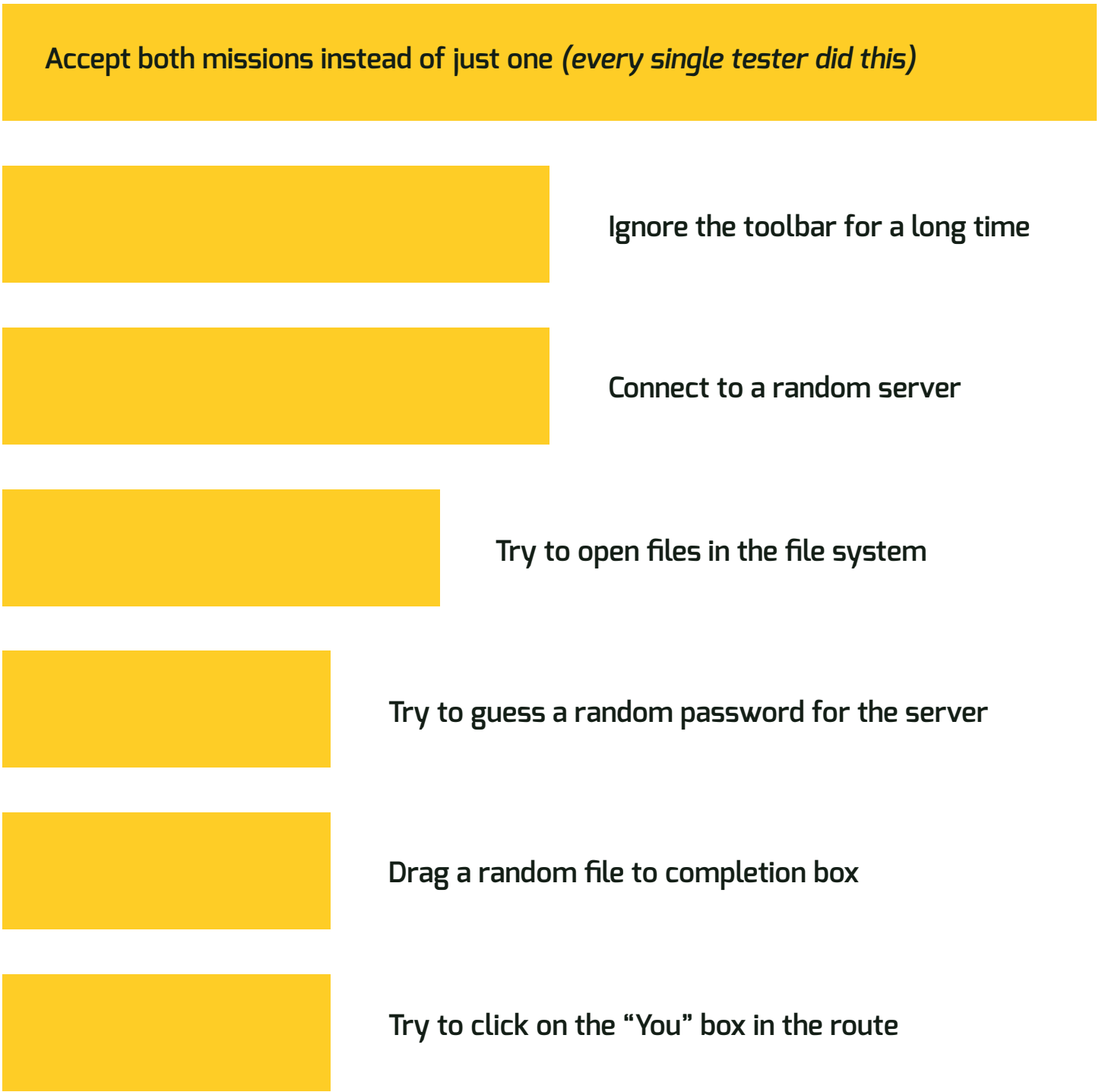
Slowest completion

9:55

Average completion

4:33

Top Issues (by number of users)



Other Issues (by number of users)

- Didn't know what the server nodes were in the map view.
- Couldn't tell the individual mission tabs were tabs (two testers tried to drag them into the "Drag file here" box.
- Tried to click on the white server node (denoting your own connection).
- Didn't read instructions.
- Tried dragging password cracker lock icon to the file box.
- Didn't understand the "Cracker" nomenclature for the password cracker.

One Offs (single-user issues)

- Expected keyboard input, tried copy-pasting information (this was a highly technical tester.)
- Didn't realize more than one window could be open at once, kept closing windows before opening other ones (used to this sort of interaction through other games).
- Randomly chained servers in map view.

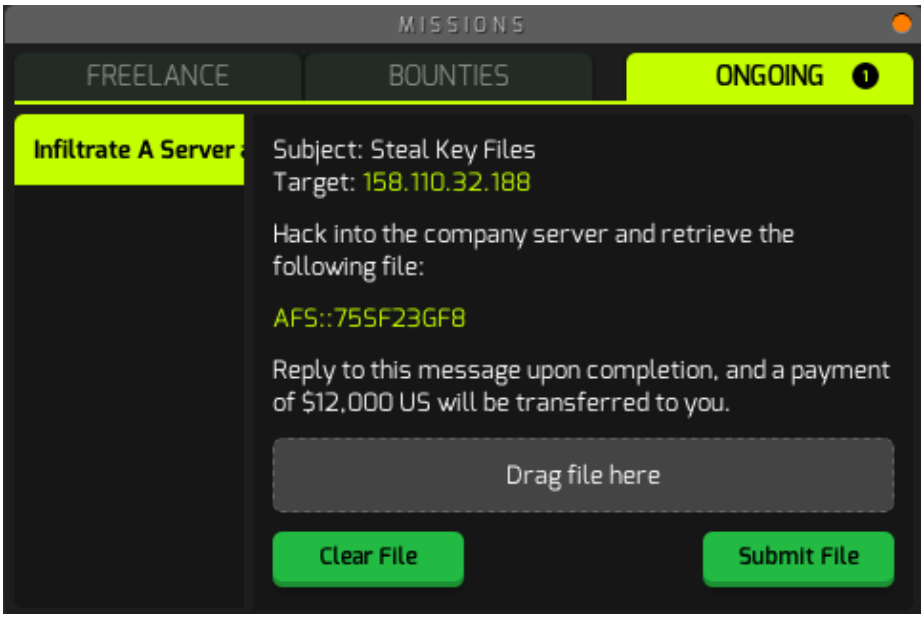
These issues may be larger than they seem, the test did not include enough testers to be conclusive regarding which issues are critical and which aren't.

Insights and Solutions

People don't read, even when it's not much

Perhaps the most saddening insight from this test was that although the mission text has been kept at less than 30 words, few of the testers took the time to understand what was being said. Some didn't even read it. This lead to many random attempts at solving the task, such as dragging random elements into the file box and connecting to random servers.

This can potentially be fixed with an addition of more white-space between the mission text and its surrounding interface. Another cause for testers skipping the text is the highlighted information, which many felt was all they needed to complete the mission. Removing the highlighted text on the first mission for learning and return-

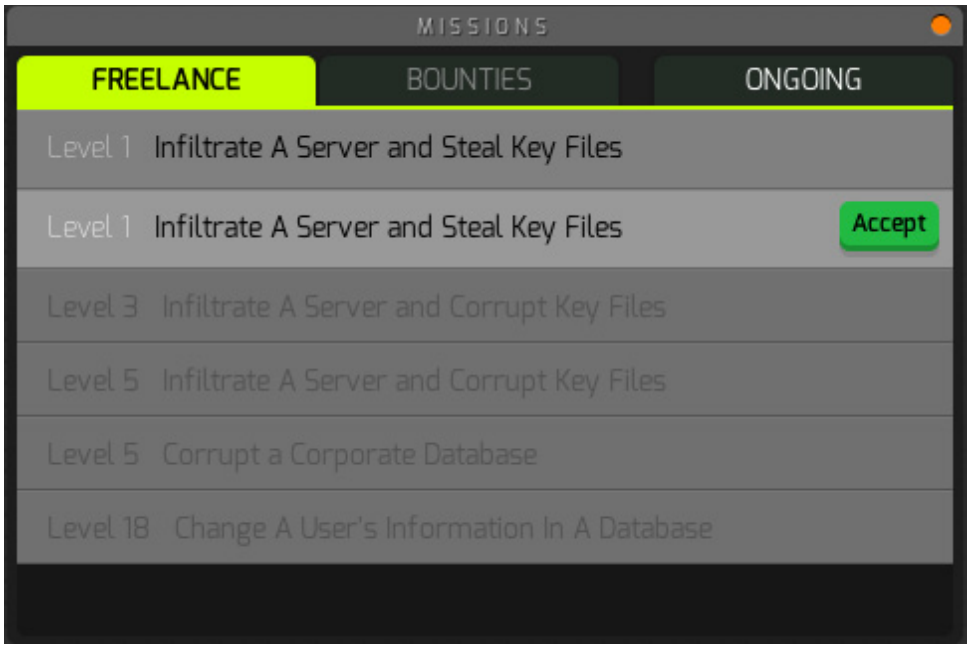


People will take any possible action in an attempt to complete a task

Every single tester accepted both missions, when their task was only to accept one and complete it. They did this because once they clicked “Accept” another accept button was still available. Only when they ran out of accept buttons did they start looking for where their missions went (the ongoing tab, which everyone found relatively quickly).

The second mission, especially with its identical title (the inner information such as target server and file were different,) caused confusion. Users jumped between the tabs and couldn’t figure out what to do.

The best solution is to create a constraint and allow users to accept a single tutorial mission instead of several.



People don't take into account what tools they have at their disposal

Although users first get to the missions through the icons at the bottom of the screen, many forget about these quickly. For instance, when accepting a mission, users spend a long time looking around the mission window before remembering that other windows exist. Two of the users who couldn't complete the mission failed because they didn't notice the password cracker. This also lead many of those who did complete the mission to attempt several guesses at the password before attempting to find another solution.

An appropriate solution would be some kind of indicator that the user should check to see if there is an appropriate "program" for the task. This could be a tip pop-up or simply an arrow.

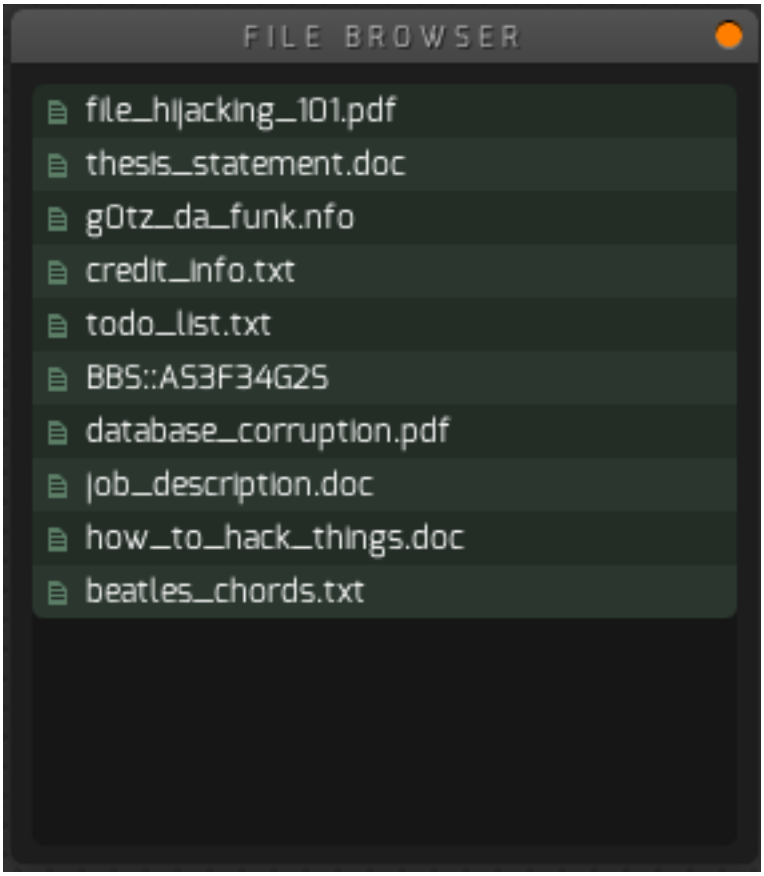
People try to click everything that resembles a button

This issue is not as critical as other issues, since users quickly understand what is clickable and what isn't. That said, further rounding button corners may help to set them apart from other elements that have rounded corners.

Once the operating system model is in place, it's fully in place

Many users attempted to double-click files in the file-system several times before realizing that these were unclickable. Although this isn't a critical issue, creating a pop-up that confirms this would save time for the user. The popup could fade out automatically so as not to require confirmation from the user.

That said, many users clicked on "how_to_hack_things.doc" - a randomly named placeholder. Perhaps by removing the placeholders and creating text files that can be opened, users will have a more detailed step-by-step tutorial if they can't understand the default one. "Readme" files can be provided with any new type of mission, regardless of how intuitive they are, thereby creating several levels of control.

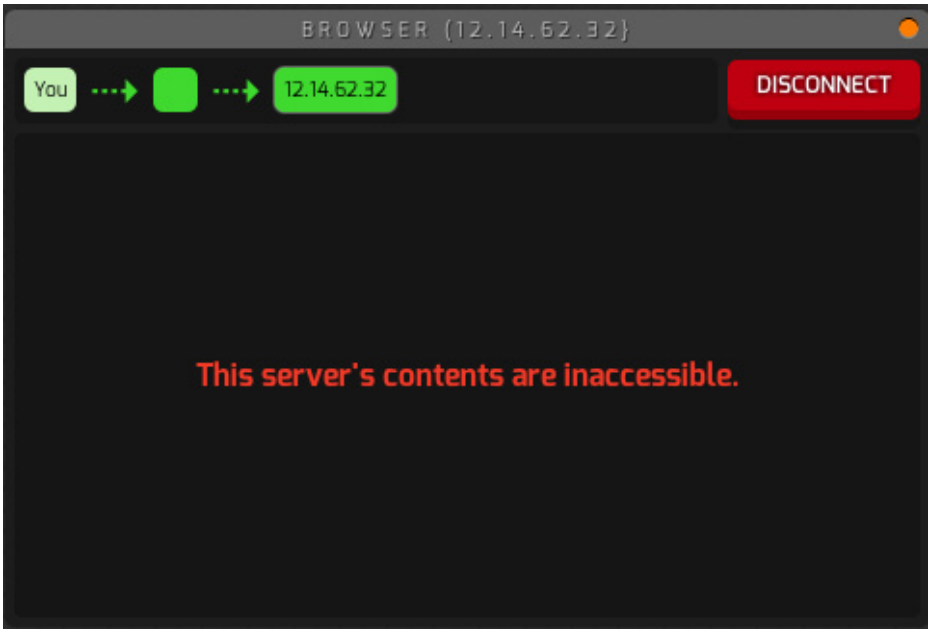


People try entirely random things in order to complete tasks

As an extension of not reading and taking every possible action, people will try anything in an attempt to get a mission done. This includes dragging random files from the file system, connecting to random servers and inserting random passwords.

Currently, dragging the wrong file only yields an error message saying “wrong file.” Inserting the wrong password yields “wrong password.” Connecting to an off-mission server yields a constraining “This server’s contents are inaccessible.”

Perhaps more descriptive error messages may help acquaint the user with the system during the first mission. For instance, having the incorrect server say “try connecting to the server IP mentioned in your mission description.” This will corrode the immersion somewhat, but since it’s only during the tutorial, it may be worth it.



Conclusion

The results were surprisingly positive. Without any explanation whatsoever (other than the word “mission”), 60 percent of testers were able to complete the first mission. Some even enjoyed the puzzle. With the addition of the lessons learned from this test, that number will be much higher, and after further testing, 100 percent of players will be able to understanding the game mechanics.

This also somewhat proves that a tutorial does not need to be a step-by-step guide, and can ask a little more of the user. Taking the time to figure things out on your own has its benefits.

Crypt, like every project during this stage, has a long way to go, and these findings will help it get there.