Poster: Project Leine - A Virtualized Study Infrastructure

Abstract—**Conducting human subjects experiments is a crucial aspect of usable security and privacy research. These human subjects studies are contingent of the participation of end-users, administrators, security workers, developers, briefly: all actors involved in shaping computer security and privacy. However, the recruitment of large and diverse samples for complex experiments can be challenging. Running online studies on platforms such as Amazon Mechanical Turk makes the recruitment of large samples easy but is limited to end-users and relatively simple experiment scenarios, e.g., online surveys. Laboratory studies allow more complex experiments and the recruitment of different types of participants, e.g., software developers but are limited to geographically restricted samples. Additionally, laboratory studies can be time-intensive and expensive, and invite laboratory bias.**

In this poster, we present work in progress to overcome the above limitations. We illustrate and discuss the design and implementation details of a virtualized experiment infrastructure that allows a similar complexity and internal validity as laboratory studies and the recruitment of large and diverse samples of different types of participants. The infrastructure provides an entirely virtual desktop environment using state of the art JavaScript based VNC technology in a browser. Participants only need a modern browser to access the infrastructure. Researchers can configure the virtual environment according to their requirements, including the installation of software or collecting telemetry data.

While still work in progress, the poster illustrates key contributions of our work and future research enabled by the project.

I. Introduction

Recruiting appropriate samples of participants for human subjects experiments in usable security and privacy research can be challenging. Laboratory studies allow researchers to conduct complex experiments: Researchers can control many variables and ask participants to test different applications or work on complex tasks such as solving programming challenges. The possibilities of laboratory experiments in usable security and privacy research comes with a substantial limitation: Participants are usually restricted to the geographic area close to the researchers, and can be biased in their performance due to being placed in an unfamiliar, observed environment. While this is limiting for experiments with end-users, for highly specialized participant populations, the recruitment restriction can be prohibitive of successful experiments. Researchers might not be able to conduct a usable security experiment with developers simply because they cannot recruit enough participants locally. In contrast, online platforms such as Amazon MTurk allow to recruit large and diverse samples. However, they are mostly limited to experiments with end-user participants and do not provide much flexibility concerning the experimental setup. While it is easy to run online surveys on MTurk with end-users, the platform cannot easily be used for security application experiments with different groups of participants.

We present work in progress that tries to overcome a central challenge in human subject experiments in security and privacy research reported in previous studies [1]–[3]: A convenient study infrastructure that allows to recruit large and diverse samples of participants for experiments that require complex setups. *Project Leine* is a fully virtualized remotely controllable study infrastructure that combines the high flexibility of laboratory studies in terms of experimental setups and the ability of recruiting geographically diverse sets of participants known from platforms such as Amazon MTurk. Researchers can flexibly setup virtualized desktop environments that can be used by participants using state of the art JavaScript VNC technology and a modern browser.

II. Virtualized Experiment Infrastructure

Project Leine is built on top of a virtualized infrastructure consisting of Kubernetes and Docker containers. It easily scales to hundreds of concurrent participants, can be flexibly configured by researchers and easily accessed by participants.

A. Researchers

Researchers can easily setup new Docker containers using configuration scripts and store them as images for use during an experiment. They can freely pick an operating system and whatever applications they need for a study.

Building of the study environment containers is directly handled by the docker engine, which allows for explicit control of included software and installed versions. This enables researchers to A/B test different configurations or software versions within the same study. Using study containers provided by researchers, the infrastructure is able to independently balance running study environments in different geographical locations between participants to optimize connection speeds and scale available environments up or down based on the number of currently connected participants. In addition, the study infrastructure is able to autonomously handle some of the tedious tasks during managing a study such as automatically randomizing tasks based on different assignment methods and redirecting participants to an exit survey. For security reasons, each participants gets a different virtual machine assigned. This makes sure participants' action are isolated from each other.

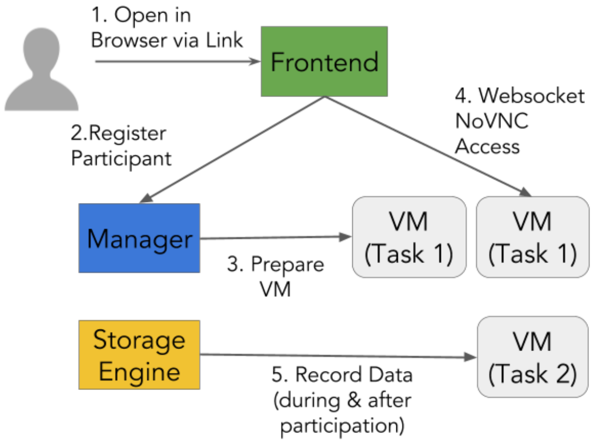
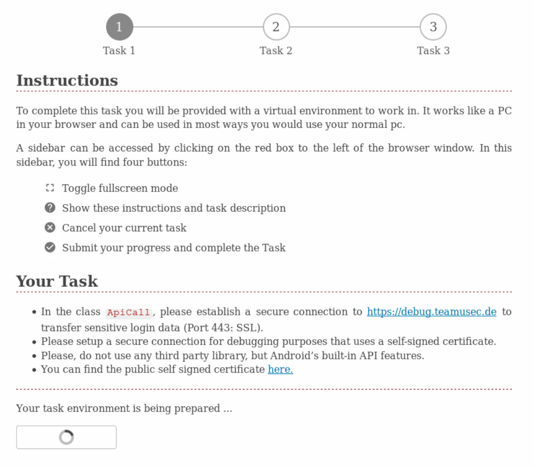

1. Open in Browser via Link
Frontend
2.Register Participant
4. Websocket NoVNC Access
Manager
3. Prepare VM
VM (Task 1)
VM (Task 1)
Storage Engine
5. Record Data (during & after participation)
VM (Task 2)

Fig. 1: Individual components of the study infrastructure



Instructions

To complete this task you will be provided with a virtual environment to work in. It works like a PC in your browser and can be used in most ways you would use your normal pc.

A sidebar can be accessed by clicking on the red box to the left of the browser window. In this sidebar, you will find four buttons:

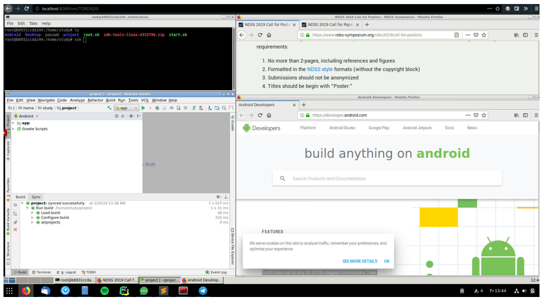
- ⛶ Toggle fullscreen mode
- ? Show these instructions and task description
- ☒ Cancel your current task
- ✓ Submit your progress and complete the Task

Your Task

- In the class `ApiCall`, please establish a secure connection to https://debug.teamusec.de to transfer sensitive login data (Port 443: SSL).
- Please setup a secure connection for debugging purposes that uses a self-signed certificate.
- Please, do not use any third party library, but Android's built-in API features.
- You can find the public self signed certificate here.

Your task environment is being prepared ...

Fig. 2: The loading transition for each task. It shows the task description, progress and a guide for the NoVNC-frontend


NDSS 2019 Call for Posters - NDSS Symposium - Mozilla Firefox
NDSS 2019 Call for Poste
NDSS 2019 Call for Paper
https://www.ndss-symposium.org/ndss2019/call-for-posters/
Android Developers - Mozilla Firefox
Android Developers
https://developer.android.com
Developers
Platform
Android Studio
Google Play
Android Jetpack
Docs
News
build anything on android
Search Products and Documentation
FEATURED
SEE MORE DETAILS
OK
Event Log
Fr 13:44

requirements:

1. No more than 2-pages, including references and figures
2. Formatted in the NDSS style formats (without the copyright block)
3. Submissions should not be anonymized
4. Titles should be begin with "Poster:"

We serve cookies on this site to analyse traffic, remember your preferences, and optimise your experience.

Fig. 3: The NoVNC client, showing Android Studio, Firefox and a shell with root access running on an LXDE desktop inside the browser

B. Participants

Participants can easily join a study conducted with *Project Leine*. With a modern browser, they can work on study tasks using a JavaScript based VNC client [1]. There is no need to download or install anything. After confirming a consent form, participants are redirected to the study description and a brief introduction to the study environment. Participants are encouraged to toggle the browser window to full screen mode, can re-read the study and task description on a mouse click, jump to another task or finish the entire study. During or after a study, the participant can be in the same browser tab seamlessly directed to surveys or additional text pages required by the study setup. Working on study tasks is as working with a regular desktop environment. Researchers can previously configure the applications participants are allowed to use, e.g., a setup for a programming task study could include a bash shell, an IDE, and a browser.

The study infrastructure captures participants' bash and browser history, tracks changes of modified source code files and copy and paste behaviour of participants. Participants might be asked to fill out short in-situ questionnaires and a more in-depth follow up survey after they completed all study tasks.

III. Future Research Enabled by Project Leine

Project Leine enables researchers to combine the best of online and laboratory studies: Participants with strong technical skills can remotely participate in complex studies with setups and observation techniques that resemble laboratory studies, but come without the drawback of being restricted to local, in-lab participation. With the help of *Project Leine*, researchers can run controlled experiments with human subjects to test software features such as private browsing modes of modern browsers using a niche group of participants that are hard to recruit for laboratory studies. Additionally, it can enable programming experiments that require certain compilers or integrated development environments. With *Project Leine*, we enable a more diverse set of researchers (e.g., those not local to highly skilled developer communities) to conduct experiments that were previously restricted to laboratories. *Project Leine* eases the overhead of local recruitment and makes research on tech populations more inclusive to all researchers, as well as making experiments easier to conduct, which can contribute to a better chance of having experiments replicated, and experiments being run with more diverse populations in the first place.

References

[1] Y. Acar, M. Backes, S. Fahl, S. Garfinkel, D. Kim, M. L. Mazurek, and C. Stransky, "Comparing the Usability of Cryptographic APIs," in *Proc. 38th IEEE Symposium on Security and Privacy (SP'17)*. IEEE, 2017.

[2] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky, "You Get Where You're Looking For: The Impact of Information Sources on Code Security," in *Proc. 37th IEEE Symposium on Security and Privacy (SP'16)*. IEEE, 2016.

[3] A. Naiakshina, A. Danilova, C. Tiefenau, M. Herzog, S. Dechand, and M. Smith, "Why Do Developers Get Password Storage Wrong?: A Qualitative Usability Study," in *Proc. 24th ACM Conference on Computer and Communication Security (CCS'17)*. ACM, 2017.

[1] cf.https://novnc.com/

A Virtualized Study Infrastructure

Dominik Wermke, Nicolas Huaman, Christian Stransky, Yasemin Acar, Sascha Fahl

Goals

- Browser accessible remote study environment
- Integrated study tools (consent forms, tokens, follow up surveys etc.)
- Flexibility for researchers and participants:

  Researcher: Provide the same easily configurable virtualized working environment for all participants.

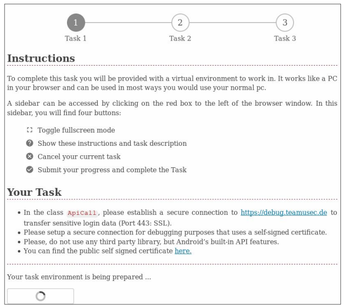
  Participant: No need to download anything; only a (modern) browser required.

Approach

- Each study environment deployed in separate docker container.
- Backend/frontend also managed as docker containers; allows easy deployment

  Input:
  - Task/condition descriptions
  - Consent form and other metadata
  - Docker Images containing virtual study setup

  Output:
  - Automatic token/participant management
  - Embedded task description, consent form, study description and follow up
  - Recording of properties defined in Task Description



Instructions

To complete this task you will be provided with a virtual environment to work in. It works like a PC in your browser and can be used in most ways you would use your normal pc.

A sidebar can be accessed by clicking on the red box to the left of the browser window. In this sidebar, you will find four buttons:

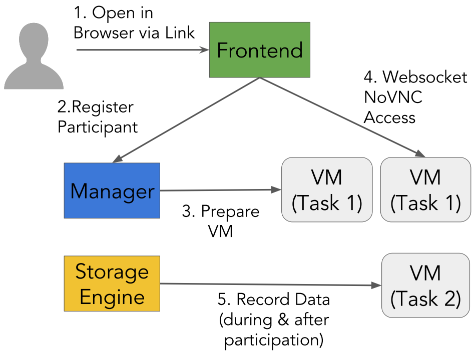
- Toggle fullscreen mode
- Show these instructions and task description
- Cancel your current task
- Submit your progress and complete the Task

Your Task

- In the class `ApiCall`, please establish a secure connection to https://debug.teamusec.de to transfer sensitive login data (Port 443: SSL).
- Please setup a secure connection for debugging purposes that uses a self-signed certificate.
- Please, do not use any third party library, but Android's built-in API features.
- You can find the public self signed certificate here.

Your task environment is being prepared ...

Fig 2: Taskdescription and -progress, environment description



1. Open in Browser via Link

Frontend

2.Register Participant

4. Websocket NoVNC Access

Manager

3. Prepare VM

VM (Task 1)

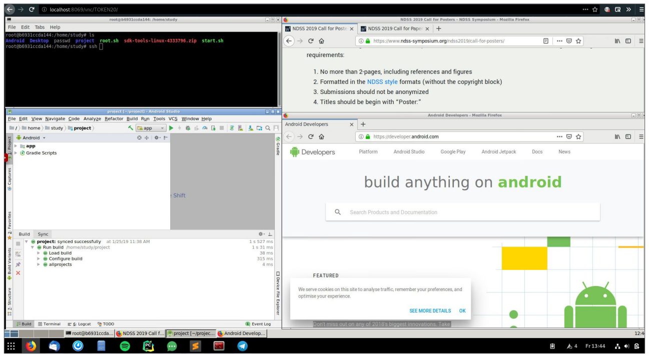
VM (Task 1)

Storage Engine

5. Record Data (during & after participation)

VM (Task 2)

Fig 3: Infrastructure Diagram



```
root@b6931ccda144:/home/study# ls
Android  Desktop  passwd  project  root.sh  sdk-tools-linux-4333796.zip  start.sh
root@b6931ccda144:/home/study# ssh
```

requirements:

1. No more than 2-pages, including references and figures
2. Formatted in the NDSS style formats (without the copyright block)
3. Submissions should not be anonymized
4. Titles should be begin with "Poster:"

```
project: synced successfully   at 1/25/19 11:38 AM   1 s 527 ms
  Run build /home/study/project                      1 s 31 ms
    Load build                                          38 ms
    Configure build                                    315 ms
    allprojects                                          4 ms
```

We serve cookies on this site to analyse traffic, remember your preferences, and optimise your experience.

Don't miss out on any of 2018's biggest innovations. Take

Fig 1: Study Environment: LXDE running Android Studio, Firefox and a Terminal inside a Firefox Browser

Features:

- Custom SSL Certificates for servers via customized Firefox without noticeable difference
- Provide any application/plugin without annoying installation/update process
- Track things like: clipboard content, selections, browser history, screen recording etc.
- Platform independent (runs on Android)
- Docker setup -> studyenvs are reusable and fixed in setup

Limitations:

- Docker setup -> complicated for smaller/simpler studies
- Larger infrastructure required (Docker and/or Kubernetes, multiple VMs)
- Browser environment might skew the results

Current status:

- Fully working prototype
- Currently working on remote deployment prototype using Kubernetes

Future:

Testing infrastructure in real world studies

Extending configuration options