

Security in Action: Demo and Tools

IEEE IoT Seasonal School

Muhammad **Rizwan** Asghar

8 July 2024

 Follow @DrRizwanAsghar

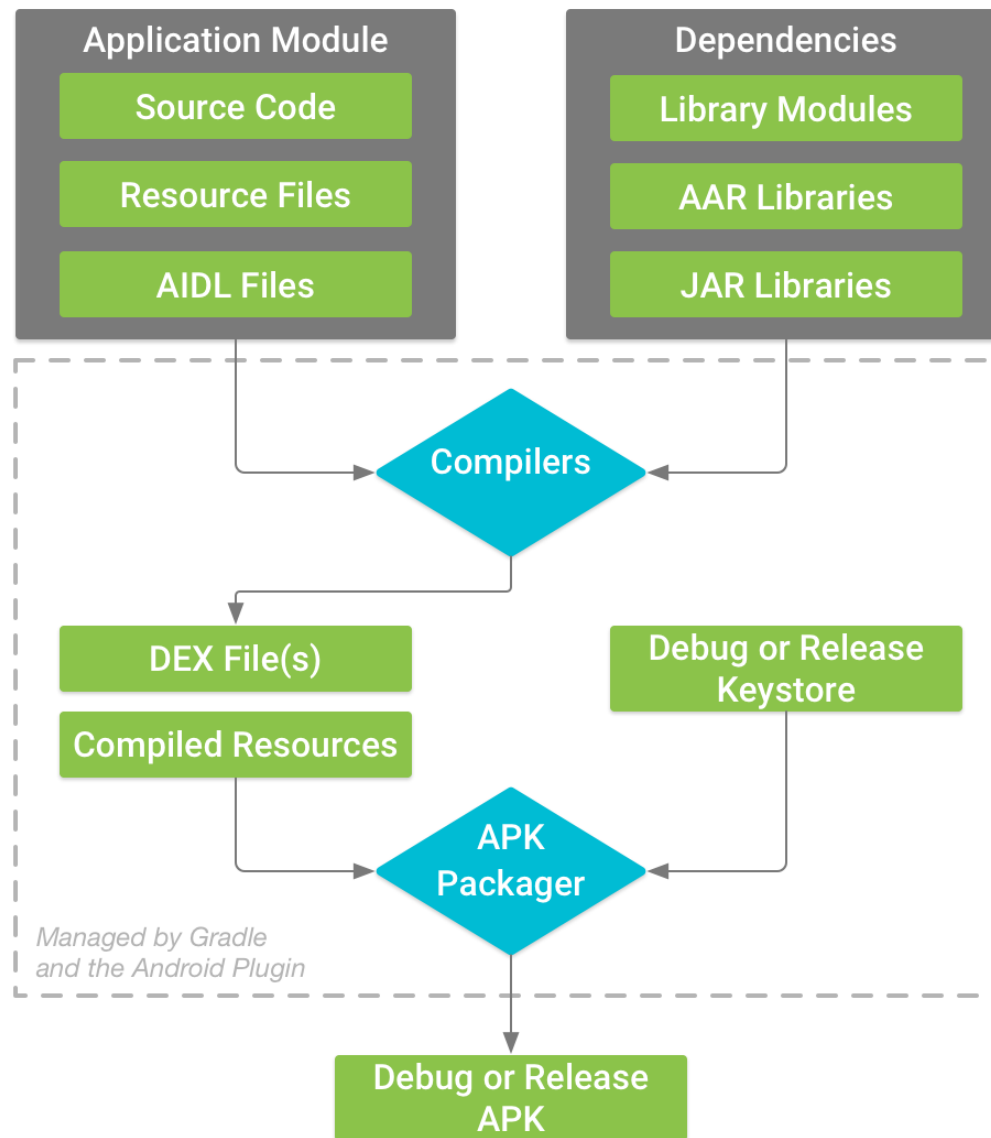


Key Focus



- App building and decoding
- Code obfuscation and reverse engineering
- A case study and results
- Some exercises

Android Build Process

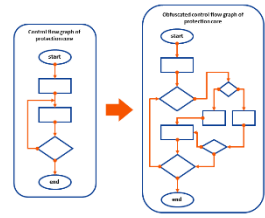


Decoding APK



- Download [Hello World Android app](#)
- Rename apk to zip
- Extract classes.dex
- Download [dex2jar](#)
- Download [java decompiler](#)

Software Obfuscation



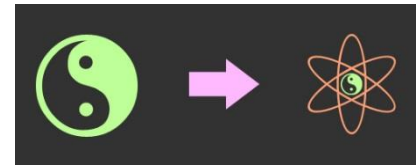
- Software **obfuscation** is a promising technique to protect sensitive information in application code
- The basic idea is to transform code such that it becomes **hard to interpret**
- Software obfuscation makes it difficult for attackers to extract sensitive information from
 - Code consisting of private **data**
 - E.g., **password** matching
 - **Control flow**
 - E.g., business **logic**

Reverse Engineering



- Sophisticated reverse engineering mechanisms and **tools** have been developed for analysing the code
- One can easily understand the code by using **reverse engineering** tools
- For designing obfuscation methods, it is necessary to **test them against** available reverse engineering tools

Obfuscation Types



- Code obfuscation can be broadly classified into four main categories [Balachandran TIFS13]
 - Layout obfuscation
 - Design obfuscation
 - Data obfuscation
 - Control obfuscation

Layout Obfuscation



- Layout obfuscation refers to obscuring the layout of the program

- Examples
 - Deleting comments
 - Removing debugging information
 - Renaming variables
 - Changing formatting of source code
 - ...

Design Obfuscation



- Design obfuscation refers to obscuring the design of the software system

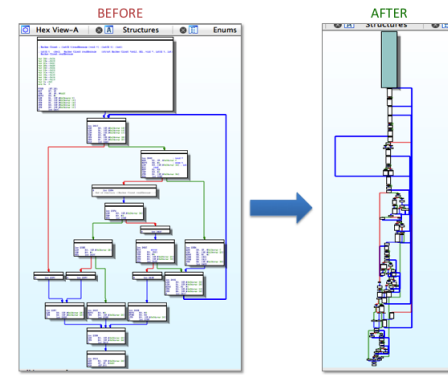
- Examples
 - Splitting classes
 - Merging classes
 - ...

Data Obfuscation

```
public static void main(String[] args) {  
    String first_name = "William";  
    String family_name= "Shakespeare";  
  
    System.out.println( first_name + " " + family_name);  
}
```

- Data obfuscation aims at preventing the adversary from extracting information from the data used in the program
- Examples
 - Data to procedure conversion
 - Encoding (or encryption)
 - Variable splitting
 - Changing lifetime of variables
 - ...

Control Obfuscation



- Control obfuscation obscures the control flow information of the program
- Examples
 - Opaque predicates
 - E.g., “if (1 > 0)”
 - Control flow flattening
 - ...

Group Project Case Study



- **70** students divided into **13 groups**
 - 5-6 students per group
- Development phase includes (9 weeks)
 - A java app (400-1000 lines of code)
 - Novel obfuscation technique and tool
 - **Obfuscated app**
- A **report** at the end of a challenge phase addressing (2 weeks)
 - Related work
 - Obfuscation technique
 - Performance analysis
 - Limitations
 - **Reverse engineering** of apps developed by other groups
- Post-challenge **group presentation** (1 week)
 - 20 minutes presentation and 10 minutes QA

PIN Authentication Example



- Original code:

```
if(input == "1234")  
{  
    //authenticate  
}
```

Data Obfuscation using Hash Function



- A hash function is a cryptographic checksum

- Let's assume:

`hash("1234")="9876"`

- The obfuscated version should be:

```
if(hash(input) == "9876")  
{  
    //authenticate  
}
```

Data Obfuscation using Splitting Variable



- Let's assume
 $v=5$
- We can split **v** into two:
 $a=2$ and $b=3$ and
replace **v** with **$a+b$**
- Likewise, also consider a string
`name="Ronald Rivest"`
- We can split this **`name`** into two:
`FirstName="Ronald"` and `LastName="Rivest"`

Control Flow Example



- Consider the following expression:
$$(a-b)^2 = a^2 + b^2 - 2ab$$
- The expression seems to be true always, but it is not the case
- Values of a and b can be chosen to trigger integer overflow on the right side

Bus Tracker App

11:33

Bus Tracker

Stop Number

8272

SEARCH

FAVOURITE

FIND NEARBY STOPS

11:32

Stop 8272
10 Maioro St

CITY CENTRE		
249	2 min delay	9min
Arriving at 15:40		
CITY CENTRE		
233	3 min delay	18min
Arriving at 15:48		
CITY CENTRE		
249	Scheduled	35min
Arriving at 16:08		
CITY CENTRE		
233	Scheduled	40min
Arriving at 16:13		
CITY CENTRE		
249	Scheduled	65min
Arriving at 16:38		
CITY CENTRE		
233	Scheduled	70min
Arriving at 16:43		
CITY CENTRE		
249	Scheduled	100min
Arriving at 17:13		

Calories Calculator App

Calories Calculator

Your weight

73 kg

Time Spent on the exercise

60 mins

Exercise

Weight Training Walking Jogging

Calories burned

744

Exercise Food

Calories Calculator

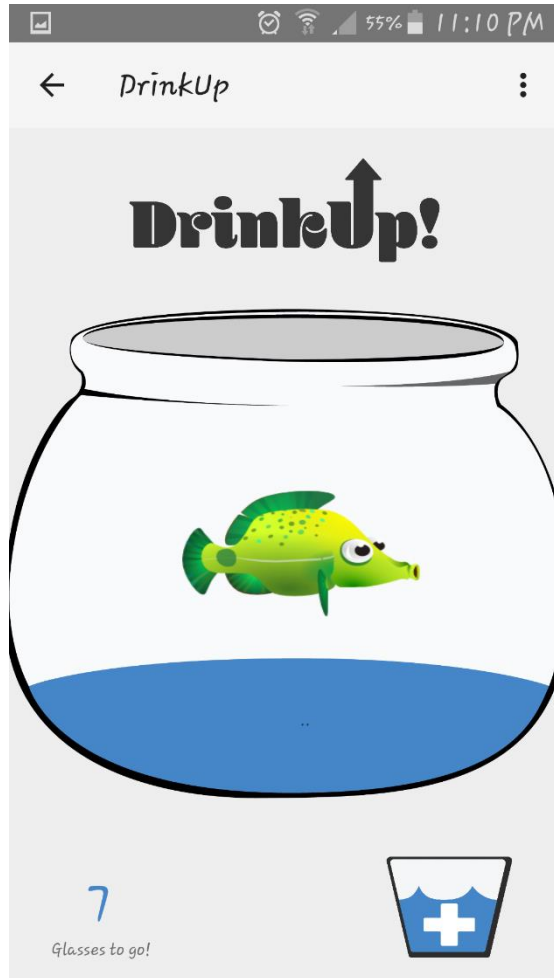
apple

SEARCH

There are 67.0 calories in 100g of apple

Exercise Food

Drinkup App



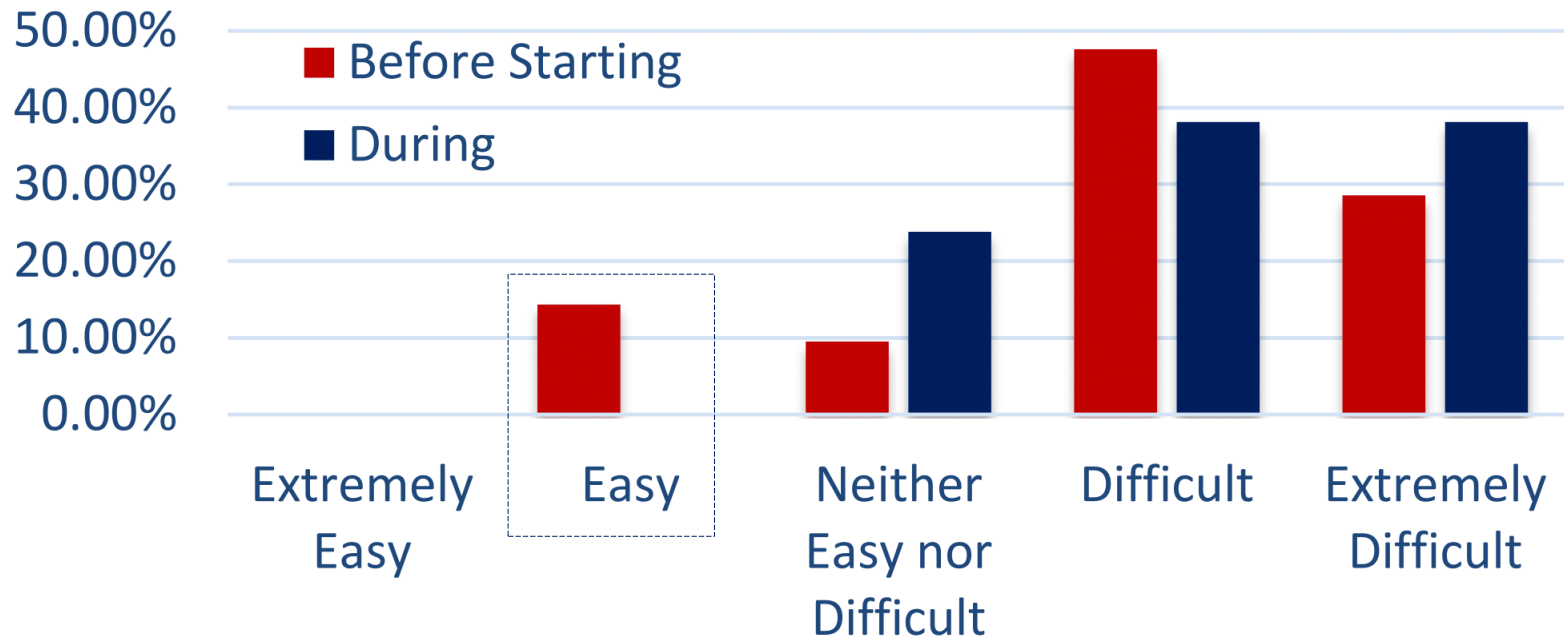
Response Rate



- **Pre-challenge questionnaire**
 - **30%** responded
 - 21 students
- **Post-challenge questionnaire**
 - **40%** responded
 - 28 students
- **Approved** by The University of Auckland Human Participants Ethics Committee
 - Reference number 019274

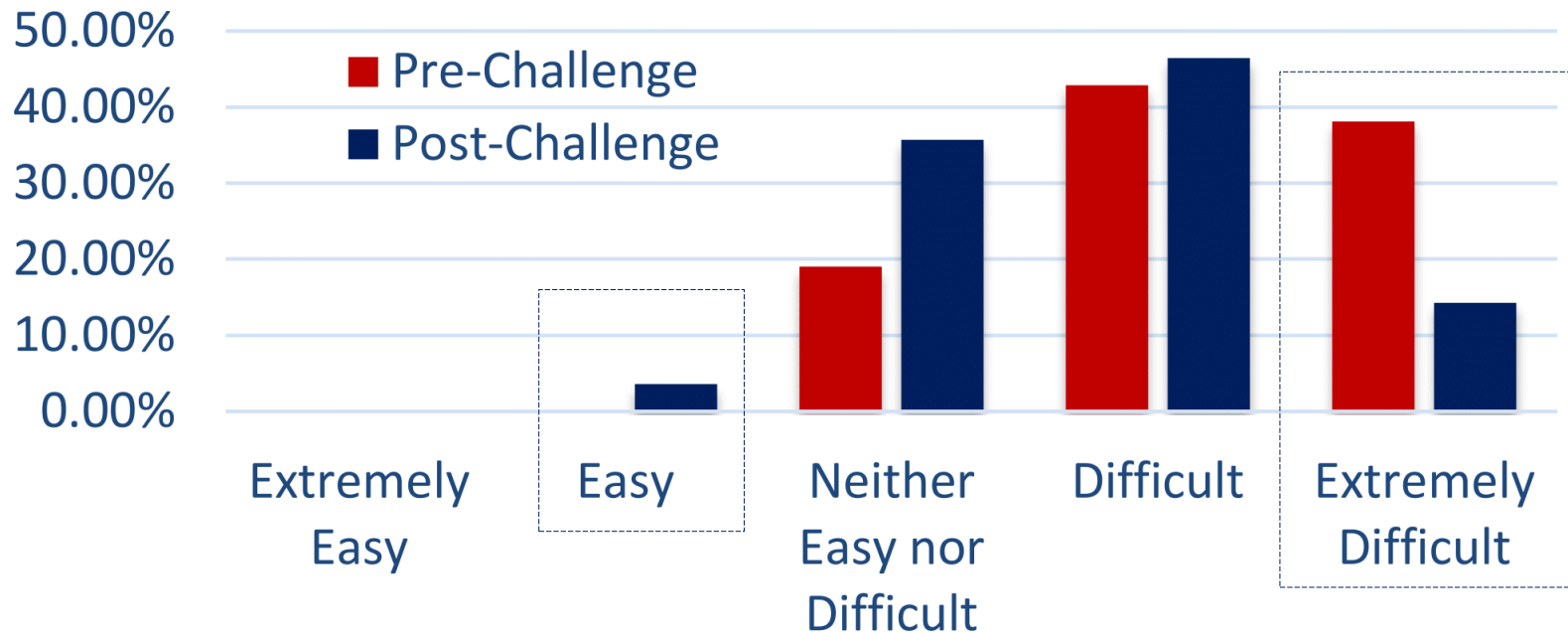
Student Perception of Novel Idea

Student Perception of Coming up with Novel Idea for Obfuscation



Student Perception of Reverse Engineering

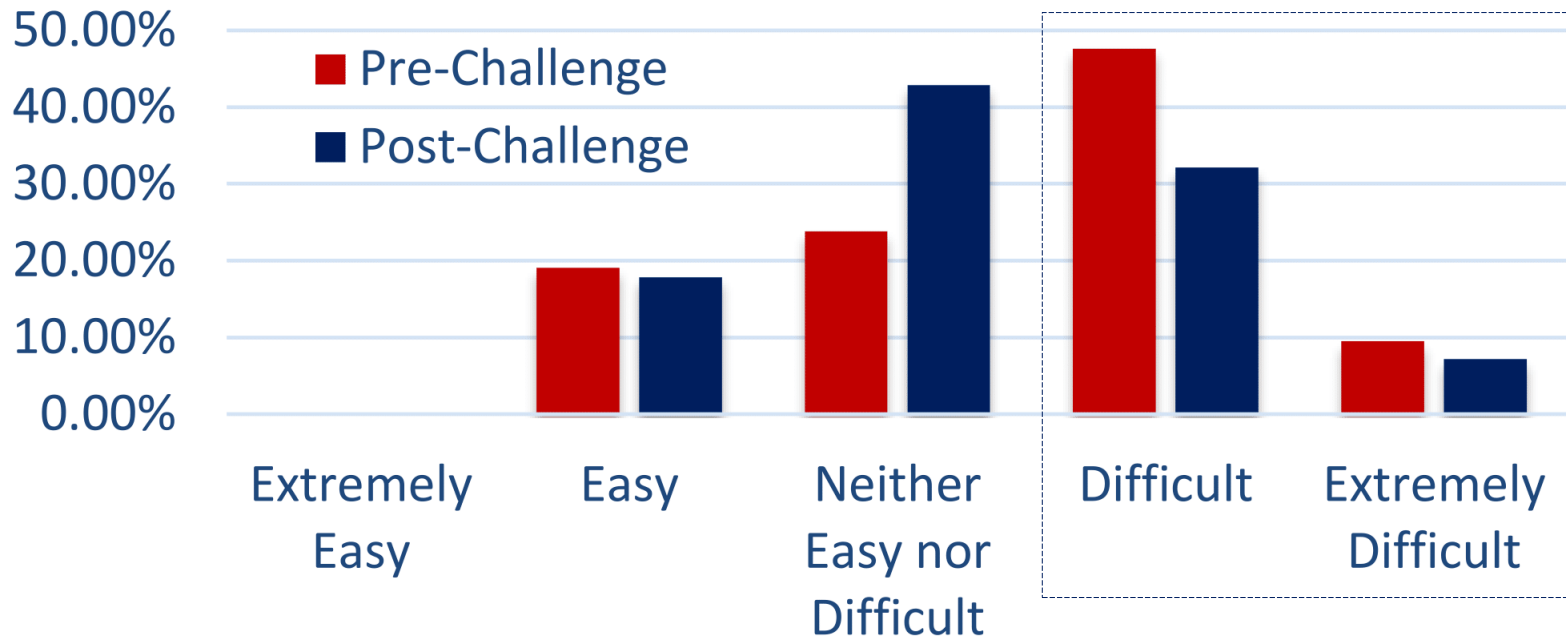
Student Perception of Doing Reverse Engineering



2-tailed Mann-Whitney U test: $\alpha=.05 < p\text{-value}=.05486$

Student Perception of Reverse Engineering by Others

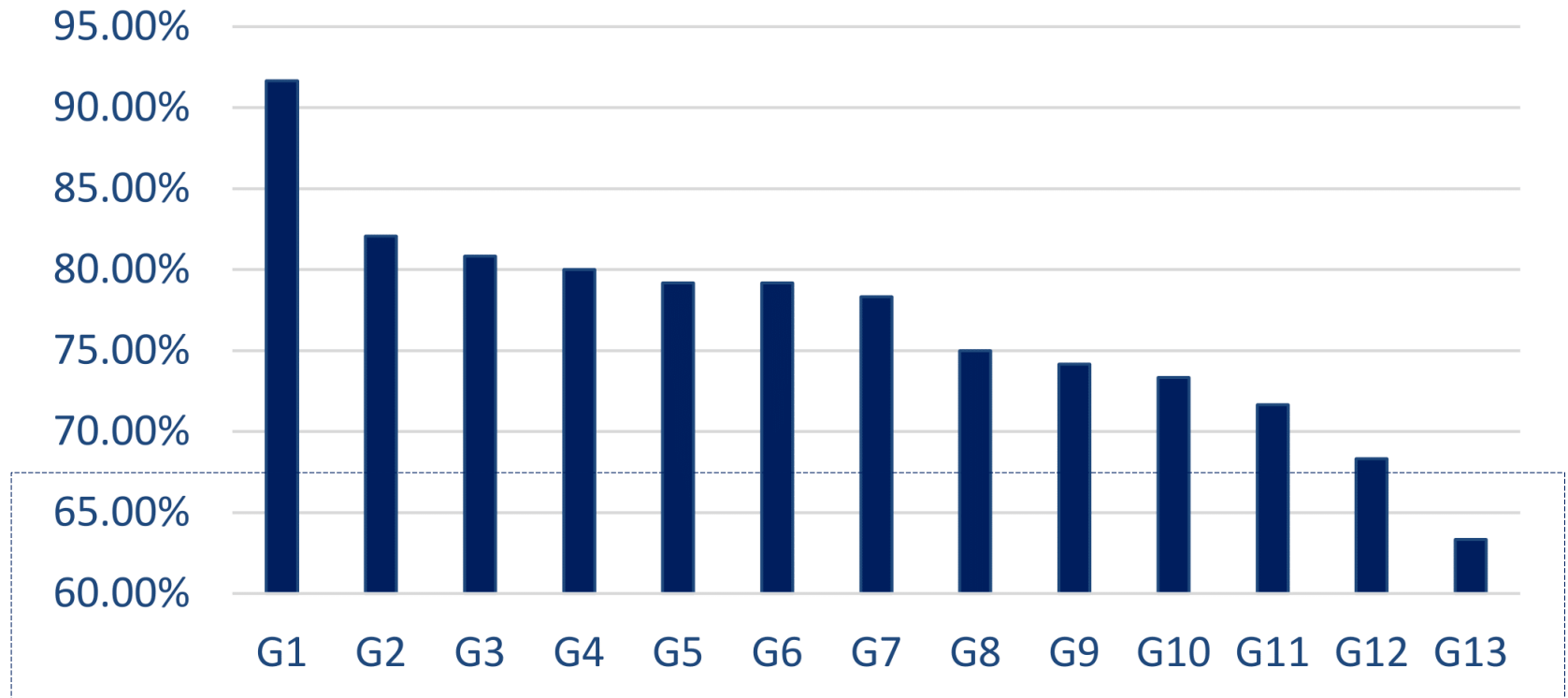
Student Perception of
Reverse Engineering by Other Groups



2-tailed Mann-Whitney U test: **alpha=.05 > p-value=.01596**

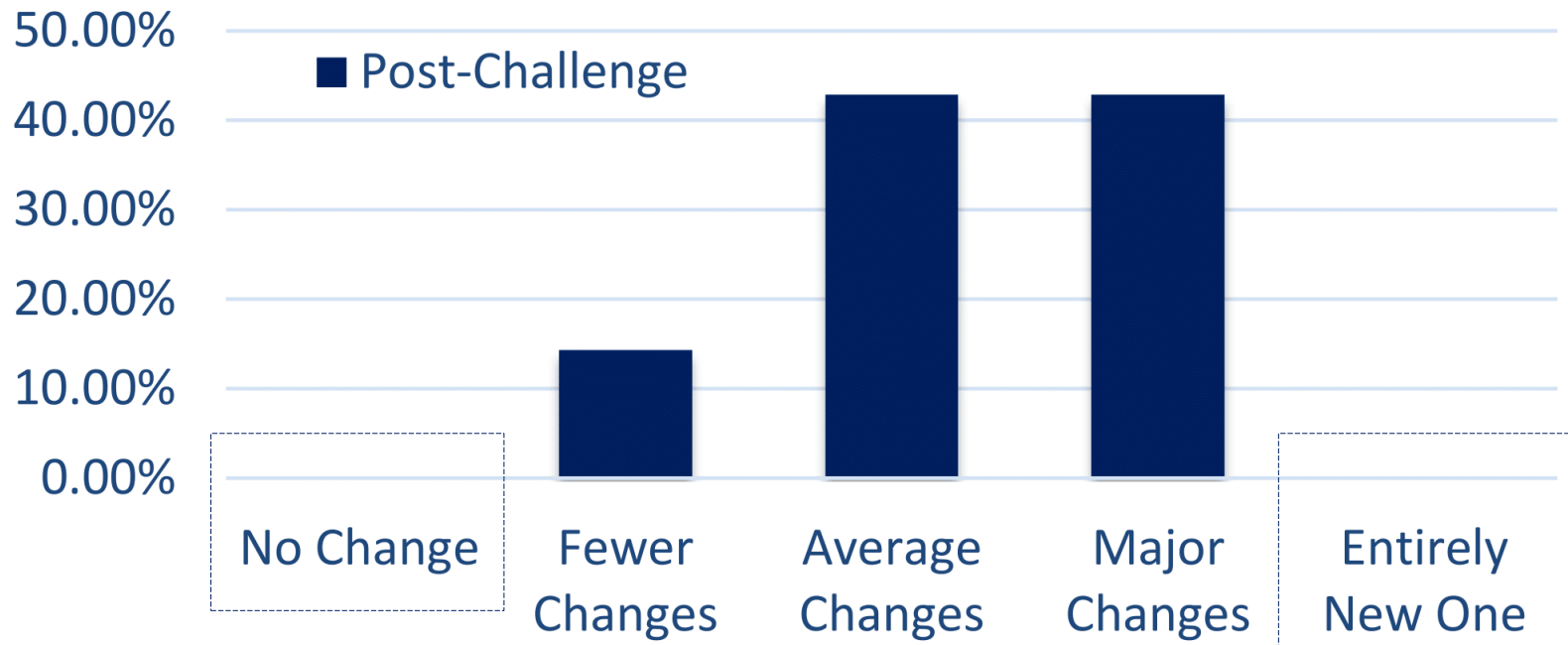
Performance of Groups in Challenge Phase

Status of Reverse Engineering

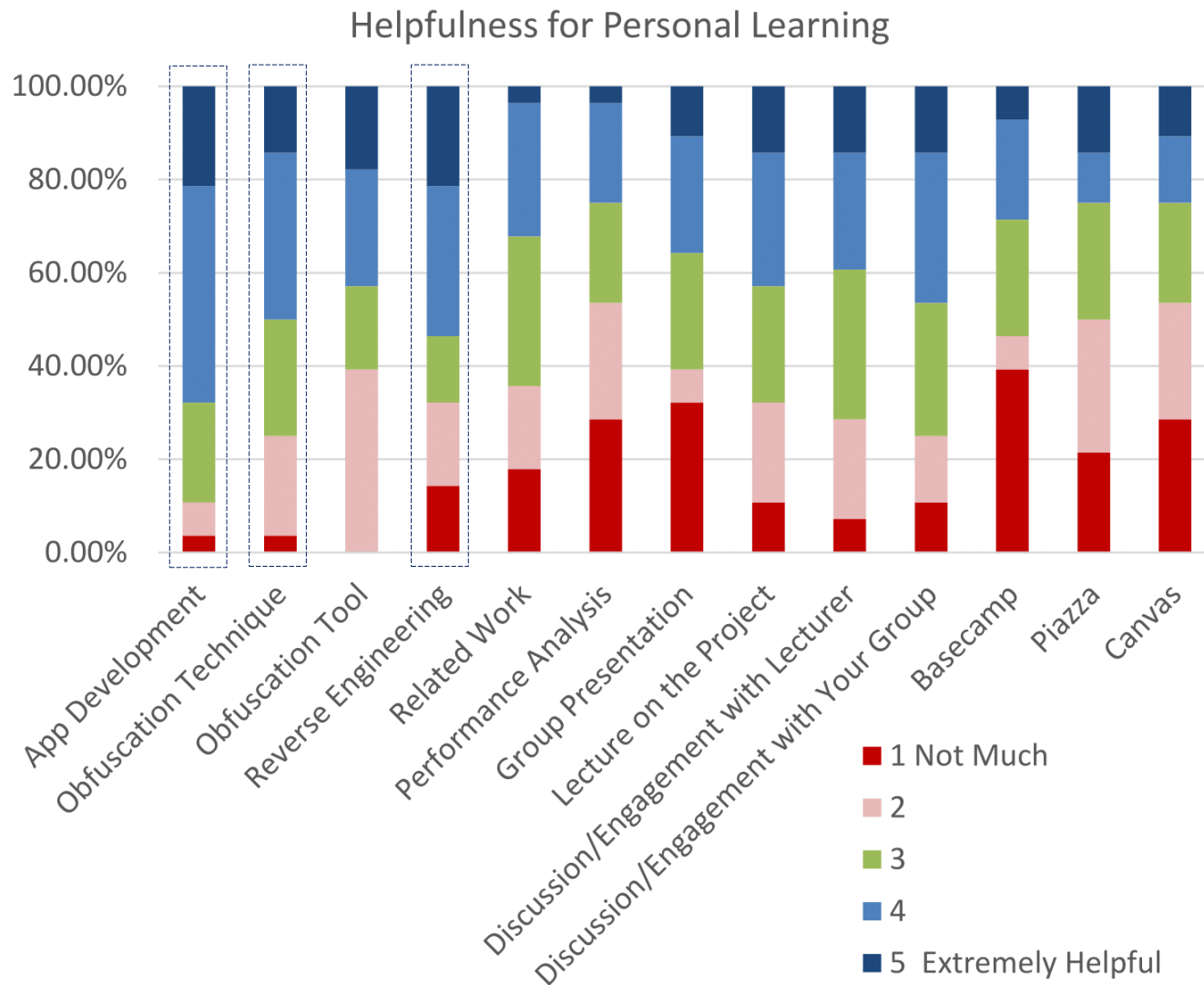


Potential Improvements

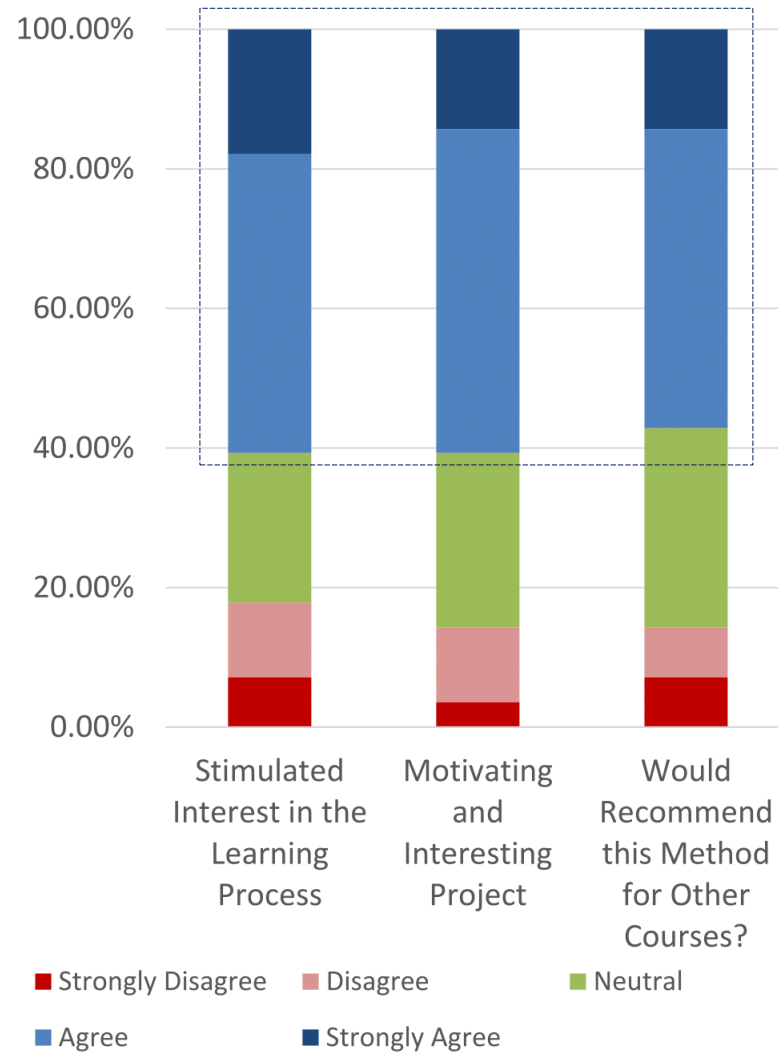
What Would you do to Improve your Obfuscation Technique?



Personal Learning



Learning and Interest



Summary



- A **novel** approach to teaching and learning cyber security
- Based on **competitive** group projects
- A **positive change in student** perception and learning activities
- Students appear to have **more realistic** expectations about the difficulty of effectively obfuscating apps
- **More** administrative work
- A **positive experience** for both students and teaching staff

Exercise on Reverse Engineering



- Reverse engineer [Hello World Android app](#)
- Reverse engineer any other app of your choice
- Observe control flow and data assets

Consider a C Program



```
#include <stdio.h>
void main() {
    char firstname[10];
    char lastname[10] = "changeme";
    printf("Enter your name: ");
    gets(first);
    printf("Your name is: %s\n", last);
}
```

Exercise on Buffer Overflow



- Download the C program and run it by providing input that overwrites the lastname string (i.e., "changeme")
- The program must print your last name
- You can run it using this environment https://www.onlinegdb.com/online_c_compiler

Resources



- [Balachandran TIFS13] Balachandran, Vivek, and Sabu Emmanuel. "Potent and stealthy control flow obfuscation by stack based self-modifying code." IEEE Transactions on Information Forensics and Security (TIFS) 8, no. 4 (2013): 669-681.
- Asghar, Muhammad Rizwan, and Andrew Luxton-Reilly. "Teaching cyber security using competitive software obfuscation and reverse engineering activities." In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, pp. 179-184. 2018.

Resources Cont.



- Apk decompiler
 - <http://www.javadecompilers.com/apk>
- Mobile security wiki
 - <https://appsecwiki.com/#/mobilesecurity>



Questions?

Thanks for your attention!