Finding Vulnerability

Experiment Reproducibility Report - E-2

Security Experiments and Measurements

Vrije Universiteit 2022

Student ID: 2726001
Student Name and Surname: Xingwen Xiao
VU email: x3.xiao@student.vu.nl

Student ID: 2722930
Student Name and Surname: Haohui Zhang
VU email: h17.zhang@student.vu.nl

Plagiarism statement

We hereby confirm that this assignment is our own work, is not copied from any other person's work (published or unpublished) and was not shared with any other person that could copy its contents.

Signatures:

Haohui Zhang

_____

Experiment Description

1. Intervention

At the time of the experiment you didn't know what was the difference between the intervention group and the control group. By the time you write this report you will know. You should describe here briefly what the intervention is and why this is interesting.

A program file might span several hundreds lines of code, thus finding vulnerabilities in it during code quality controls and code review might not be compatible with the limited time available to the developers. To help developers/code reviewers to address the above issue, one might use a linter or static analyzer that suggests specific lines of code may be responsible for the vulnerability. However, it may still be necessary to abstract away from the surrounding code and focus on the key instructions to effectively identify and address the vulnerabilities.

To help developers, a company considers to show developers only a ***slice***: a fragment of a program that is recursively extracted from a seed of some initial instructions.

- One finds the variables that are modified by those initial instructions and then collects the subsequent instructions where the values of these variables are used for the computation (forward slicing).
- Dually, one collects the variables whose value is used in the initial instructions and go back to the preceding instructions to see where these variables have been assigned (backward slicing).
- The process is iterated considering the added instructions as the new initial instructions until no more instructions can be added.

The idea behind the experiment E2a is that giving developers a slice of the file instead of the full file helps to find more vulnerabilities. In this report, we describe a controlled experiment with totally 81 master students, who were asked to distinguish the vulnerable codes from 4 files with four different vulnerability types, separately path traversal (Path), injection (User), cross site scripting (XSS) and denial of service (DoS). Each subject will receive 2 complete files and 2 sliced files (different from group to group). In the training procedure, all subjects received the same material. Ergo, the intervention in this experiment is vulnerability types.

It is important to consider the control factor of **vulnerability types** when implementing the above processes. It is possible that certain types of vulnerabilities may be more easily detected through the process being used, while others may not be as effectively identified. As a result, it may be necessary to distinguish which types of vulnerabilities are more readily detectable through the specific detection process being used. By controlling for the type of vulnerabilities, we can better understand which ones are more or less challenging to detect and remediate using the process being evaluated.

The data from the past shares the same setup and similar questions with the experiment in 2023, so in this report we may **combine** them together to show a more convincing result with more available data samples. Besides, we assume the data from last year's experiment has the same ground truth as this year.

2. Metric(s) of Success

Explain how you would measure the success of the intervention using the data that would be given you. This is not the analysis which will happen later. Here you just explain why this is a good choice in your opinion.

RQ 1: The *actual effectiveness in terms of correct answers* of the slicing intervention measured by correctly identified vulnerable lines rather than wrong not relevant lines identified by the participant.

In this report, we separately use measured precision per file, measured precision per group, binary overall metrics found/not found (all/one vulnerabilities) per file and self-defined metrics (which is the accuracy among subject's 4 questions not among all lines) to evaluate the effectiveness of slicing intervention. The details of these metrics will be elaborated in Section 5.

RQ 2: The *perceived usefulness* of the slicing intervention as collected by participants' questionnaires with Likert scale-like questions.

Answers to the questionnaire are mapped to 1 to 5 accordingly. For example, there are questions about the difficulty of identifying different threats, in which 0-20% is mapped to 1 and 80-100% is mapped to 5.

These metrics are picked simply because they are intuitive and ingredients of most tests, and we may derive the conclusion whether our conclusions are statistically significant. These results should be controlled based on the factor describing the vulnerability.

3. Subjects & Process

Describe the background of the experiment subjects (who participated in the experiment). Describe the process that was used to perform the experiment in a replicable manner (pay attention to details, e.g. what were the exact steps, is randomization of steps present, etc.). ***Explicitly mention the data on the past study and how this could impact your results.***

Participants of the experiment are master students from VU Amsterdam and all of them are computer science backgrounds. There are totally 81 subjects attending the experiment. However, some of them haven't taken part in the training procedure and some of them don't consent to use their data. After filtering the invalid data, the grouping situation is in TABLE 1 and their specialization is visualized in FIGURE 1.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Experiment Data	12	12	15	18	9	15
Valid Data	10	23	25	16	8	15

TABLE 1

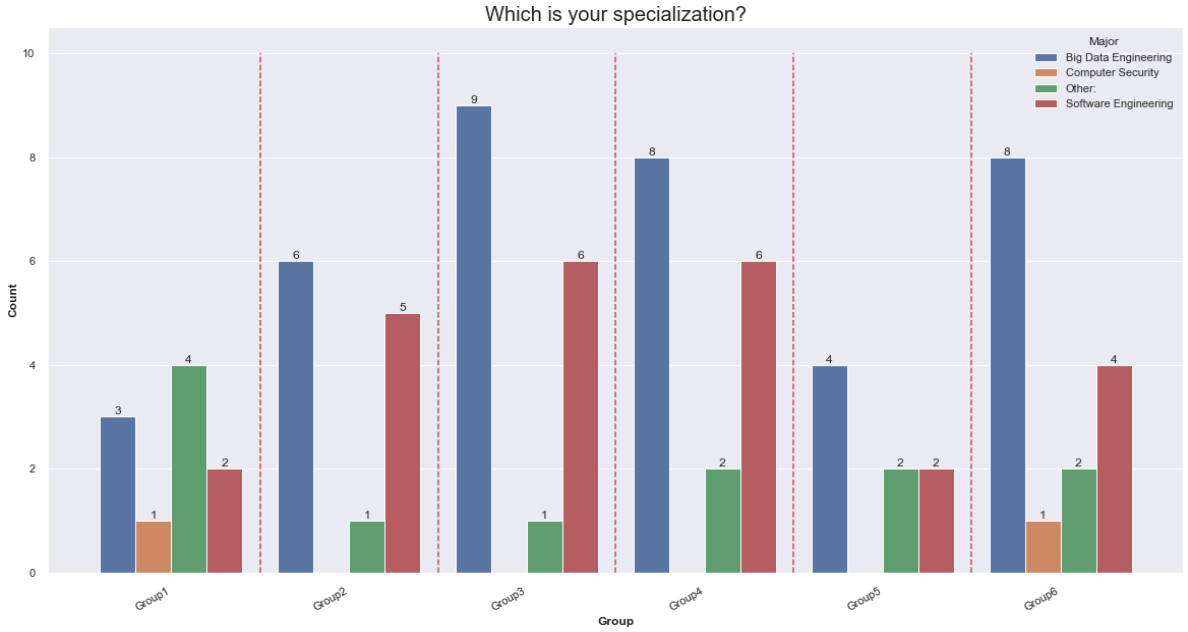

FIGURE 1

From the above figure, we can observe that the subjects' background of grouping is fairly even for Group 2, 3, 4 and 6. However, it is clear that the distribution of subjects in Group 1 and Group 5 has a large difference compared with other groups of subjects. Also, these two groups are the least numerous. Although the different backgrounds of the subjects will definitely influence the experiment results, according to the statistics, the bias can be ignored if we exclude the Group 1 and Group 5 results.

The main step of the experimental process is the execution of the comparison. The replicable process is specifically as follows:

0. ATTENDED training session

Participants were provided with training videos and slides on four different kinds of vulnerabilities (Path Traversal, Injection Vulnerability, XSS, DoS). They were assigned to 6 different groups randomly. Note that using VSCode as the editor was recommended but not mandatory in this experiment. Besides, the teacher controlled both groups having the same reading time and tried to make sure all the subjects had sufficient time to read the material and have a clear understanding of what the task asked them to do.

1. PARTICIPATED in an experiment

The second step is conducting intervention. After reading and watching all the material, the relevant knowledge background of the subjects was firstly investigated. Then, participants were required to find the vulnerable lines of code into the Java files, which is not about reporting how to fix them or why it is vulnerable or reporting where else to find information on this vulnerability. 6 groups were assigned with different file to analysis:

Group 1: dos_s, path_f, user_s, xss_f
Group 2: dos_f, path_s, user_f, xss_s
Group 3: dos_f, path_s, user_s, xss_f
Group 4: dos_s, path_f, user_f, xss_s
Group 5: dos_f, path_f, user_s, xss_s
Group 6: dos_s, path_s, user_f, xss_f

All files were in Java. File ending with “s” is a sliced version of the corresponding file ending with “f”, which is elaborated in Section 1.

	Full version from Github	Sliced version
Path Traversal	path_f.java	path_s.java
User Enumeration	user_f.java	user_s.java
XSS	xss_f.java	xss_s.java
DoS	dos_f.java	dos_s.java

TABLE 2

2. ANSWERED question in the Qualtrics survey

After identifying the vulnerable lines, participants were asked to fill out a survey, which was about some perceptual questions like their confidence in threat identification and perceived difficulties. This evaluation is very important to the experiment because it reflects the subjects’ knowledge of the material and the subject's intuitive perception of the experimental questions and their answers.

3. DETERMINED the ground truth

All the answers are recorded, the related time is recorded, and all the relevant information is aggregated. All data are aggregated into two Excel files according to the training and experiment procedures, combining with two Excel files with last year data. We also need to determine the ground truth of this experiment by two independent assessors with an agreement or consensus meeting. Then we can use two types of validation procedures: automatic and manual. An automatic validation procedure would use an algorithm to validate the results, while a manual validation procedure would involve human input to validate the results.

The below two processes are not exactly part of the experiment process, but we put them here just for the sake of clarification.

4. PREPROCESSED the collected date

We performed data cleaning on the data in four excel, specifically deleting the dirty data based on the duration time, whether they consent to the usage of their data and whether they took the training procedure. After that, we matched the participants’ determined lines of code against the ground truth, and identified our own criteria to get true positives. Then, we proposed various statistics measures for evaluation, plotted the data for easier comparison and elaborated the results quantitatively and qualitatively.

5. CONCLUDED the results and experiment

We conducted the analysis procedure by using a statistical test. After that, we analyzed the subjective data of all subjects and compared it with the results of our test. We also conducted the same procedure for the last year data, and compared them with the data from another university. Note that we didn’t combine the last year data and this year data together to conduct the statistical test for RQ1 because of the ambiguity of the ground truth for the last year data. As for the RQ2 (perception), because of the same types of vulnerabilities, the above concerns are not impactive anymore. We just applied the same analysis procedure and compared the conclusions. Therefore, the past data could not impact the final results. At last, we generated the conclusion.

4. Discussion and Limitations

Reflect on this experiment and describe what are the key criticalities and limitations of this particular experiment design. The discussed limitations and critical points must be specific to the experiment (e.g. identified co-founding variables), and not general limitations of experimentation.
Explicitly mention if the data on the past could impact your results.

The key criticalities of this experiment design are the intervention implementation, the determination of the ground truth and the success metric. These three factors are so crucial. If the intervention and control of the experiment fail, the validity and usefulness of collected data cannot be compromised. All the effective analysis is based on a correct implementation of an intervention. Otherwise, all the data is dirty data and is considered to be of poor quality. In regard to the determination of the ground truth, it is necessary for determining the metrics. Since judging vulnerability is a relatively subjective process, we do not have very accurate ground truth for each file. We have the correct answers about which lines are vulnerable in each file, however, we can also say the function or class containing the vulnerable lines are also vulnerable. Therefore, it is better to use precision as the success metric. Note that the definition to determine the true positive variable is diverse, and we will further explain it in the next section. As for the determination of the success metric, it is essential to the validity of experimental results, because the experimental conclusions we get are based on valuable metrics. The experimental conclusions are meaningless without a good success metric. Because we only need to consider the correctly identified vulnerable lines rather than wrong not relevant lines identified by the participant, we opted to mainly use precision instead of accuracy and recall as a success metric.

The key limitations of this experiment design are the sample size, the subjects' background and the randomization. Due to the small sample size, the statistical power of the experiment is limited. A good solution would be to combine the past data with this year's data to expand the sample size. However, due to the last year code files being not provided and the absence of some relevant data, after careful deliberation, we decided to not combine two years data together in analyzing RQ1 and only use the past data as a control data. On the contrary, we combine the two years' perception data together to analyze RQ2. Secondly, the experiment does not account for the confounding variables that may affect the ability of the subjects to identify vulnerable lines, such as their level of expertise in the field or attention during the training. From FIGURE 1, we can perceive that the randomization is not successful for Group 1 and Group 2. Hence, the utilization of data from these two groups are doubtful.

Besides, from Section 8, we can observe that the difficulty of finding vulnerabilities in provided files, especially the unsliced files, apparently exceeds the capability of the subjects. The precision rates are terrible for most groups. Furthermore, the explanations of why they select specific lines as the answer is impossible to utilize, unless we pretained a nlp model and use the model to label those explanation texts, we can only use those texts in the manual validation procedure. Therefore, this is also a key limitation. Changing the task into a multiple choice task might be a possible solution.

Ground Truth Decision

5. Definition

You have collected from the subjects some experimental artefacts and you should specify what makes the artefact measurable (for example correct or incorrect threats) or not qualified for evaluation of the success metrics (for example missing values or fields). If you have multiple measures of success (for example finding two types of vulnerabilities) the definition should be given for each measure.

In our experiment, the "artefacts" referred to software code that are being evaluated. The lines of artefacts that invoke vulnerabilities have been located through our analysis and comparisons with the corresponding patch. Vulnerabilities are identified in two ways, as shown in the form below, either by specifying which line is the source of the vulnerability, or by specifying the range of function that contains the vulnerability.

File Name	Vulnerable Lines	Outer Function Range
dos_f.java	463, 475	[454, 482]
dos_s.java	51, 55	[47, 59]
path_f.java	368, 369, 370	[361, 434]
path_s.java	76	[75, 96]
user_f.java	147, 148, 151, 153, 159, 163	[142, 167]
user_s.java	61, 62, 66	[58, 69]
xss_f.java	441, 444, 445, 446, 447	[425, 458]
xss_s.java	73, 74, 75, 76	[65, 79]

TABLE 3

Based on TABLE 3, we have four different ways to define the true positive of identified vulnerabilities. The definitions or **criteria** are as follows.

1. When all and only exact the Vulnerable Lines are found in a java file, the subject's answer (for that type of vulnerability) is defined as the correct answer. Everything else is considered an incorrect answer.
2. When one of the Vulnerable Lines is found in a java file, the subject's answer (for that type of vulnerability) is defined as the correct answer. Everything else is considered an incorrect answer. We named this criteria “**BoolCorrect**”.
3. When one of the lines in the vulnerability’s Outer Function Range are found in a java file, the subject's answer (for that type of vulnerability) is defined as the correct answer. Everything else is considered an incorrect answer. We named this criteria “**BoolCorrectOuter**”.

4. We build a slice with the instructions suggested by the student participant and then check whether all vulnerable instructions are inside the slic*e*. If inside, the subject's answer (for that type of vulnerability) is defined as the correct answer.

The restriction ranges from strong to weak. The first one might be overly restrictive, and the last one might be too optimistic. If we apply the first definition of correctness, we can use the mean, geometric mean and standard deviation of **precision per group** as the measures of success. The mean, geometric mean and standard deviation of **precision per file** are also applicable. If we apply the second and third definitions of correctness, we can only obtain a binary metric (found/not found per file) for each subject. We can calculate the mean and standard deviation of this **binary metric** among different groups or among different files. Except for the above measures, we also define a metric called **accuarcy among the subject's answers**, obtained by dividing correctly differentiated vulnerable lines by the number of lines answered by the subject.

6. Process and Guidelines

You might give both an automatic or a manual validation procedure. If you give an automatic procedure you should specify the algorithm and what are the possible errors that this might introduce in the measure of success. For a manual procedure you should start by assigning artefacts to different group members, start a preliminary evaluation, come to common scoring guidelines (reported), and then continue scoring.

After defining the criteria of the correctness evaluation, we need to apply a ground truth generator to find the truly good ground truth. We applied an automatic procedure, which is a python script, to evaluate subjects 'artefacts. The automatic procedure contains three steps, separately artefacts preprocessing, criteria selection and correctness determination.

Artefacts preprocessing: Since many subjects did not fill in the questionnaire as required, this step is the most tedious step. We automatically transfer the subjects' answer into the wanted data type (list) based on rules in the below table.

Input	Output
3;4;5	[3,4,5]
The result should be 3, 4, 5	[3,4,5]
3,4,5	[3,4,5]
3-5;7	[3,4,5,7]
3:5,7,10	[3,4,5,7,10]
NAN	[ ]

TABLE 4

Criteria selection: After the preprocessing, we defined the ground truth based on the four criteria and transfer the ground truth into the wanted data type (list) and find the intersection

between two lists by: # intersection= [set(a).intersection(b)) for a, b in zip(A,B)]#, where A is the ground_truth list and B is the sbject_anwser list.

Correctness determination: Finally, we determined the correctness of each subject based on different criteria. We used # min(1, len(intersection) / len(ground_truth))# for the criteria 1, and the output of this function is the precision of a specific subject for a specific file. We used # min(1, len(intersection))# for criteria 2 and criteria 3. Note the output of the above function is a binary, which is whether “0” or “1”. We can use the output to calculate the precision and recall ratio among each subjects’ four questions.

There may be some possible errors regarding the preprocessing step as our script cannot conclude all the possible situations. As for the correctness determination, we didn’t consider the nearby lines as the possible correct answers as the range of criteria 3 and 4 are big enough. Besides, we do not consider the subjects’ explanation on the vulnerabilities they found, as they are subjective and have high entropy. We will refer those explanation texts in the manual validation and error correction step. Generally speaking, the most serious error is incorrectly labeling the subjects’ correctness. Our algorithm effectively avoids this problem, and this can be verified in the next section.

7. Validation and Error Correction

If you do an automatic analysis you should select a random subset of artefact which you manually validate to determine the error rate of the automatic process.

We adopted a manual validation procedure for our algorithm. Firstly, we randomly sampled10 artefacts in a random sequence and assigned them to different group members, which are two independent humans. All the artefacts were unmarked and accompanied by the subjects’ explanations. We applied the human assessors with conflict resolution method as in FIGURE 2 to generate a reference ground truth for the 10 sampled artefacts. The preliminary evaluation would involve each group member individually assessing the artefacts they were assigned and providing a preliminary score or evaluation. The score is defined as a binary (whether “0” or “1”) in this experiment. Then, the two members would then work together to come up with common scoring guidelines that everyone agrees on, which would be reported. During the discussion, we came to a conclusion that the explanation texts are really an important reference and we decided to apply the **BoolCorrect** as the correctness criteria in this procedure. Once the common scoring guidelines are established, the group members would continue scoring the remaining artefacts according to these guidelines. This approach is intended to minimize the potential for bias or inconsistency in the manual validation process, since scoring is independent. The validation results are shown in the below table. The accuracy of our automatic analysis is as high as 95 percent. The two errors exist because after the human assessors read the explanation texts, both of them agreed the correctness, although the answered lines are not exactly matched with the ground truth. With regard to the low error rate, we didn’t apply the error correction for the misaligned results. This is also because reading all the explanation texts is highly inefficient.

	Human Assessors	Algorithm
Truly Good	9	7
Truly Bad	31	33

TABLE 5

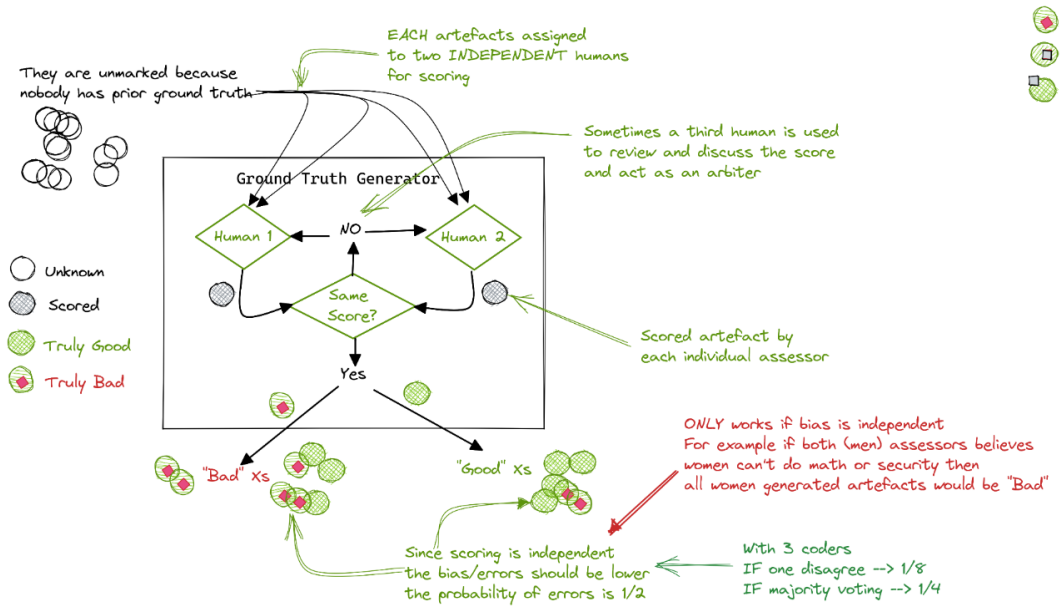

FIGURE 2

Analysis Description

8. Descriptive Statistics

Present key characteristics of the data you are analyzing using some basic descriptive statistics measures (e.g. mean, standard deviation of subjects or other metrics of the intervention). Add graphics is appropriate. ***Explicitly mention if the data from the past has been merged or is analyzed separately.***

After determining the ground truth from the source code, we first processed the data and filtered out the invalid data based on whether they consent to the usage of their data and whether they took the training procedure. We also filtered out the replicate data. Then, we visualized the duration time to find the outliers. Luckily, according to FIGURE 3, nearly all subjects finished in the requested time, and the timeout is not exceeded too far. Therefore, we do not need further cleaning. Following the preprocessing stage, we established three specific criteria and subsequently implemented an automated procedure to ascertain the true positives for each subject. These true positives were then utilized for further analysis of the data, using basic descriptive statistical measures. In this section, we conduct interpretation and analysis based on two research questions. When analyzing RQ1, we also conduct in-depth quantitative and qualitative analysis based on different aggregation methods. Note that we didn't merge the two years data together for RQ1 because of the ambiguity of the ground truth for the last year data. We just analyzed them separately and compared the results. As for RQ2, we merged the data together for the sake of compatibility.

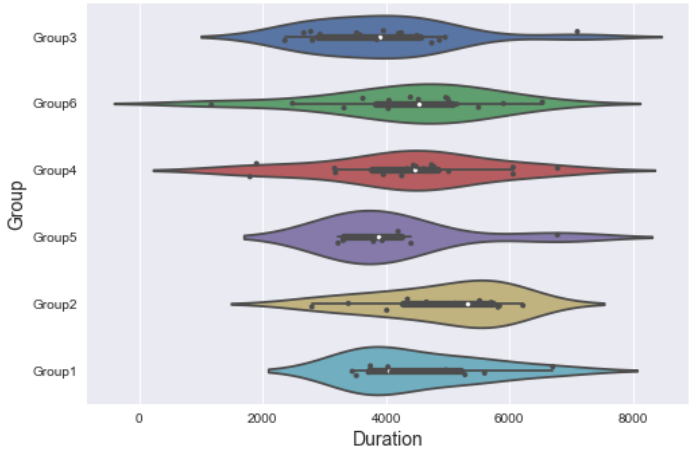

FIGURE 3

RQ1: Measurement Per Group

Applying the first criteria, and following the correctness determination step, we can obtain the precision of a specific file for each subject. FIGURE 4 is the boxplot which reflects the precision for each subject in the six groups. We can observe that the precision on original files of DoS and Path is really terrible regardless of median, minimum, maximum or interquartile range. Only very few of subjects successfully find out one or all the vulnerabilities in Path and DoS. As for the Injection and XSS, no subject finds all the vulnerabilities even if he was provided with the sliced files. As the premise of all high precision situations is that the subject distinguishes sliced files, it is also clear that the precision of subjects on sliced files is higher than the precision of subjects on the original files no matter in which group.

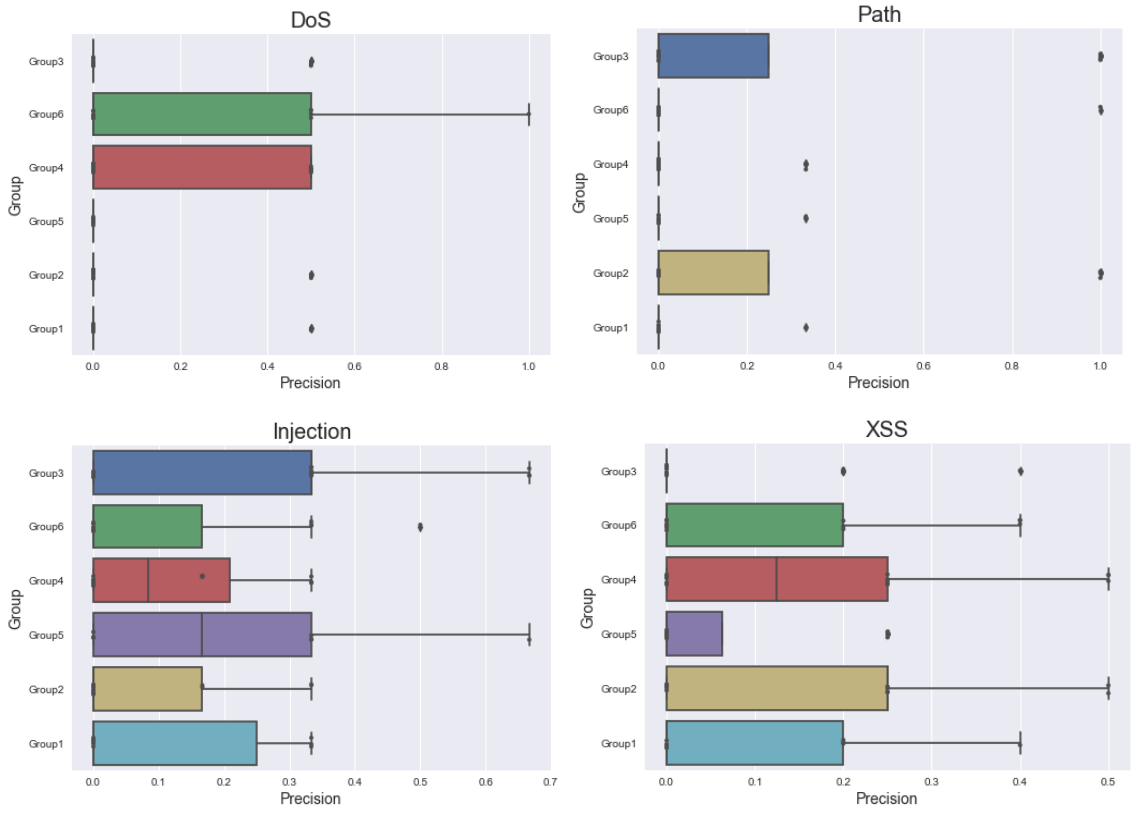

FIGURE 4

When applying the second criteria (BoolCorrect) and the third criteria (BoolCorrect), we can obtain a binary metric following the correctness determination step. This represents that whether the subject's answer on the specific file (e.g. DoS) is correct or not. Therefore, for each subject, he or she has four binaries, we can use these four binaries to find the mean, median accuracy of different groups.

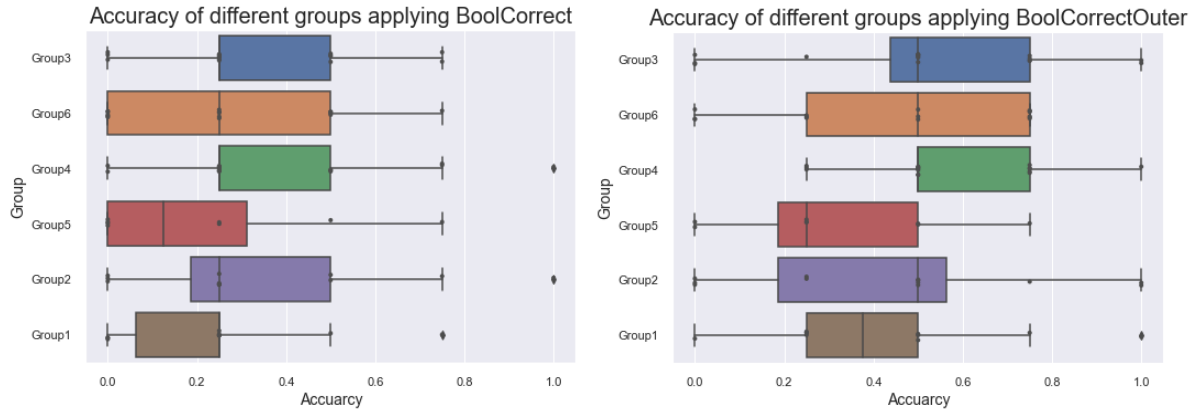

FIGURE 5

FIGURE 5 provides valuable insights into the performance of different groups. It is evident that Group 1 and Group 5 exhibit the worst performance across all measures, including median, maximum, and interquartile range. This finding validates our conclusions found in the subject background analysis. In addition, the performance of Group 6 is not optimistic. Interestingly, when applying BoolCorrect, the performances of the remaining three groups are relatively similar. However, when applying the weakest constraint, BoolCorrectOuter, all groups show a marked improvement in accuracy.

We cannot judge the success of intervention based on the above aggregation (measurement per group), but we can verify the success of randomization by analyzing the average accuracy of each group. When the results differ significantly between groups, as observed in our experiment, it is reasonable to infer that other potential factors may be affecting the subjects' performance, for example, the difficulty of the tasks. Further analysis is necessary to identify and mitigate these factors in future experiments to ensure accurate and reliable results.

RQ1: Measurement Per File

Different from the precision for each group, we can also calculate the precision for each individual file. FIGURE 6 depicts a box plot of the precision achieved for each file when applying the first criteria. It is apparent that subjects who were able to distinguish sliced files performed much better than those who distinguished full files regardless of median, maximum or interquartile range, highlighting the importance of this process in the overall success of the intervention. However, the high correctness requirements of this criteria pose a significant challenge, resulting in unsatisfactory performance for most subjects, with only a few achieving full precision. This low success rate also hinders the availability of experimental results, and the overall measure of success of the intervention might be underestimated. Consequently, it may not be prudent to conduct an in-depth analysis of the

data for this criteria at this stage. As such, we may need to refine the criteria to achieve better results.

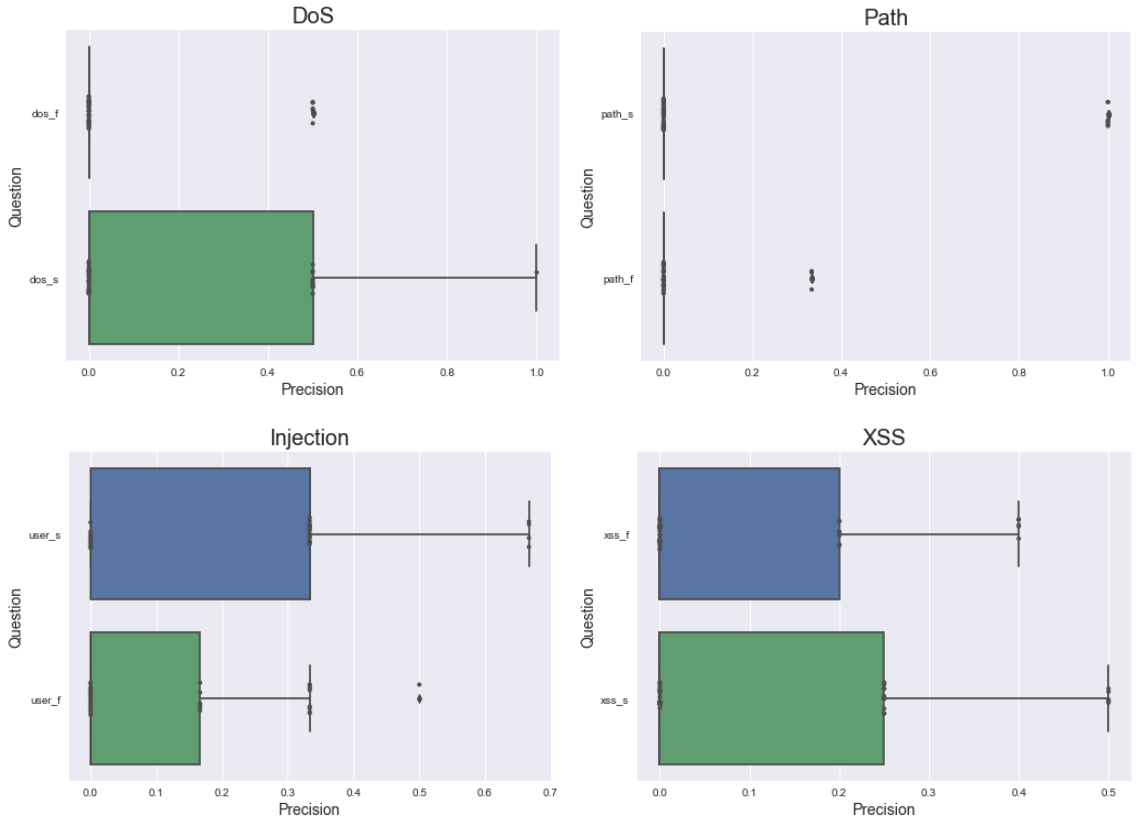

FIGURE 6

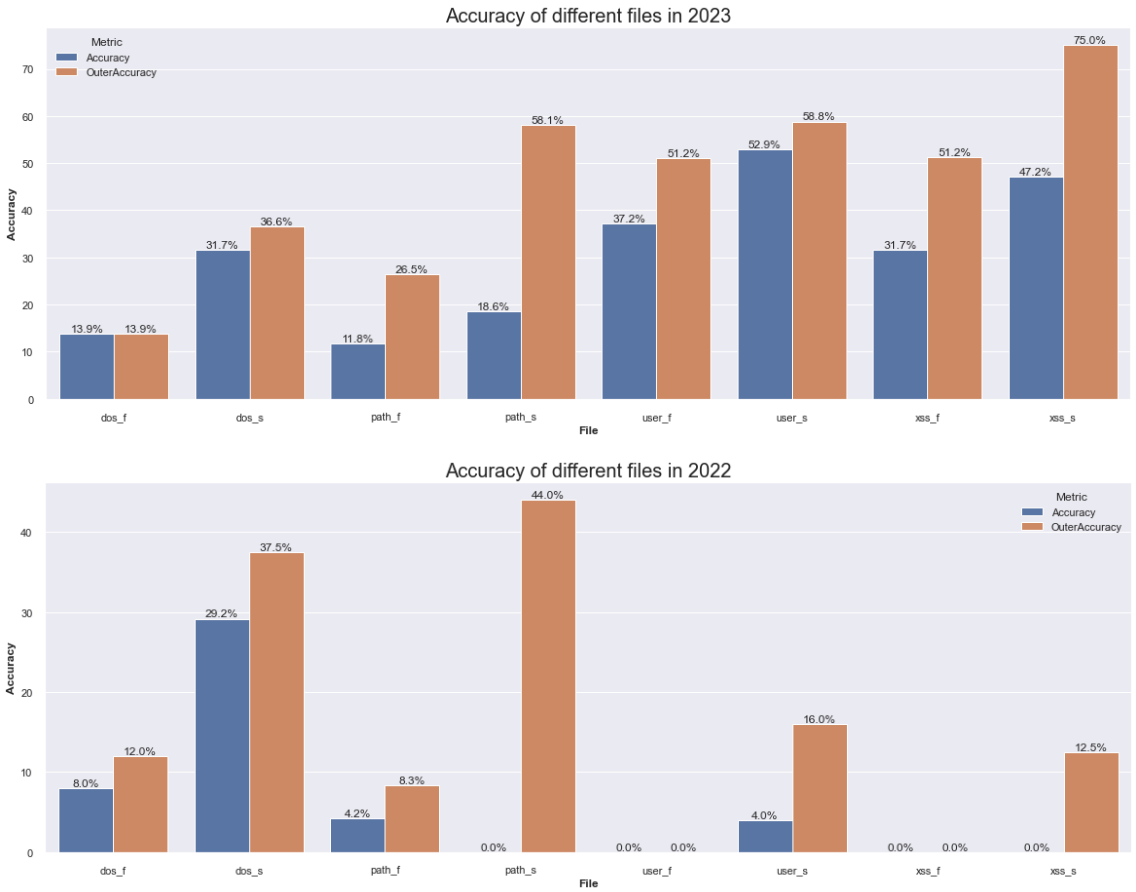

FIGURE 7

FIGURE 7 depicts the mean of accuracy among each file when applying the BoolCorrect and BoolCorrectOuter criteria for the two years data. The blue bars represent the average accuracy when applying BoolCorrect. The orange bars represent the average accuracy when applying BoolCorrectOuter. The procedure of calculating the accuracy for each subject is identical to the elaborations above FIGURE 5. This figure very intuitively shows the effective improvement of the accuracy and efficiency of finding vulnerabilities by using the sliced tool. Whether it is this year's data or last year's data, **the impact of intervention is extremely significant**.

It is worth noting that the process of processing last year's data is exactly the same as the process of processing this year's data. However, due to the lack of descriptions of last year's source code and related information, we can speculate from the above figure that last year's code file allocation may not be consistent with this year's code file allocation. As a result, some columns in last year's data exhibit a data accuracy rate of 0. This is why we do not integrate two years of data when analyzing RQ1. By comparing the valid data from the two years, we found that the overall distribution of the data remains relatively consistent.

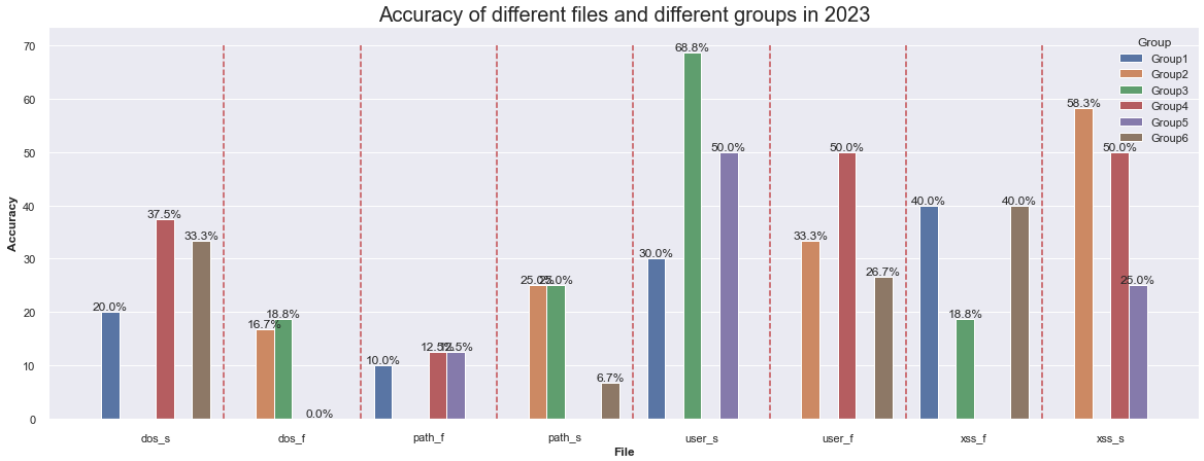

FIGURE 8

	Dos		Path		Injection		XSS	
Group	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Group 1	0.30000	0.48305	0.30000	0.48305	0.40000	0.51640	0.70000	0.48305
Group 2	0.16667	0.38925	0.50000	0.52223	0.33333	0.49237	0.75000	0.45227
Group 3	0.18750	0.40311	0.68750	0.47871	0.75000	0.44712	0.43750	0.51235
Group 4	0.43500	0.51235	0.31250	0.47871	0.68750	0.47871	0.81250	0.40311
Group 5	0.00000	0.00000	0.12500	0.35355	0.50000	0.53452	0.62500	0.51755
Group 6	0.33333	0.48795	0.53333	0.51640	0.46667	0.51640	0.46667	0.51640

TABLE 6

FIGURE 8 shows the mean of accuracy among each file and among each group applying the BoolCorrect criteria. TABLE 6 shows the mean and standard deviation of accuracy among each file and among each group applying the BoolCorrectOuter criteria. Note that nearly all the standard deviations of accuracy are very high. We can generate the same conclusion from the FIGURE 8 and TABLE 6, as **the impact of intervention is extremely significant.**

RQ2

According to the measurement mentioned in section 2, we have transformed the perceived usefulness to numbers 1 - 5. This year data and past data are compatible, so we combine them and analyze them together. In this way, we can get more accurate statistics.

TABLE 7 shown below contains the numeric feature of file usefulness.

File Groups	Mean	Standard Deviation
dos_f	2.5	1.1319231422671772

dos_s	2.732394366197183	1.125351231292292
path_f	2.984848484848485	1.1347463238395623
path_s	2.463768115942029	1.0710919546030075
user_f	2.563380281690141	1.2071663288572463
user_s	2.671875	1.09050858977589
xss_f	2.289855072463768	1.1433671259893343
xss_s	2.257575757575758	1.2099412214699956

TABLE 7

Based on the mean and standard deviation values provided in the table, we can observe that the usefulness score falls within a narrow range, with most of the scores ranging between 2 and 3. Furthermore, the standard deviation value, which is slightly above 1, suggests that the scores are not widely dispersed.

It is worth highlighting that *path_f.java* was rated as the most useful file, while xss_s was considered the least useful. Interestingly, *xss_s.java* was also the file that triggered the most diverse opinions among the participants, while path_s garnered a more consistent rating across the board.

Taking into account these observations, it is possible to conclude that the participants' perception of the files' usefulness was relatively consistent, with the exception of *xss_s*. This result could potentially inform future decisions on how to optimize the performance of the files and improve their usefulness.

The distribution of the data is shown in the following box plots. Each plot represents a different threat type, with the plot title being the filename and the y-axis representing the perceived usefulness metric. By examining the plots, we can observe that the distribution of perceived usefulness varies across different threat types. It is worth noting that, in general, the median perceived usefulness of the sliced version is usually higher than that of the full version, except for the file *path_s.java*. Interestingly, when identifying the files *path_s.java* and *user_s.java*, participants seemed to agree that the sliced version was more useful, whereas this was not the case for *dos_s.java* and *xss_s.java*. It is important to remember that the y-axis denotes the usefulness score, with higher scores indicating greater perceived usefulness.

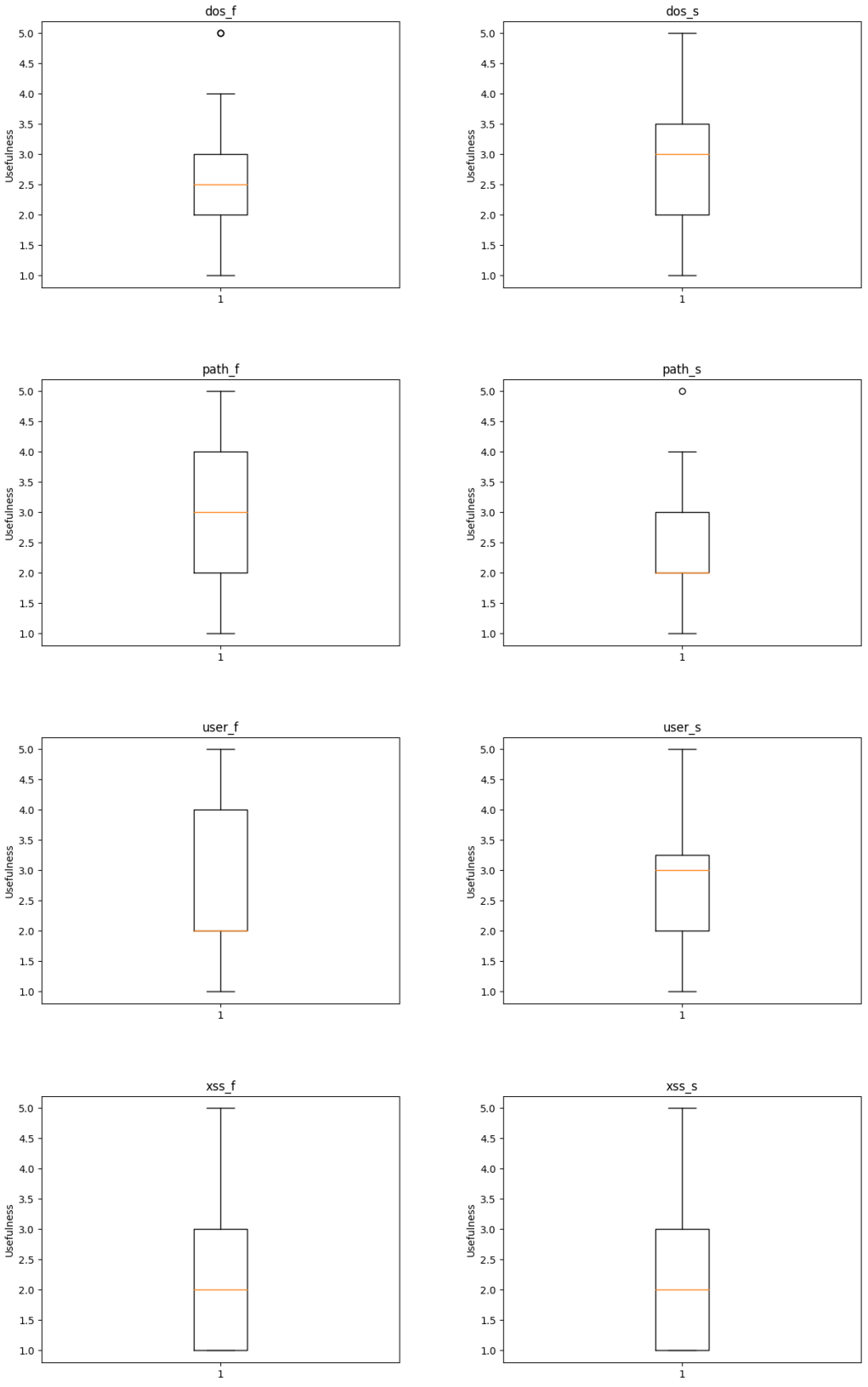

FIGURE 9

To determine whether these data follow a normal distribution, we can examine the Q-Q plots displayed below. The y-axis denotes the sample quantile with the x-axis denotes the

theoretical quantile. These plots compare the distribution of the data to the expected normal distribution, assessing whether the data points fall on a straight line. In this case, the Q-Q plots reveal that the samples are too sparse to accurately assess their normality. Therefore, we may need to consider alternative methods to analyze the data besides the t test.

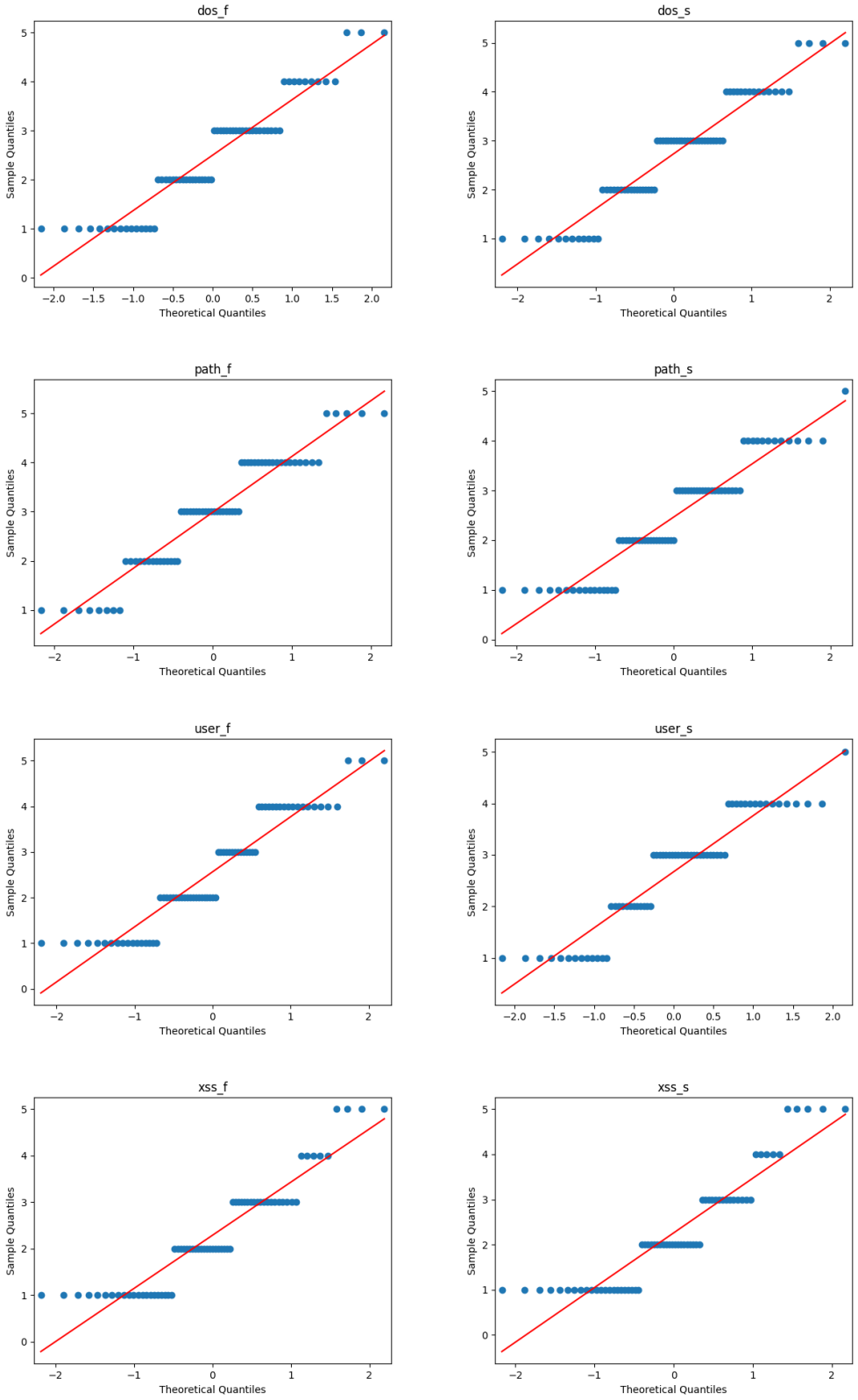

FIGURE 10

We can hardly describe them as normally distributed because Q-Q plots do not form diagonal lines. As a consequence, we adopted a test that does not require the sample to be normally distributed.

9. Analysis Procedure and Results

Describe the statistical tests used, including the explanation of which hypothesis the test is addressing. The characteristics of the presented data (in Section 5) must be in line with the assumptions (about the sample) that the test requires. Present your dependent / independent variables, and describe the steps you took to perform the analysis. Add graphics if appropriate. Describe the final statistical results.
Explicitly mention how you decided to analyze the data (a replication from past study, a new analysis observing different measures or using different statistical tests).

In this section, we perform the analysis with the Mann-Whitney test, which is suitable for non-normally distributed small samples, and the result is shown below (all null hypotheses are that the full version of the file is perceived as less useful than the sliced version).

RQ1

We applied the one-sided Mann-Whitney test using the same data as to visualize the FIGURE 7, which is the mean of accuracy among each file when applying the BoolCorrect and BoolCorrectOuter criteria. The procedure of calculating the accuracy for each subject is identical to the elaborations above FIGURE 5. We separate the data into two class, respectively full version and sliced version, each one has four data points. We then used the Mann-Whitney test to find the degree of difference between these two classes. The null hypotheses are that the full version of the file is perceived as less useful than the sliced version and the difference is significant.

	u-value	p-value
BoolCorrect	3.5	0.9267554080323958
BoolCorrectOuter	2.0	0.9714285714285713

TABLE 8

Based on the above p-values, we find that the distribution of using the full version of the file is stochastically greater than the distribution of using the sliced version of the file, therefore, we cannot reject the null hypothesis. Thus, we can generate our conclusion from the Mann-Whitney test, **the impact of intervention is significant**. It is important to note that this finding supports the use of sliced versions of files in our experiment, rather than full versions, as they are perceived to be more useful.

RQ2

There is a manual but not automatic analysis as mentioned in Section 6. The validation is a test between the experiment conducted at VU and the past data to see if there are inconsistencies (about statistics). The comparison is conducted with Mann-Whitney test

between the two unis. The null hypothesis is that the two experiments are not statistically different from each other and there are no significant errors.

File Name	p-value
dos_f.java	0.6678510739027325
dos_s.java	0.9011965116408269
path_f.java	0.9884696829615185
path_s.java	0.46910137711833466
user_f.java	0.92142710115277
user_s.java	0.5483721960847759
xss_f.java	0.37777925510487104
xss_s.java	0.999364571953617

TABLE 9

In conclusion, the validation suggests that there is no significant difference between the two experiments conducted at different places.

The goal of investigating this question is to show how differently participants view the full version and the sliced version of the codes. We divided all 6 groups of participants into 4 groups corresponding to the files containing 4 different types of threats. In each group, the comparison between the perceived usefulness of full version and sliced version is conducted.

File Groups	p-value
dos	0.8879840923021165
path	0.00457338992070353
user	0.7330509516680677
xss	0.37270462057325515

TABLE 10

It is safe to say that in the group of Injection Vulnerability, XSS (Cross_Site Scripting), DoS (Denial of Service), the participants as a whole consider the sliced version to be more useful. However, in the Path Traversal group, we have to reject the null hypothesis because its p-value is smaller than the conventional threshold of 0.05, so the participants thought the full version of the file was more useful.

The analysis for RQ2 also adopts a manual validation procedure, which is a comparison between the experiment conducted in VU and the past data in another university. The comparison takes the form of a Mann-Whitney test between different groups of records.

There may be some possible errors regarding the sample size (which may lead to lack of evidence to support a hypothesis) or the inconsistency across years, so the experiment needs to be revised or repeated.

Summary of Experiments

End the report with your interpretation of the results and provide your main findings. Make sure that the main findings are in fact supported by the analysis described in Section 6 and explain how you derived those conclusions. Eventually, you will have to exercise your judgement in determining whether the effect is actually practically significant or just statistically significant or insignificant from all perspectives.
Compare your findings with the past study (did you find the same supporting evidence or does your analysis support a different explanation?).

The results of our study, which are presented in Sections 8 and 9, demonstrate the impact of threat type and file slicing on perceived usefulness and factual accuracy. Specifically, we found that participants' perceptions of usefulness varied depending on the type of threat they were presented with, as well as whether the file was presented in its entirety or in a sliced format.

Interestingly, our analysis revealed that when participants were presented with a path traversal threat type, they found the full version of the file to be more useful in identifying potential threats. On the other hand, when presented with other types of threats, participants found the sliced version to be more helpful.

Furthermore, the inclusion of historical data in our study serves to bolster the credibility of our findings. By comparing data from previous years to the present, we were able to show that there were no significant differences between the two, as discussed in Section 8 and 9. These results are statistically significant and underscore the importance of considering threat type and file slicing when evaluating perceived usefulness in security contexts.