Defining Controlled Experiments Inside the Access Control Environment

Said Daoudagh[1,2][a] and Eda Marchetti[1][b]
[1]*ISTI-CNR, Pisa, Italy*
[2]*Department of Computer Science, University of Pisa, Pisa, Italy*

Keywords: Access Control, Controlled Experiment, Goal-Question-Metric, Testing, XACML.

Abstract: In ICT systems and modern applications access control systems are important mechanisms for managing resources and data access. Their criticality requires high security levels and consequently, the application of effective and efficient testing approaches. In this paper we propose standardized guidelines for correctly and systematically performing the testing process in order to avoid errors and improve the effectiveness of the validation. We focus in particular on Controlled Experiments, and we provide here a characterization of the first three steps of the experiment process (i.e., Scoping, Planning and Operation) by the adoption of the Goal-Question-Metric template. The specialization of the three phases is provided through a concrete example.

1 INTRODUCTION

Nowadays, quality of Information and Communication Technology (ICT) systems and modern applications is strictly tied with the security and privacy. Among security mechanisms, a critical role is played by Access Control (AC) systems, which aim to ensure that only the intended subjects can access the protected data and get the permission levels required to accomplish their tasks and no much more.

Due to the complexity of AC systems, for ensuring the required security level, a key factor becomes the application of effective and efficient testing approaches: knowing in advance the criticality of the systems lets to put in practice efficacious corrective actions so as to improve the overall security of the system. However, testing phase is a time consuming, error prone and critical step of the development process, which involves different activities: from test strategy selection, to the test case derivation, from execution to the final test results evaluation. Bad choices in each stage of the testing phase may compromise the entire process, with the risk of releasing inadequate security solutions that allow unauthorized access from the *security perspective* or unlawful processing from the *legal perspective*.

In the last years different proposal are targeting the efficacious management of the testing phase by proposing techniques that combines Model Based Testing (MBT) (Utting et al., 2012) and Test driven Development (TDD) (Nanthaamornphong and Carver, 2017) techniques so that model-based tests can guide the development. Among them the Model-Based Test Driven Development (MBTDD) (Sadeghi and Mirian-Hosseinabadi, 2012) one of the first tentative for extending the TDD cycle is extended with MBT steps. However, the main issues of the MBTDD is that it does not deal with the reuse of test cases along the iterations. For this different improved solutions have been conceived in order to better support the test phase development, management and evaluation (Harumi et al., 2016). In line with these proposals, this paper provides a revised approach of MBTDD in the context of testing access control systems. In particular, the paper focuses on the use of controlled experiments for ensuring the integrity and replicability of the testing results.

In literature, different solutions are currently available for testing AC systems and their behavior (Bertolino et al., 2013; Bertolino et al., 2014a; Hu et al., 2017), but there are not standardized guidelines for correctly and systematically performing the testing process in order to avoid errors and improve the effectiveness of the validation. In particular, the lack of a formalized specification of the testing activity can have the following consequences: impossibility of replicating and controlling the process especially in case of regression testing (Yoo and Harman, 2012), difficulties in the generalization of the testing

[a] https://orcid.org/0000-0002-3073-6217
[b] https://orcid.org/0000-0003-4223-8036

Daoudagh, S. and Marchetti, E.
Defining Controlled Experiments Inside the Access Control Environment.
DOI: 10.5220/0009358201670176
In *Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2020)*, pages 167-176
ISBN: 978-989-758-400-8; ISSN: 2184-4348

results and consequent derivation of statistical significance values; and problems in defining and sharing a common testing knowledge so as to avoid recurring failures and speeding up the corrective process.

A reply to these issues comes from the software engineering context, where Controlled Experiments (CEs) (Juzgado and Moreno, 2001; Wohlin et al., 2012; Basili and Rombach, 1988) are commonly used to investigate the cause-effect relationships of introducing new methods, techniques or tools and to build a body of knowledge supported by observation and empirical evidence. Therefore, the controlled experiments let to validate the different activities of the testing process by means of the identification of important variables, the definition of specific testing models and objectives, and the derivation of empirical evidence. In the controlled experiment different treatments can be applied to, or by, different subjects, while other variables are kept constant and the effects on response variables are measured.

Authors in (Juzgado and Moreno, 2001; Wohlin et al., 2012) categorize experiments as either technology-oriented or human-oriented, depending on whether artifacts or human subjects have given various treatments. In this paper, we revise and customize the technology-oriented experiments in order to provide general guidelines for correctly end effectively performing the testing of the AC systems. Therefore, we provide the characterization of the first three (over the five) steps of the *Experiment Process* that namely are *Scoping*, *Planning*, and *Operation*. We refer to (Daoudagh et al., 2020) for a concrete application example as well as a detailed checklist of the required implementation steps.

Outline. Section 2 introduces the main concepts used along the rest of the paper related to Controlled Experiment, the Goal-Question-Metric, and Access Control, as well as the related work; Section 3 illustrates our proposal of a family of Controlled Experiments in the context of Access Control. In particular, in Sections 4, 5 and 6 we detail the first three phases of the Controlled Experiment; finally, Section 7 concludes the paper and depicts the future work.

2 BACKGROUND AND RELATED WORK

In this section we firstly describe the main concepts related with (1) Controlled Experiment; (2) Access Control & Testing; and (3) Goal-Question-Metric (GQM) used along the rest of the paper and their related works.

Controlled Experiment. Experiments (or CEs) are used in software engineering to investigate the cause-effect relationships. They consist of a well-defined *Experiment Process* including five specific phases: *(i)* Scoping, *(ii)* Planning, *(iii)* Operation, *(iv)* Analysis and Interpretation, and *(v)* Presentation and Package. However, in the Experiment Process it is not mandatory to finish an activity before starting the next one. As a consequence, it is possible to go back and refine a previous activities before continuing with next one. In this sense it is partially iterative.

The purpose of a CE is therefore to systematically define the elements necessary for ensure the integrity and replicability of the obtained results. Very briefly, the main element are: (1) *objects* on which the experiment is run are the *experimental units* and can involve the all the systems or part of it; (2) *subjects* that represent artifacts on which the methods or techniques are applied; (3) the outcome of an experiment is referred to a quantitative *response variable* (also called *Dependent Variable*); (4) each considered characteristic target of the experiment to be studied that can affect the response variable is called a *factor* (also called *Independent Variables*); (5) the possible values of the factors are called *levels*; and (6) *parameter*, i.e., any other invariable (qualitative or quantitative) characteristic of the software project that does not influence the result of the experiment.

Consequently, in each experiment a combination of alternatives of factors are applied by a subject on an unit. A defined and precise specification of the experiment guarantees both: the *External replication* (Judd et al., 1991), i.e., reproducing the experiment in different contexts and environments so as to increase the confidence in experiment results; the *Internal replication*, i.e., the repetition of the experiment more time in the same environment or condition to increases the reliability of the experiment results.

In Software Engineering field, CEs are gaining a lot of attention (Sjøberg et al., 2005; Ko et al., 2015) and different proposals are trying to give guidance on how to conduct CEs (Juzgado and Moreno, 2001; Wohlin et al., 2012). Following this tendency, our proposal want to come up with a Goal Definition Framework that enables one to conduct technology-oriented experiments in the Access Control (AC) context. More precisely, the novelty of our proposal is to provide general guidelines for correctly end effectively performing the testing of AC systems.

Access Control & Testing. AC systems are means to help organizations to improve their security from the point of view Confidentiality, Integrity and Availability (i.e., the CIA Triad). Often AC systems are regulated by Access Control Policies (ACPs) that define which subject is allowed to access a protected resources. ACPs are usually written by using the eXtensible Access Control Markup Language (XACML) (OASIS, 2013) standard. This standard defines both a reference architecture and a language based on XML to express ACPs and AC request/response.

One of the main the main components of XACML standard is Policy Decision Point (PDP), which evaluates the ACP against the request and returns the response, including the authorization decision. For more details about the XACML standard we remaind the reader to its specification (OASIS, 2013).

In literature, several works are focused on AC systems testing, and they can be mainly divided into the following research fields: i) test strategies definition (Bertolino et al., 2013; Bertolino et al., 2018); ii) test strategy assessment (Bertolino et al., 2014b; Lonetti and Marchetti, 2018; Daoudagh et al., 2019a); iii) test cases generation and execution (Bertolino et al., 2010; Hu et al., 2017); iv) test execution and oracle derivation which are focused on approaches for evaluating theAC replies to specific inputs (Daoudagh et al., 2015; Calabrò et al., 2017; Bertolino et al., 2018; Daoudagh et al., 2019b).

The lack of formality of the conducted studies in the above work do not enable external replication of the result. Differently our work want to contribute to formally and thoroughly conduct CEs in the context of AC.

Goal-Question-Metric. Originally presented in (Basili and Rombach, 1988), the Goal-Question-Metric (GQM) paradigm proposes a top-down approach to define measurement: goals lead to questions, which are then answered by metrics. A GQM model is a hierarchical structure as presented in Figure 1 starting with a goal by specifying purpose of measurement, object to be measured, issue to be measured, and viewpoint from which the measure is taken (*Conceptual level*). The goal is refined into several questions that usually break down the issue into its major components (*Operational level*). Each question is then refined into metrics, some of them objective and others subjective (*Quantitative level*). The same metric can be used to answer different questions under the same goal as well as different goals (Basili et al., 1994).

In security domain there are a few proposals using the GQM and they are used to mainly identify security requirements and metrics. For example, authors in (Islam and Falcarin, 2011) used GQM approach to define clear and comprehensible measures for a set of established security requirements. The GQM approach based on Standard security metrics and on Service Oriented Architecture maturity is presented in (Kassou and Kjiri, 2013), where scholars aimed at supporting organizations to assess SOA Security as well as to ensure the safety of their SOA based collaborations. To assessing the security of data stored in cloud storage, authors in (Yahya et al., 2015) attempt to provide practical guidance and example of measurements using GQM. A more recent work is presented in (Weldehawaryat and Katt, 2018) where the authors presented a quantitative evaluation approach for defining security assurance metrics using two perspectives, vulnerabilities and security requirements.

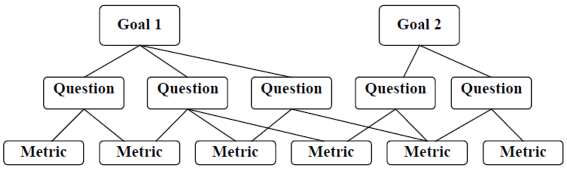

Figure 1: The Goal Question Metric (GQM) model (adopted from (Basili et al., 1994)).

Differently from the above works, our proposal aims at enabling the derivation of metrics for answering questions related to investigation goals in the context of AC. In particular, the intention is to enable CEs in the context of AC by covering all the phases of the process. In this paper however we focus on the first three phases of the process and we refer to (Daoudagh et al., 2020) for more details about the remaining phases.

3 A GQM PROPOSAL FOR ACCESS CONTROL TESTING

The general idea behind our proposal is to provide a set of CE families useful for formally and thoroughly describing scientific investigations in the context of Access Control (AC) systems. Indeed, our intuition is to use the standard and consolidated GQM template (Basili and Rombach, 1988), as guidance to select, and consequently classify, concepts of interest in the domain of AC. Then, exploiting the knowledge and the techniques typical of the software testing scientific environment, a concrete AC-based goal definition framework can be derived. This set will be well-defined, specific and achievable AC testing goals to be exploited for different experimentations.

The proposal of the paper, although grounded in a domain-related AC testing, represents an example of realization of CE families, that can be easily applied in all the domains where a scientific investigation in which a formal and rigorous fashion should be performed.

As in Figure 2, the proposal is composed of five conceptual components: the Goal Question Metric ①, the Access Control Context ② and Software Testing ③, which represent the conceptual models of the target experiment.

These models are integrated in the Goal Definition Framework component ④ so as to define a specialized GQM, which is the common basic knowledge for the AC families. Then, the GQM exploited in the Main Research Goal component ⑤ for defining scientific testing goals in AC testing process and therefore for selecting specific and achievable AC testing goals ⑥ to be evaluated in real context.

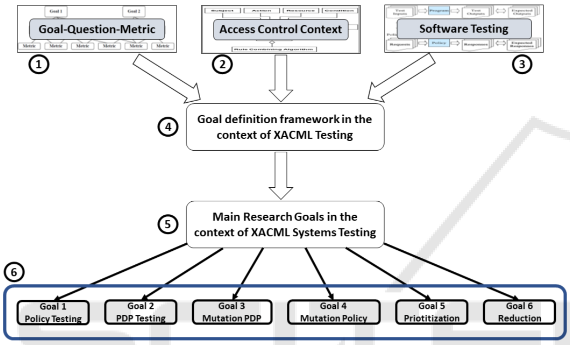

Figure 2: GQM Access Control Model.

In the following sections we illustrate how the use of a specialized GQM can be a important innovation for the development of Controlled Experiments in AC context. In particular, by referring to the structure of a CE presented in Section 2, we detail the execution of the first three steps of the process (i.e., *Scoping*, *Planning* and *Operation*), which are those that need to be specialized for the AC domain. We refer to (Daoudagh et al., 2020) for a complete example including also the last two phases.

4 EXPERIMENT SCOPING

The purpose of the scoping phase is to determine the foundations of the experiment by defining goals according to a specific framework. As described in the previous section, The idea here is to use the Goal-Question-Metric (GQM) method, integrated with concepts of AC and Software Testing for deriving a specialized template for the definition of CEs goals in the AC testing context. By referring to Figure 2, the scoping phase exploits the domain specific concepts of components ①, ② and ③ so as to define a reference framework, i.e., Goal Definition Framework (component ④). In the remainder of the section the use of the three components is better detailed.

According to (Basili and Rombach, 1988), the GQM template consists of five elements: (1) *object of study* is target entity of the experiment. It can be a product, process, resource, model, metric or theory. (2) *purpose* defines the intention of the experiment is. It may be to evaluate the impact of two different techniques or to characterize the learning curve of an organization. (3) *quality focus* is the primary effect under study in the experiment. It can y be effectiveness, cost, reliability etc. (4) *perspective* describes the viewpoint from which the experiment results are interpreted. Examples are developer, project manager, customer and researcher. (5) *context*is the environment in which the experiment is run. It defines which personnel is involved in the experiment (subjects) and which software artifacts, called objects[1] are used in the experiment.

Consequently, the intention of the GQM template is to *Analyze* <Object(s) of study> *for the purpose of* <Purpose> *with respect to their* <Quality focus> *from the point of view of the* <Perspective> *in the context of* <Context>.

Table 1: AC concepts.

GQM elements	AC concepts
Object of study	XACML-based PDPs XACML-based ACPs
Purpose	-
Quality focus	-
Perspective	ACP Architect AC System Developer AC System Administrator
Context	Subjects (XACML Policies) Objects (XACML-based PDPs)

AC Model. There are different access control model in literature, among them, in this study we refer to the Attribute-Based Access Control (ABAC) model and in particular to its implementation, i.e., the XACML standard. More precisely, we refer to both the ACP model and the XACML reference architecture.The objective here is to characterize the CE in the context of AC by gathering the main concepts, terms and components that can be used to formulate an interesting goals from the scientific point of view. The selected elements are then used in the GQM template for the *object of the study*, the *purpose*, the *perspective* and the *context*. The classi-

[1] Note that the *objects* here are generally different from the *objects of study*

fication we propose in this paper is summarized in table Table 1. In particular, the first column (`GQM elements`) lists the GQM element while the second one (column `AC concepts`), reports concepts useful for defining meaningful research investigation in the context of AC.

Software Testing. In literature different proposals exist that leverage well-known software techniques to test ACPs and AC mechanisms. By analyzing current literature, we summarize in Table 2 in the column `Software Testing concepts` some of the main software testing concepts useful in generic controlled experiment. We also classify them according to the GQM template elements (column `GQM elements`).

Table 2: Software Testing concepts.

GQM elements	Software Testing concepts
Object of study	Test case generation strategy Test case prioritization technique Mutation Generators Test case reduction technique Oracle Derivation
Purpose	Characterize Evaluate
Quality focus	Effectiveness Cost Size APFD Performance
Perspective	Researcher Tester Project manage User
Context	-

However, without the pretend to be exhaustive and in the aim of simplicity, the table reports a simplification of a possible classification. In particular, in this paper we limit ourself to the definition and assessment of a test case generation strategies, because they are recognized as ones of the most crucial activities of the testing process. In the assessment of the effectiveness of a test strategy, concepts as coverage criteria and mutation analyses or test oracle are often used, and therefore included in Table 2. We also add the prioritization and reduction concepts because they are commonly adopted techniques for reducing the number of test case to be executed and consequently the effort and time due to overall testing phase.

Goal Definition Framework. On the bases of the concepts of Table 2 the specialized Goal Definition Framework is derived. This is a comprehensive framework based on the GQM for the definition of research investigation goals for testing tools, methodologies and strategies in the AC (both ACPs and AC mechanisms) context. To the best of the authors' knowledge, this proposal is the first attempt to provide a formally and thoroughly solution for the definition of a Controlled Experiment in AC domain. Table 3 reports the conceived framework, which represent the output of component ④ of our proposal depicted in Figure 2.

Specifically Table 3 has a column for each of the five GQM where the identified AC and Software Testing concepts are reported: namely `Object of study`, `Purpose`, `Quality focus`, `Perspective`, and `Context`.

Research Goals in AC Context. Combining the elements of the different columns of Table 3 a defined and focused scientific investigation goals that enable the specification of CE in the context of AC can be identified. Thus, the Goal Definition Framework lets the definition of families of goals for the access control systems testing. In Table 4 a not exhaustive list of the mostly adopted research goals are reported. In particular, the fist column (`Research Goal`) reports a label associated to each defined goal, whereas, the second column (`Goal Definition`) contains the definition of the goal using the GQM template customized with a specific combination of the elements of Table 3. Note that not all the possible combinations of those elements enable the definition of an interesting a well-defined goal. It is up to the user of the framework to choose the correct combination depending on the concrete objective.

5 EXPERIMENT PLANNING

The Planning activity consists of different steps where foundation of the experiment is defined. More precisely, the context of the experiment is determined and the hypothesis is stated formally, including a null hypothesis and an alternative hypothesis. Then, we need to determine variables, both independent variables (inputs) and dependent variables (outputs), and to identify the subjects of the study. After the design step, which includes choosing a suitable experiment design, the instrumentation of the experiment is defined by identifying and preparing suitable objects and measurement procedures. As a part of the planning, it is important to consider the question of

Table 3: Goal definition framework in the context of XACML Testing.

Object of study	Purpose	Quality focus	Perspective	Context
Test case generation strategy	Characterize	Effectiveness	Researcher	Subjects (XACML Policies)
Test case prioritization technique	Evaluate	Cost	Tester	Objects (XACML-based PDPs)
Mutation Generators	Assess	Size	Project manager	
Test case reduction technique		APFD	User	
XACML-based PDPs		Performance	ACP Architect	
XACML Policies			AC System Developer	
XACML-based Oracle Derivation			AC System Administrator	

Table 4: Main Research Goals in the context of XACML Systems Testing.

Research Goal	Goal Definition
Goal 1: Policy Testing	***Analyze*** test case generation strategies ***for the purpose of*** evaluation ***with respect to their*** effectiveness and size of test suite produced ***from the point of view of the*** researcher ***in the context of*** XACML policy testing.
Goal 2: PDP Testing	***Analyze*** test case generation strategies ***for the purpose of*** evaluation ***with respect to their*** effectiveness and size of test suite produced ***from the point of view of the*** researcher ***in the context of*** XACML policy decision point testing.
Goal 3: Mutation PDP	***Analyze*** mutation generators ***for the purpose of*** evaluation ***with respect to their*** applicability ***from the point of view of the*** researcher ***in the context of*** XACML policy decision point testing.
Goal 4: Mutation Policy	***Analyze*** mutation generators ***for the purpose of*** evaluation ***with respect to their*** effectiveness and size of test suite produced ***from the point of view of the*** researcher ***in the context of*** XACML policy testing.
Goal 5: Prioritization	***Analyze*** test case prioritization techniques ***for the purpose of*** evaluation ***with respect to their*** effectiveness (rate of fault detection, using APFD (Average Percentage Faults Detected) metric) ***from the point of view of the*** researcher ***in the context of*** XACML policy testing.
Goal 6: Reduction	***Analyze*** test case reduction techniques ***for the purpose of*** evaluation ***with respect to their*** effectiveness (rate of fault detection, using APFD (Average Percentage Faults Detected) metric) ***from the point of view of the*** researcher ***in the context of*** XACML policy and PDP testing.

validity of the results we can expect. Validity can be divided into four major classes: internal, external, construct and conclusion validity.

In the remainder of the section, in order to clarify the steps of the Planning phase we refer to a real example: the testing of a PDP engine. This can be translated into the selection of the best test strategy for testing the PDP in order to improve its quality and reduce the testing effort. As reported in Table 5, in this case three sub-goals, each focuses on a specific research question, are identified:

- **RQ1 Effectiveness:** How much does the quality of a test suite produced by $Strategy_1$ (TGS_1) differ from the quality of test suite produced by $Strategy_2$ (TGS_2) in terms of Effectiveness, i.e., the mutation score?

- **RQ2 Size:** How much does the cost of a test suite produced by $Strategy_1$ differ from the cost of test suite produced by $Strategy_2$ in terms of Size, i.e., the number of test cases?

- **RQ3 APFD:** How much does the Average Percentage Faults Detected (APFD) of a test suite produced by $Strategy_1$ differ from the APFD of test suite produced by $Strategy_2$?

Context Selection. The first activity of the planning phase is the Context Selection. According to (Wohlin et al., 2012) the experiment contexts can be classified as in Table 6. Considering the PDP testing example, because two test strategies should be compared, the context is a *Multi-test within object study*.

Considering the implementation of the considered experiment, the Policy Decision Point is the Sun-PDP (Sun Microsystems, 2006) is the target PDP, (**One Object**). We decided for Sun's PDP engine because it is currently one of the most mature and widespread used engine for XACML policy implementation, which provides complete support for all the mandatory features of XACML 2.0 as well as a number of optional features. The strategies to be compared are the *Multiple* test strategy (Bertolino et al., 2013) and *XACMET* test strategy (Daoudagh et al., 2019b); a set of real world XACML policies are used for test case derivation (**Multiple Subjects**), and mutation techniques adopted to assess the test strategies considered; the comparison is done by evaluating the effectiveness, the size and the APFD of the test suite generated for each XACML policy, applying both strategies.

Hypothesis Formulation. We consider the following null hypotheses:

Table 5: Sun PDP Testing Goal.

Policy Decision Point Testing Goal (Goal 2)		
Analyze Multiple and XACMET Strategies ***for the purpose of*** evaluation ***with respect to their*** effectiveness and size of test suite produced ***from the point of view of the*** researcher ***in the context of*** Sun PDP testing.		
Research Questions		
RQ 1: Effectiveness	RQ 2: Size	RQ 3: APFD
Research Subgoals		
Analyze Multiple and XACMET Strategies ***for the purpose of*** evaluation ***with respect to their*** test suite effectiveness ***from the point of view of the*** researcher ***in the context of*** Sun PDP testing without constraints.	***Analyze*** Multiple and XACMET Strategies ***for the purpose of*** evaluation ***with respect to their*** cost in terms of number of test cases generated ***from the point of view of the*** researcher ***in the context of*** budget programming.	***Analyze*** Multiple and XACMET Strategies ***for the purpose of*** evaluation ***with respect to their*** effectiveness in terms of APFD ***from the point of view of the*** researcher and quality manager ***in the context of*** interruption of Sun PDP testing activity.
Metrics		
m1: Effectiveness	m1: Size of the test suite	m1: APFD

Table 6: Experiment context classification.

		# Objects	
		One	More than one
# Subjects per object	One	Single object study	Multi-object variation study
	More than one	Multi-test within object study	Blocked subject-object study

- H_{0Eff} : $\mu_{EffSt1} = \mu_{EffSt2}$ the Strategy1 finds on average the same number of faults, i.e., the effectiveness, as the Strategy2, where μ denotes the average percentage of the killed mutants using the complete test suites generated by the two strategies;
- H_{0Size} : $\mu_{NSizeSt1} = \mu_{NSizeSt2}$ the size of test suite *is equal* for strategy1 and strategy2;
- H_{0APFD} : $\mu_{APFDSt1} = \mu_{APFDSt2}$ the average APFD *is equal* for strategy1 and strategy2.

A null hypothesis states that there are no real underlying trends or patterns in the experiment setting; the only reasons for differences in the observations are coincidental. This is the hypothesis that we wants to reject with a high significance as possible.

When the null hypothesis can be rejected with relatively high confidence, it is possible to formulate an alternative hypothesis, as following:

- H_{1Eff} : $\mu_{EffSt1} \neq \mu_{EffSt2}$ the Strategy1 and Strategy2 find on average a different number of faults, i.e. their effectiveness are *Not equal*;
- H_{1Size} : $\mu_{SizeSt1} \neq \mu_{SizeSt2}$ the size of test suite *is Not equal* for strategy1 and strategy2;
- H_{1APFD} : $\mu_{APFDSt1} \neq \mu_{APFDSt2}$ the average APFD *is Not equal* for strategy1 and strategy2.

Variables Selection. By referring to the PDP testing example, the unique *independent variable* is the test case generation strategy with two levels or alternatives (treatments) for the main factor: {Multiple and XACMET}. The *dependent variables* are the Effectiveness, the Size of the test suites and the APFD metrics.

The *object* of the experiment a complex object composed by SunPDP (Gold PDP) and some mutated versions of it. Because mutation techniques are considered in the experiment, the *mutation generator* can be identified as a possible *Parameter*.

Selection of Subjects. The selection of subjects is important when conducting an experiment, because closely connected to the generalization of the results from the experiment. In order to generalize the results to the desired population, the selection must be representative for that population, thus, it is also called a sample from a population. In the example considered the XACML Policies are the Subjects.

Experiment Design. The design we use is the paired comparison design , a particular kind of one factor with two treatments (Wohlin et al., 2012). The same design is called “randomized paired comparison design: two alternatives on one experimental unit” in (Juzgado and Moreno, 2001). In this design, each subject uses both treatments on the same object, i.e., both Strategies are applied to each XACML policy and the obtained test suites are evaluated using the SunPDP and its mutants. This design allows to compare the two treatments (Multiple and XACMET strategies) against each other; the most common operation is to compare the means of the dependent variable (Effectiveness, Size and APFD) for each treatment. In particular, both Strategies are applied to each XACML policy.

Instrumentation. The overall goal of the instrumentation is to provide means for performing the experiment and to monitor it, without affecting the control of the experiment. If the instrumentation affects the outcome of the experiment, the results are invalid.

In the planning of an experiment, the instruments are chosen. Before the execution, the instruments are developed for the specific experiment. The instruments for an experiment are of three types, namely objects, guidelines and measurement instruments.

In the example considered the *Object* is the Sun PDP.

Usually in a testing experiment the number and the nature of faults in the testing objects is known in advance. In the experiment considered the mutation technique can be applied to the Object for deriving a controlled number of faulty versions of the Sun PDP. In this case for instantiating the *mutation generator* parameter, different levels can be considered, such as μJava) (seung Ma et al., 2005), Javalanche (Schuler and Zeller, 2009), Major (Just, 2014) or Judy (Madeyski and Radyk, 2010), that may influence the result of the experiment.

Guidelines and Measurement. Guidelines are procedural steps for executing the Controlled Experiment. They include process descriptions, checklists, tools and facilities useful for performing measurement and enabling the result analysis and interpretation. Among the available proposals for automating the overall testing process of AC, in this paper we use the solution provided in (Daoudagh et al., 2019a), so as to be compliant with the selected research goal (Goal 2 of Table 5). The selected framework enables the collection of the target measures, i.e., Effectiveness, Size and APFD as reported in Table 5.

6 EXPERIMENT OPERATION

The third activity of the experimental process is Operation, which consists of three steps: preparation, execution and data validation. Figure 3 reports the activity diagram of the experiment operation phase considering the Sun PDP Testing of the Goal 2 of Table 5.

As in the Figure 3, during the preparation step, subjects, the object and parameters are instantiated on the selected Testing Framework. In particular during the activities Ⓐ,Ⓑ and Ⓒ the following steps are performed: the Subject represented by XACML policies, are selected; the treatments, i.e., the test case generation strategies (Multiple and XACMET in the example)and the Object of the experiment, i.e., the Sun PDP, are defined. Then, the test cases (i.e., the XACML requests) and the required mutants (i.e., the mutated version of SunPDP) are derived during the activities Ⓓ, Ⓔ and Ⓕ. Afterwards, the execution step, which consists of XACML requests evaluation, data collection, and measures computation, is in charge of activities Ⓖ and Ⓗ. Finally, data validation consists of data selection, filtering and measures computing, i.e., the calculation of Size, Effectiveness and APFD metrics that are in charge of activities Ⓘ of Figure 3.

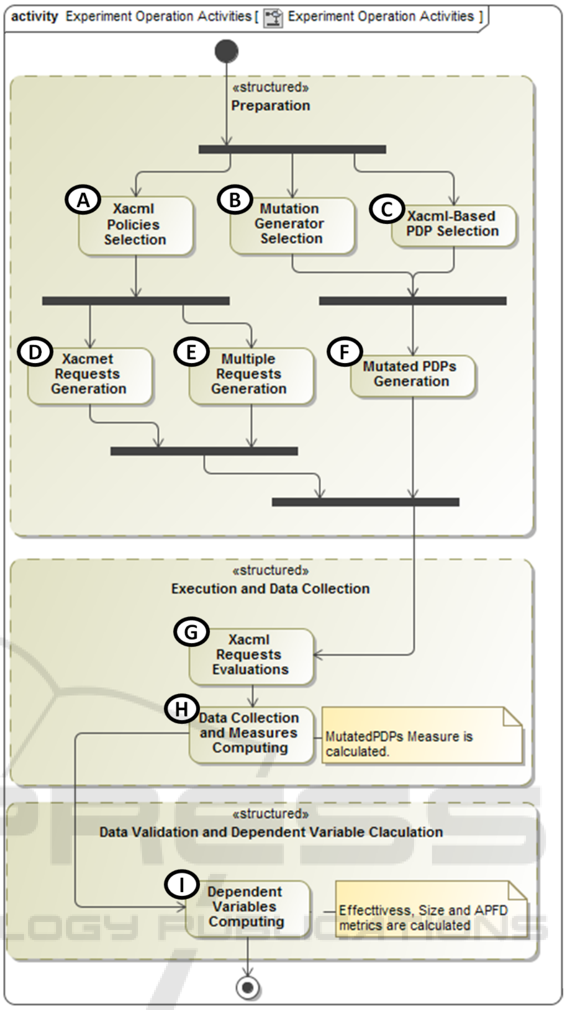
activity Experiment Operation Activities [Experiment Operation Activities]
«structured»
Preparation
A
Xacml Policies Selection
B
Mutation Generator Selection
C
Xacml-Based PDP Selection
D
Xacmet Requests Generation
E
Multiple Requests Generation
F
Mutated PDPs Generation
«structured»
Execution and Data Collection
G
Xacml Requests Evaluations
H
Data Collection and Measures Computing
MutatedPDPs Measure is calculated.
«structured»
Data Validation and Dependent Variable Claculation
I
Dependent Variables Computing
Effecttivess, Size and APFD metrics are calculated

Figure 3: Experiment Operation Activities.

Once the experimental data will be collected the Null Hypothesis test could be performed. In the considered experiment this can be translated into the H_{1Eff}, H_{1Size} and H_{1APFD} defined in Section 5. According to (Juzgado and Moreno, 2001; Wohlin et al., 2012), we can apply the Paired T-Test to formally verify the Null Hypothesis with the confidence level of 95%. This choice was a natural consequence of the type of design adopted, i.e., the paired comparison. Therefore, following the standard best practices, we can accept a probability of 5% of committing a Type-1-Error (Juzgado and Moreno, 2001; Wohlin et al., 2012), i.e., the Null Hypothesis is rejected if the computed p-value is less or equal to 0,05 ($alpha = 0.05$).

7 CONCLUSIONS

In this paper we presented a family of controlled experiments in the context of AC testing. The idea was to define a set of standardized guidelines for correctly and systematically performing the testing process in order to avoid errors and improve the effectiveness of the validation. The proposal relies on a characterization of the first three steps of the experiment process (i.e., Scoping, Planning and Operation) by leveraging the Goal-Question-Metric template. Thus, we detailed the activities necessary for performing the first three steps of the experiment process (i.e., Scoping, Planning and Operation). The example of the testing of the Sun PDP engine is taken as a reference for better explaining the three phases.

It was out of the scope of the paper providing the complete list of testing goals or the realization of all the possible testing frameworks. The example provided in the paper wanted to highlight the peculiarity of the Controlled Experiments and the potentiality they represent for the testing activity.

As a future work we intent to provide other implementations of the Controlled Experiments for different testing purposes, so as to demonstrate its flexibility and adaptability. We want also to apply the proposed Controlled Experiment in real environments so as to collect testing results and perform statistical analysis.

ACKNOWLEDGEMENTS

This work is partially supported by CyberSec4Europe Grant agreement ID: 830929.

REFERENCES

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of Software Engineering*. Wiley.

Basili, V. R. and Rombach, H. D. (1988). The tame project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, 14(6):758–773.

Bertolino, A., Daoudagh, S., Lonetti, F., and Marchetti, E. (2018). An automated model-based test oracle for access control systems. In *Proceedings of the 13th International Workshop on Automation of Software Test*, AST '18, pages 2–8, New York, NY, USA. ACM.

Bertolino, A., Daoudagh, S., Lonetti, F., Marchetti, E., Martinelli, F., and Mori, P. (2014a). Testing of polpa-based usage control systems. *Software Quality Journal*, 22(2):241–271.

Bertolino, A., Daoudagh, S., Lonetti, F., Marchetti, E., and Schilders, L. (2013). Automated testing of extensible access control markup language-based access control systems. *IET Software*, 7(4):203–212.

Bertolino, A., Le Traon, Y., Lonetti, F., Marchetti, E., and Mouelhi, T. (2014b). Coverage-based test cases selection for xacml policies. In *Proceedings of ICST Workshops*, pages 12–21.

Bertolino, A., Lonetti, F., and Marchetti, E. (2010). Systematic XACML Request Generation for Testing Purposes. In *Proc. of 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, pages 3 –11.

Calabrò, A., Lonetti, F., and Marchetti, E. (2017). Access control policy coverage assessment through monitoring. In *Proc. of TELERISE*, pages 373–383.

Daoudagh, S., El Kateb, D., Lonetti, F., Marchetti, E., and Mouelhi, T. (2015). A toolchain for model-based design and testing of access control systems. In *Proc.of MODELSWARD*, pages 411–418. IEEE.

Daoudagh, S., Lonetti, F., and Marchetti, E. (2019a). A framework for the validation of access control systems. In Saracino, A. and Mori, P., editors, *Proceedings of the 2nd International Workshop on Emerging Technologies for Authorization and Authentication*.

Daoudagh, S., Lonetti, F., and Marchetti, E. (2019b). XACMET: XACML modeling & testing an automated model-based testing solution for access control systems. *Software Quality Journal*.

Daoudagh, S., Lonetti, F., and Marchetti, E. (2020). Assessing testing strategies for access control systems: A controlled experiment. In *Proceedings of ICISSP 2020, Valletta, Malta, February 25-27, 2020*.

Harumi, T. et al. (2016). D-mbtdd: An approach for reusing test artefacts in evolving systems. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE.

Hu, V. C., Kuhn, R., and Yaga, D. (2017). Verification and test methods for access control policies/models. *NIST Special Publication*, 800:192.

Islam, S. and Falcarin, P. (2011). Measuring security requirements for software security. In *2011 IEEE*

10th International Conference on Cybernetic Intelligent Systems (CIS), pages 70–75.

Judd, C. M., Smith, E. R., and Kidder, L. H. (1991). Research methods in social relations, fort worth: Holt, rinehart and winston.

Just, R. (2014). The major mutation framework: Efficient and scalable mutation analysis for java. In *Proceedings of the 2014 international symposium on software testing and analysis*, pages 433–436. ACM.

Juzgado, N. J. and Moreno, A. M. (2001). *Basics of software engineering experimentation*. Kluwer.

Kassou, M. and Kjiri, L. (2013). A goal question metric approach for evaluating security in a service oriented architecture context. *CoRR*, abs/1304.0589.

Ko, A. J., Latoza, T. D., and Burnett, M. M. (2015). A practical guide to controlled experiments of software engineering tools with human participants. *Empirical Softw. Engg.*, 20(1):110–141.

Lonetti, F. and Marchetti, E. (2018). On-line tracing of xacml-based policy coverage criteria. *IET Software*, 12(6):480–488.

Madeyski, L. and Radyk, N. (2010). Judy - a mutation testing tool for java. *IET Software*, 4(1):32–42.

Nanthaamornphong, A. and Carver, J. C. (2017). Test-driven development in scientific software: a survey. *Software Quality Journal*, 25(2):343–372.

OASIS (2013). eXtensible Access Control Markup Language (XACML) Version 3.0. `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html`.

Sadeghi, A. and Mirian-Hosseinabadi, S.-H. (2012). Mbtdd: Model based test driven development. *International Journal of Software Engineering and Knowledge Engineering*, 22(08):1085–1102.

Schuler, D. and Zeller, A. (2009). Javalanche: Efficient mutation testing for java. In *Proceedings of ESEC/FSE*, pages 297–298, New York, NY, USA. ACM.

seung Ma, Y., Offutt, J., and Kwon, Y. R. (2005). Mujava : An automated class mutation system. *Journal of Software Testing, Verification and Reliability*, 15:97–133.

Sjøberg, D. I., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N.-K., and Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *IEEE transactions on software engineering*, 31(9):733–753.

Sun Microsystems (2006). Sun's XACML Implementation. `http://sunxacml.sourceforge.net/`.

Utting, M., Pretschner, A., and Legeard, B. (2012). A taxonomy of model-based testing approaches. *Software Testing, Verification and Reliability*, 22(5):297–312.

Weldehawaryat, G. K. and Katt, B. (2018). Towards a quantitative approach for security assurance metrics. In *The Twelfth International Conference on Emerging Security Information, Systems and Technologies; SECURWARE 2018 September 16, 2018 to September 20, 2018-Venice, Italy*. International Academy, Research and Industry Association (IARIA).

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., and Regnell, B. (2012). *Experimentation in Software Engineering*. Springer.

Yahya, F., Walters, R. J., and Wills, G. B. (2015). Using goal-question-metric (gqm) approach to assess security in cloud storage. In *International Workshop on Enterprise Security*, pages 223–240. Springer.

Yoo, S. and Harman, M. (2012). Regression testing minimization, selection and prioritization: A survey. *Softw. Test. Verif. Reliab.*, 22(2):67–120.