

200 РОКІВ
ОСВІТНІХ ТРАДИЦІЙ



Том 2

**ТЕЗИ
70-ої наукової конференції
професорів, викладачів, наукових
працівників, аспірантів та студентів університету**

**ПОЛТАВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЮРІЯ КОНДРАТЮКА**

Міністерство освіти і науки України
Північно-Східний науковий центр НАН України та МОН України
Полтавський національний технічний університет
імені Юрія Кондратюка

Тези

70-ої наукової конференції професорів,
викладачів, наукових працівників, аспірантів
та студентів університету

Том 2

23 квітня – 18 травня 2018 р.

Полтава 2018

УДК 043.2
ББК 448лО

*Розповсюдження та тиражування без офіційного дозволу
Полтавського національного технічного університету
імені Юрія Кондратюка заборонено*

Редакційна колегія:

- Онищенко В.О. д.е.н., проф., ректор Полтавського національного
технічного університету імені Юрія Кондратюка
- Сівіцька С.П. к.е.н., доц., проректор з наукової та міжнародної роботи
- Гришко В.В. д.е.н., проф., директор навчально-наукового інституту
фінансів, економіки та менеджменту
- Іваницька І.О. к.х.н., доц., декан гуманітарного факультету
- Нестеренко М.П. д.т.н., проф., декан будівельного факультету
- Матвієнко А.М. к.т.н., доц., заступник директора навчально-наукового
інституту нафти і газу
- Муравльов В.В. к.т.н., доц., в.о. декана архітектурного факультету
- Шульга О.В. д.т.н., доц., директор навчально-наукового інституту
інформаційних технологій та механотроніки

Тези 70-ої ювілейної наукової конференції професорів, викладачів,
наукових працівників, аспірантів та студентів університету. Том 2.
(Полтава, 23 квітня – 18 травня 2018 р.) – Полтава: ПолтНТУ, 2018. – 380 с.

У збірнику тез висвітлені результати наукових досліджень
професорів, викладачів, наукових працівників, аспірантів та студентів
університету.

©Полтавський національний технічний
університет імені Юрія Кондратюка,
2018

можуть заважати зміни коду, а не сприяти йому. Ситуація не така вже й рідкісна, а тому й далеко не остання в списку неприємностей.

Але найгірше те, що навіть тести, успішно виконані й перевиконані після внесення усіляких змін, не дають ніякої гарантії, що в коді немає помилок. І справа тут навіть не в підгонці тестів під код, а в людському факторі. Іноді дещо змінивши значення вхідних параметрів в рамках тестів програміст може, сам того не бажаючи, пропустити помилку в тестованому коді або, гірше того, внести її туди. Але, від помилок, пов'язаних з людським фактором, не застрахований ніхто і ніколи.

Вище я згадував, що є спеціальні інструменти, які допомагають програмісту писати unit-тести, нижче наведено основні із них.

Назви більшості популярних фреймворків для unit-тестування утворені від назви мови, на якій пишеться програмний код, і слова "unit". Наприклад, для Java це JUnit, для .NET'овських мов - NUnit, для Delphi - DUnit, для C ++ - CPPUnit, для PHP - PHPUnit, для Python - PyUnit, для ActionScript - AsUnit ... Ну і так далі. А якщо не знаходиться фреймворк для unit-тестування для тієї мови, на якій пишете, швидше за все, ця мова або ще не придумали, або вже ніде не використовують, або це Асемблер.

Що ж, як бачите, в самій ідеї unit-тестування немає нічого складного. Та й в практиці, мабуть, теж. Потрібно тільки знайти час та сили на вивчення тестіровочного фреймворка для використовуваного вами мови програмування і для того, щоб писати самі тести.

Література

1. KV.by [Електронний ресурс].– Режим доступу: <https://www.kv.by/archive/index2009421107.htm>.

2. Про Тестинг - Тестирование Программного Обеспечения [Електронний ресурс].– Режим доступу: <http://www.protesting.ru/testing/levels/component.html>.

3. Хабрахабр [Електронний ресурс].– Режим доступу: <https://habrahabr.ru/post/169381/>.

ПРИЗНАЧЕННЯ ТЕСТУВАННЯ БЕЗПЕКИ. ВИДИ УРАЗЛИВОСТЕЙ

Тестування програмного забезпечення – оцінка розроблюваного програмного забезпечення / продукту, з метою перевірити його можливості та відповідність очікуваним результатам. Тестування програмного забезпечення є невід'ємною частиною циклу розробки програмного забезпечення [1].

Комп'ютерні системи дуже часто є мішенню незаконного проникнення. Під проникненням розуміється широкий діапазон дій: спроби хакерів проникнути в систему з спортивного інтересу, помста розгніваних службовців, злом шахраями для незаконної наживи.

Тестування безпеки перевіряє фактичну реакцію захисних механізмів, вбудованих в систему, на проникнення. В ході тестування безпеки випробувач грає роль зломщика. Йому дозволено все:

- спроби дізнатися пароль за допомогою зовнішніх засобів;
- атака системи за допомогою спеціальних утиліт, які аналізують захист;
- придушення системи (в надії, що вона відмовиться обслуговувати інших клієнтів);
- цілеспрямоване введення помилок в надії проникнути в систему в ході відновлення;
- перегляд несекретних даних в надії знайти ключ для входу в систему.

Звичайно, при необмеженому часі й ресурсах хороше тестування безпеки зламає будь-яку систему. Завдання проектувальника систем – зробити ціну проникнення вищою, ніж ціна одержуваної в результаті інформації.

Основним завданням системного тестування є перевірка як функціональних, так й не функціональних вимог в системі в цілому. При цьому виявляються дефекти, такі як неправильне використання ресурсів системи, непередбачені комбінації даних, несумісність з оточенням, непередбачені сценарії використання, відсутня або неправильна функціональність, незручність використання та ін. Для мінімізації ризиків, пов'язаних з особливостями поведінки в системі в тому чи іншому середовищі, під час тестування рекомендується використовувати оточення максимально наближене до того, на яке буде встановлено продукт після видачі.

В даний час найбільш поширеними видами уразливості в безпеці програмного забезпечення є [2]:

XSS (Cross-Site Scripting) – вид уразливості програмного забезпечення (Web додатків), при якій, на генерованій сервером сторінці, виконуються шкідливі скрипти, з метою атаки клієнта. Самі по собі XSS атаки можуть бути дуже різноманітними. Зловмисники можуть спробувати вкрасти куки, перенаправити на сайт, де відбудеться більш серйозна атака, завантажити в пам'ять будь-який шкідливий об'єкт та ін., всього навсього розмістивши шкідливий скрипт на сайті.

XSRF / CSRF (Request Forgery) – вид уразливості, що дозволяє використовувати недоліки HTTP протоколу, при цьому зловмисники працюють за такою схемою: посилання на шкідливий сайт встановлюється на сторінці, що користується довірою у користувача, при переході по шкідливому посиланні виконується скрипт, який зберігає особисті дані користувача (паролі, платіжні дані та ін.), або відправляє СПАМ повідомлення від особи користувача, або змінює доступ до облікового запису користувача, для отримання повного контролю над нею.

Code injections (SQL, PHP, ASP i m.д.) – вид уразливості, при якому стає можливо здійснити запуск виконуваного коду з метою отримання доступу до системних ресурсів, несанкціонованого доступу до даних або виведення системи з ладу.

Server-Side Includes (SSI) Injection – вид уразливості, що використовує вставку серверних команд в HTML код або запуск їх безпосередньо з сервера.

Authorization Bypass – вид уразливості, при якому можливо отримати несанкціонований доступ до облікового запису або документам іншого користувача. Користувач А може отримати доступ до документів користувача Б. Припустимо, є реалізація, де при перегляді свого профілю, що містить конфіденційну інформацію, в URL сторінки передається userID, а даному випадку є сенс спробувати підставити замість свого userID номер іншого користувача. І якщо ви побачите його дані, значить ви знайшли дефект.

Прикладів вразливостей й атак існує величезна кількість. Навіть провівши повний цикл тестування безпеки, не можна бути на 100% впевненим, що система по-справжньому безпечна. Але можна бути впевненим в тому, що відсоток несанкціонований проникнень, крадіжок інформації, втрат даних буде на кілька порядків нижче, ніж у тих хто не проводив тестування безпеки [3].

Література

1. Блог Web программіста [Електронний ресурс] – Режим доступу: <http://juice-health.ru/program/software-testing/495-software-testing-methods>
2. Getbug [Електронний ресурс] – Режим доступу: <http://getbug.ru/testirovanie-bezopasnosti-saytov-i-prilozheniy/>
3. Тестирование безопасности [Електронний ресурс] – Режим доступу: <https://poisk-ru.ru/s11582t2.html>

<i>І.В. Ромашко</i> ВИКОРИСТАННЯ ПРОТОКОЛУ VTR ДЛЯ МАСШТАБУВАННЯ ЛОКАЛЬНИХ МЕРЕЖ	160
<i>І.В. Ромашко</i> ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ АГРЕГУВАННЯ КАНАЛІВ ДЛЯ ПІДВИЩЕННЯ ПРОПУСКНОЇ ЗДАТНОСТІ МЕРЕЖІ.....	160
<i>І.Я. Гудзенко, Сокол Г. В.</i> ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАВАДОСТІЙКОГО КОДУВАННЯ НА ОСНОВІ C++.....	161
<i>В.А. Василевська, Часовських І.С., Г.В. Сокол, Т.В. Буряк</i> WI-FI РАДІО З ІНТЕРАКТИВНИМ БЕЗДРОТОВИМ УПРАВЛІННЯМ.....	162
<i>А.В. Виноградова, Г.В. Сокол, Т.В. Буряк</i> АНАЛІЗ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ РОБОТИЗОВАНИХ КОМПЛЕКСІВ	164
<i>В.Ю. Литвиненко, Г.В. Сокол</i> ПОРІВНЯЛЬНИЙ АНАЛІЗ ЛІЦЕНЗІЙНИХ ТА БЕЗКОШТОВНИХ КОДЕКІВ ДЛЯ ОБРОБКИ БАГАТОВИМІРНИХ СИГНАЛІВ	166
<i>О.В. Мосієнко, Г.В. Сокол</i> АНАЛІЗ АУДІОПЛЕСРІВ ОБРОБКИ ОДНОВИМІРНИХ СИГНАЛІВ.....	168
<i>В.Р. Ткаченко, Г.В. Сокол</i> АНАЛІЗ ПРОБЛЕМ МОНІТОРИНГУ СИСТЕМ SMART HOUSE	169
СЕКЦІЯ КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В СИСТЕМ	
<i>Ляхов О.Л., Демиденко М.І., Фурсова Н.А.</i> АРХІТЕКТУРА РОЗПОДІЛЕНОЇ АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КРУВАННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ ВНЗ	170
<i>С.П. Альошин, О.О. Бородіна</i> НЕЙРОМЕРЕЖЕВИЙ ПРЕДИКТИВНИЙ МЕТОД ОПТИМІЗАЦІЇ В ЗАДАЧІ БАГАТОФАКТОРНОГО АНАЛІЗУ	173
<i>С.О. Зайка, А.Т. Лобурець</i> РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗРАХУНКУ ДІАГРАМ СТАНУ ТЕРМОДИНАМІЧНИХ СИСТЕМ	176
<i>О.А. Руденко, М.І. Демиденко, А.А. Швидкий</i> ПРОГРАМНИЙ МОДУЛЬ «ОБЛІК УСПІШНОСТІ» АВТОМАТИЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ	178
<i>Гайтан О.М., Горошко А.І.</i> АНАЛІЗ СИСТЕМ ПЕРЕВІРКИ НАУКОВИХ ТА АКАДЕМІЧНИХ ТЕКСТІВ НА ОРИГІНАЛЬНІСТЬ.....	180

<i>О.О. Бородіна, Д.С.Цюман, А. Шабанова, О.В.Куц</i> ПІДГОТОВКА ТЕКСТІВ ДЛЯ ПЕРЕКЛАДУ (КОНВЕРТУВАННЯ ТА НОРМАЛІЗАЦІЯ)	183
<i>О.О. Бородіна, М. М. Філонич</i> SELENIUM.АВТОМАТИЗОВАНЕ ТЕСТУВАННЯ WEB ДОДАТКІВ	185
<i>О.О. Бородіна, М. М. Філонич</i> ПРОДУКТИВНІСТЬ ТА ПОМИЛКИ АНАЛІЗУ ЧАСОВИХ РЯДІВ ...	187
<i>О.О. Бородіна, А.О. Горошко</i> UNIT ТЕСТУВАННЯ ЯК СПОСІБ ВИЯВИТИ СЛАБКІ МІСЦЯ В ANDROID ПРОГРАМІ.....	189
<i>О.О. Бородіна, В. М. Фіней</i> ПРИЗНАЧЕННЯ ТЕСТУВАННЯ БЕЗПЕКИ. ВИДИ УРАЗЛИВОСТЕЙ	191
<i>О.О. Бородіна, В. М. Фіней</i> ЕКСПЕРТНІ СИСТЕМИ	193
<i>О.О. Бородіна, Д.О. Клименко</i> ШТУЧНИЙ ІНТЕЛЕКТ ЯК РЕВОЛЮЦІЯ В МЕДИЦИНІ.....	196
<i>О.О. Бородіна, Д.М. Кривицкий</i> КОМ'ПЮТЕРНИЙ ЗІР	198
<i>О.О. Бородіна, В.О. Величко</i> ШТУЧНИЙ ІНТЕЛЕКТ В МАРКЕТИНГУ ТА ЙОГО ВИКОРИСТАННЯ	200
<i>О.О. Бородіна, В.О. Величко</i> МОБІЛЬНЕ ТЕСТУВАННЯ.....	202
<i>О.О. Бородіна, Д.О. Клименко</i> ПРОБЛЕМА ВИБОРУ МЕТОДОЛОГІЇ ТЕСТУВАННЯ ПЗ	204
<i>М.І. Демиденко, Сузима І.Ю.</i> «РЕДАКТОР РОЗКЛАДУ ЗАНЯТЬ У ВНЗ».....	206
<i>Горошко А., Демиденко М.І.</i> ПРОГРАМНО - АПАРАТНИЙ КОМПЛЕКС ДЛЯ ВИЗНАЧЕННЯ ЗОРОВОЇ ВТОМИ ЛЮДИНИ	208
<i>А.М.Гафіяк, М. Мизюра, Віктор Гусак, Володимир Гусак, С.Х. Хосейні</i> РОЗРОБКА КЛІЄНТ-РОЗКЛАДУ ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ	210
<i>А.М.Гафіяк, А.А.Гаврилишин</i> МЕТОДОЛОГІЯ ЕКОНОМІКИ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ ..	212
<i>А.М.Гафіяк, А.С. Кікоть</i> СУЧАСНІ РЕФОРМИ ЗАКОНОДАВСТВА В ІТ-СФЕРІ УКРАЇНИ ...	214
<i>А.М.Гафіяк, М.Г. Колтунов</i> ПРОБЛЕМИ ІНФОРМАЦІЙНОГО СУСПІЛЬСТВА	216
<i>А.М.Гафіяк, В.В. Кузнецов</i> ВЗАЄМОЗВ'ЯЗОК ЕКОНОМІЧНОЇ ІНФОРМАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ.....	218