Юрій Яремчук
Вінницький національний технічний університет
УДК 681.3.067

ЗАХИЩЕНА ВЕБ-СИСТЕМА ТЕСТУВАННЯ ЯКОСТІ ЗНАНЬ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ КРИПТОГРАФІЧНИХ ТЕХНОЛОГІЙ НА ОСНОВІ РЕКУРЕНТНИХ ПОСЛІДОВНОСТЕЙ

Анотація: **Розроблено систему тестування якості знань з використанням крос-платформних рішень та веб-технологій, яка забезпечує надійний захист тестування та має гнучку систему доступу кандидатів до системи тестування без прив'язування до конкретної платформи чи операційної системи. Представлено структурну схему захищеної веб-системи тестування, а також взаємозв'язки між її компонентами. Розглянуто особливості розробки кожного програмного модуля веб-системи тестування, а також розроблено механізми захисту даних на різних рівнях їх використання у системі захисту. При цьому усі необхідні технології шифрування, ключового обміну, автентифікації та цифрового підписування розроблено на основі власних методів, що базуються на єдиному математичному апараті рекурентних V_k –послідовностей. Це дозволило не лише підвищити рівень безпеки системи тестування, але й підвищити швидкість виконання в ній криптографічних перетворень.**

Summary: **We have developed a testing system of knowledge quality, using cross-platform solutions and web-technologies, that provides reliable testing protection and has a flexible access system of candidates to the testing system, without linking to a particular platform or operating system. We prepared a structural scheme of the secure web-based testing system and relationship between its components. We considered development features of each software module of the web-based testing system, and developed mechanisms to protect data at different levels of their use in the protection system. At the same time, all necessary technologies of encryption, key exchange, authentication and digital signature were developed based on own methods, that are based on a single mathematical apparatus of recurrent V_k sequences. This allowed us not only to improve the testing system security, but also to increase its speed of cryptographic transformations.**

Ключові слова: **Криптографія, шифрування, автентифікація, цифрове підписування, рекурентні послідовності, тестування якості знань, веб-система**

I Вступ

Найбільш сучасним методом контролю знань є тестування [1]. Це пов'язано з тим, що поширення систем дистанційного навчання [2–6] потребує якісного контролю знань. Всі міжнародні компанії та найвідоміші університети давно перейшли до такого виду контролю знань. Це дозволило досить просто, швидко та об'єктивно оцінювати знання, оскільки в тестуванні присутні різноманітні комплексні завдання, які можуть в собі поєднувати як теоретичний так і практичний матеріал.

Одним із суттєвих недоліків сучасних комп'ютерних систем тестування [1, 7, 8] є прив'язка клієнтського додатку до платформи та операційної системи, під які вона реалізується, що тим самим обмежує клієнтські додатки щодо програмного середовища використання. Хоч на сьогоднішній день існує всього три найбільш розповсюджені операційні системи, такі як Windows, MacOS та Linux, але у кожної з цих систем є як мінімум чотири версії, а в останньої їх більше сотні, тому підтримка та розробка універсального програмного продукту є досить проблематичною та потребує дуже великих ресурсів.

Окрім цього існує проблема безпеки систем тестування, оскільки кожна операційна система має свої власні вразливості, які можуть впливати на роботу програмного продукту. Як наслідок, зловмисник може отримати доступ до системи тестування та впливати на її роботу.

Проблема забезпечення безпеки системи тестування є дуже актуальною, оскільки під час контролю якості знань здійснюється зберігання і обробка бази з тестами та правильними відповідями, а також результатів тестування кандидатів, тому порушення конфіденційності та цілісності цих баз та результатів може звести нанівець весь процес тестування знань.

Враховуючи вищесказане, актуальною є розробка такої системи тестування, яка б забезпечувала надійний захист тестування і про цьому мала гнучку систему доступу кандидатів до системи тестування без прив'язування до конкретної платформи чи операційної системи.

ІІ Розробка захищеної веб-системи тестування

Для розробки системи тестування пропонується використати крос-платформні рішення [9], які мають широкі можливості щодо побудови захищених систем з високим рівнем безпеки, що значно підвищить загальну безпеку процесу тестування та унеможливить виникнення неконтрольованого втручання. Також крос-платформні рішення дозволяють будувати гнучкі системи з можливістю легкого та швидкого перенесення середовища тестування з одного сервера на інший і при цьому не зважати на саму операційну систему (Windows, Linux, Unix), її версію та середовище обробки веб-сторінок (IIS, Apatch). Саме таке рішення повинно забезпечити мобільність та безпечність систем тестування, що, в свою чергу, приведе до підвищення якості оцінювання знань в цілому.

За останні декілька років на ринку розробки програмного забезпечення кількість замовлень на розробку програм під конкретну операційну систему суттєво зменшалась, в той час як кількість замовлень на розробку мобільних програм значно збільшилась. Обумовлено це позитивними тенденціями збільшення доступності та якості надання послуг доступу до глобальної мережі Інтернет, оскільки кожна операційна система має можливість підключення до системи Інтернет та спеціалізоване програмне забезпечення, відоме як браузер, для зручного перегляду різноманітного контенту веб-сайтів. Враховуючі це, розробку системи тестування будемо здійснювати з використанням веб-технологій, що дозволить використовувати її на будь-якому пристрої, який буде підключений до глобальної мережі Інтернет або локальної мережі з використанням власного веб-серверу.

При розробці системи тестування особливу увагу слід приділити забезпеченню захисту всього процесу тестування. Для цього необхідно, в першу чергу, контролювати користувачів, що використовують систему тестування, контролювати пристрої, через які підключаються користувачі, забезпечувати безпечне передавання даних між сервером та браузером користувача, зберігати інформацію та обмежувати до неї доступ. Саме тому при розробці системи тестування значну увагу слід приділити побудові системи захисту усіх її компонентів та розглядати весь механізм як комплексну систему захисту.

Проаналізуємо проблеми, основні компоненти (модулі) системи, які можуть піддаватись атакам з боку зловмисника, та визначимо можливі заходи протидії цим атакам.

Для забезпечення безпеки системи тестування перш за все слід забезпечити захист даних під час їх передавання та зберігання від можливих загроз щодо порушення конфіденційності, цілісності та доступності. Розроблено досить багато механізмів, які дозволяють вирішити ці проблеми в різноманітних системах, але в більшості випадків до кожної системи існують свої якісь особливі вимоги. Для системи тестування, що розробляється, можна виділити такі важливі вимоги: надійність зберігання даних, безпека їх передавання через незахищену мережу, швидкодія веб-додатку.

Другою проблемою, яку необхідно вирішити, є можливість використання зловмисником спроби авторизуватись на сайті (в системі тестування) та отримати доступ до якихось внутрішніх елементів системи. Якщо спроба несанкціонованого доступу зловмисника стане вдалою, то він зможе почати досліджувати систему тестування із середини і при виявленні якихось недоліків у системі пошкодити її цілісність. Для запобігання таким діям можна запропонувати декілька програмних та організаційних заходів, а саме:

– обмежена (контрольована) реєстрація на сайті;

– перед проходженням тестів адміністратор повинен підтвердити користувача та його робоче місце;

– обмежити периметр мережі, тобто надавати доступ до системи тестування тільки в рамках контрольованої зони.

Обмеження реєстрації на сайті дозволить значно ускладнити процес доступу для зловмисника. Наприклад, для реєстрації на сайті системи тестування кожен студент повинен використовувати електронну пошту університету та вказувати номер студентського квитка. Оскільки інформацію про номер квитка має тільки його власник, то отримання поштової скриньки, не будучи студентом університету, є неможливим, відповідно майже унеможливлюється і неконтрольований доступ до системи.

Також досить дієвим захистом є підтвердження робочих місць учасників тестування. Це дозволяє чітко визначити з якого електронного пристрою буде проходити тест учасник і, у випадку використання мобільних пристроїв, підключених через бездротову мережу Wi-Fi, унеможливить підключення іншого учасника з чужим логіном та паролем.

Під час розробки системи тестування слід також звернути увагу на визначення периметру мережі, в якому буде працювати система. Якщо ніяк не обмежувати доступ і підключити систему до глобальної мережі без обмежень, то виникає досить вірогідна загроза атаки типу відмова в обслуговувані DOS або DDOS [10]. Це може призвести до перенавантаження серверного обладнання та мережі і, як наслідок, поставити під загрозу працездатність всієї системи тестування в цілому. Якщо все ж таки розглядати підключення до глобальної

системи, то слід використовувати спеціалізоване апаратне або програмно-апаратне забезпечення протидії таким атакам, відоме як мережеві екрани.

При використанні системи тестування в обмеженому доступі, скажімо локальній мережі університету, вірогідність подібних атак досить мала, але досить великою є вірогідність миттєвої ідентифікації зловмисника за його робочим місцем, особливо при використанні бездротових мереж, коли створити контрольовану зону стає майже неможливо, бо чіткого обмеження в радіусі дії бездротової мережі не існує, оскільки він залежить від чутливості та потужності приймаючого пристрою. Отже використання такого типу з'єднання не є безпечним, але з точки зору зручності є досить перспективним, саме тому слід використовувати механізм ідентифікації робочих станцій.

Враховуючи вказані проблеми забезпечення безпеки та можливості їх вирішення пропонується розробити систему тестування у вигляді веб-додатку, функціонал якого в основному повинен забезпечувати виконання таких дій: реєстрація та авторизація користувачів, ідентифікація робочої станції користувача, створення та модифікація бази даних тестів, механізм тестування та зберігання результатів, захист даних, що передаються та зберігаються. Для реалізації цього функціоналу необхідно розробити відповідні програмні модулі.

Оскільки такі функціональні дії як авторизація, ідентифікація і захист даних, що передаються та зберігаються, потребує використання криптографічних перетворень (розподіл ключів, шифрування, автентифікація, цифрове підписування), доцільним є створення одного єдиного криптографічного модуля, який буде реалізовувати технології різного криптографічного призначення. При цьому такі модулі як авторизація та ідентифікація робочих станцій слід реалізовувати окремо і пов'язати їх з криптографічним модулем. Незалежна реалізація цих модулів істотно підвищить загальну безпеку системи, бо успішна атака на модуль авторизації або ідентифікації не дозволить зловмиснику отримати повноцінний доступ до веб-системи тестування.

Досить важливим функціоналом при розробці веб-системи тестування є розподілення прав доступу. Це також потребує окремної, незалежної програмної реалізації, бо даний модуль буде досить часто використовуватись і постійно контролювати всі дії користувачів.

Модуль, що безпосередньо реалізує систему тестування, та модуль адміністрування слід розробляти окремо один від одного, оскільки механізм адміністрування в основному є незмінним, в той час як процес тестування може потребувати модифікацій для додавання якихось нових функціональних особливостей.

Враховуючи усі вищеописані модулі, що реалізують весь необхідний функціонал, пропонується веб-система тестування, структурна схема якої з відображенням модулів та взаємодії між ними представлена на рис. 1.

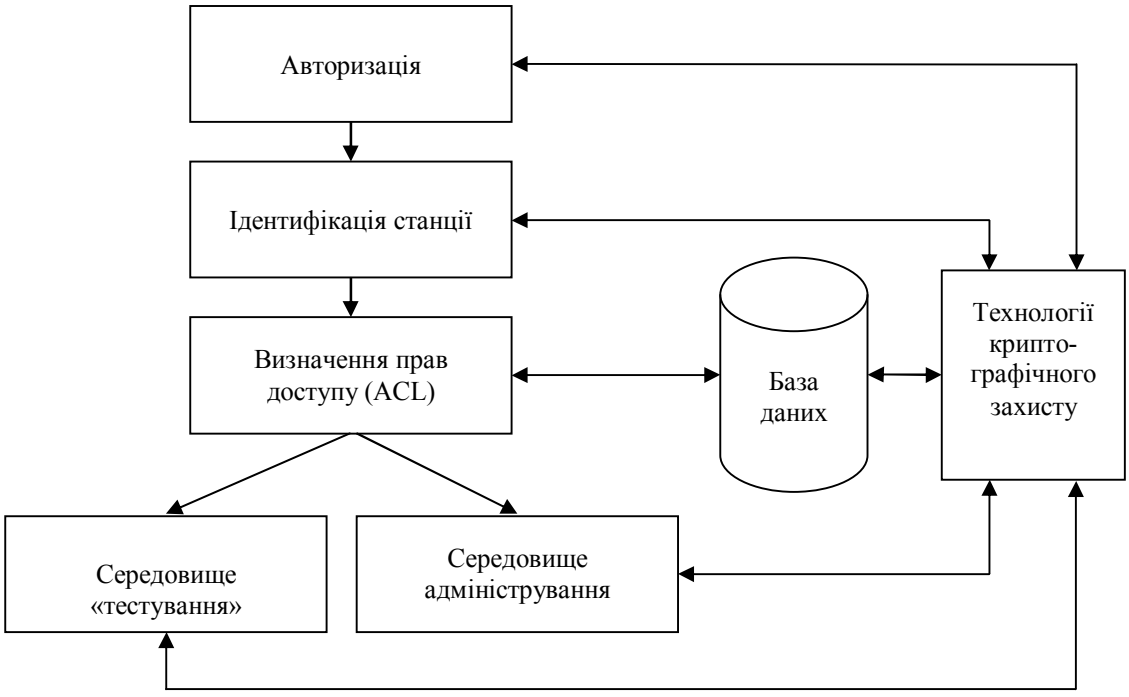

Рисунок 1 – Структурна схема захищеної веб-системи тестування

Робота представленої веб-системи тестування буде здійснюватись таким чином. Спочатку користувач, для того, щоб скористуватись ресурсом системи, повинен пройти авторизацію. Механізм авторизації потребує використання спеціалізованих криптографічних перетворень та перевірки наявності відповідного користувача у базі даних веб-ресурсу. Таким чином, для роботи даного модуля необхідно використовувати певний функціонал криптографічного модуля, який, у свою чергу, взаємодіяв би з базою даних та надавав відповідну інформацію, зв'язки між цими модулями показано на рис. 1. Далі виконується ідентифікація робочої станції користувача, що також передбачає використання криптографічного модуля та взаємодії з базою даних, цю взаємодію відображено на рис. 1.

Після авторизації користувача та ідентифікації його робочої станції необхідно визначити, до яких ресурсів має доступ цей користувач. Це можна реалізувати шляхом перевірки (співставлення) даних ідентифікатора користувача, отриманих попередньо, та відповідних даних у базі даних. Результатом такої перевірки буде певний набір правил, який дозволить визначити, який саме функціонал відображати користувачеві. Відповідний зв'язок показано на рис. 1.

Наступним кроком роботи розроблюваної системи тестування є отримання доступу до панелі тестування чи адміністрування, залежно від визначених попередньо прав доступу. Панелі тестування та адміністрування використовують (відображають та оброблюють) дані, що зберігаються у базі даних. Ці дані мають особливі вимоги до зберігання та обробки, так як вони є конфіденційними і їх цілісність може вплинути як на стабільність роботи системи, так і на якість оцінювання знань. Для того, щоб уникнути неприпустимих дій з цими даними, необхідно забезпечити відповідний рівень безпеки за рахунок використання особливостей (функціоналу) криптографічного модуля для цих даних. Відповідну взаємодію відображено на рис. 1.

Розглянемо тепер більш детально реалізацію кожного модуля представленої веб-системи тестування.

Реалізація модуля авторизації.

Модуль авторизації є основною ланкою підсистеми захисту. Цей модуль необхідний для контроля та обмеження доступу до сайту, він повинен лише авторизованим користувачам давати можливість проходження тестів, а також створення та керування базою питань. Необхідно визначити певне коло осіб, які мають право створювати та редагувати тести. Створення такого модуля в системі тестування дозволяє реалізувати контроль доступу та обмежити права доступу для різних користувачів.

У веб-додатку, що розробляється, пропонується використовувати досить популярну модель керування доступом користувачів, відому як RBAC (Role Based Access Control) [11]. Усі користувачі веб-додатку зберігаються у відповідних таблицях бази даних, у яких вказується роль кожного користувача відповідно. Визначено чотири ролі користувачів, що мають ієрархічну структуру, в якій кожна роль розширює батьківську.

Гість – найменш привілейована роль користувачів. Дана роль надається усім користувачам, що не пройшли процес автентифікації. Для користувачів, що володіють даною роллю, доступна обмежена кількість сторінок веб-додатку, а саме сторінка автентифікації, сторінка з контактною інформацією та сторінка загальних відомостей про веб-додаток. Для користувачів з роллю «гість» немає необхідності в зберіганні їх даних у базі даних. Роль «гість» є базовою і розширюється усіма наступними ролями.

Кандидат – роль, що надається користувачам, яким необхідно мати доступ до проведення тестування. Дана роль успадковує усі права ролі «гість», тому користувачі з роллю «кандидат» мають усі права як у користувачів ролі «гість». Для ролі «кандидат» також визначені додаткові права, серед яких право перегляду доступних тестів, запуск тесту та формування запиту на проходження тесту.

Автор – роль, яка надається користувачам, що конструюють тест. Дана роль успадковує усі права ролей «гість» та «кандидат». Таким чином автор тесту може самостійно пройти тест для перевірки достовірності питань, варіантів відповідей та процесу підрахунку балів. Для ролі «автор» визначені додаткові права. Автори можуть створювати нові тести та редагувати вже існуючі за умови, що вони належать автору. Автор має можливість надавати та обмежувати доступ до здавання тесту, формувати календарні та часові обмеження для запуску тесту. Керування доступом проводиться як на основі окремих облікових записів «кандидатів», так і на основі їхніх груп. Авторам також доступна інформація про результати здавання тесту.

Адміністратор – найбільш привілейована роль. Користувачам, яким надана роль «адміністратор», доступні усі дії ролей «гість», «кандидат» та «автор». Саме користувачі з роллю «адміністратор» можуть проводити керування обліковими записами користувачів та надавати їм відповідні ролі.

Пройшовши процес автентифікації користувачеві надається його відповідна роль. Усі дані авторизації зберігаються у сесії сервера та являють собою маркер доступу також відомий як token access. Маркер доступу містить ідентифікатор користувача, унікальний ідентифікатор робочої станції та ідентифікатор ролі. Маркер доступу перевіряється при кожній дії користувача. Усі дані маркеру зберігаються в шифрованому

вигляді та можуть використовуватись лише на серверній частині веб-додатку, завдяки чому гарантується їх безпека та автентичність.

Реалізація модуля ідентифікації робочої станції.

В усіх існуючих на цей день системах тестування використовуються стандартні механізми захисту, які дозволяють обмежувати доступ на основі облікового запису користувача системи. Але в разі можливості публічного доступу до системи тестування через мережу Інтернет стає актуальною проблема відстеження робочого місця, а точніше пристрою, з якого користувач підключаються до системи, оскільки будь-який користувач може передати власний логін та пароль іншому користувачеві, який може авторизуватись та здати тест замість нього.

Вирішенням даної проблеми може бути здійснення ідентифікації робочої станції користувача та обмеження на використання інших станцій під час тестування. Тут під робочою станцією розуміється будь-який пристрій, що має операційну систему, браузер та підтримує підключення до локальної або глобальної мережі. Для ідентифікації робочої станції будемо використовувати інформацію про версію операційної системи, версію браузеру, унікальний ідентифікатор операційної системи та мережеві параметри робочої станції. Дана інформація формує собою ідентифікатор робочої станції, яка збирається на етапі автентифікації та зберігається в маркері доступу. Ідентифікатори робочих станцій можуть бути проконтрольовані до початку тесту, завдяки чому можна чітко визначити на яких робочих станціях буде здійснюватися тестування. Якщо на будь-якому етапі роботи з веб-додатком буде виявлено невідповідність даних ідентифікатора робочої станції з реальними даними, або дублікатом ідентифікатору, маркер доступу буде знищено. В результаті чого процес тестування для даного кандидату буде припинено.

Реалізація модуля тестування.

Модуль тестування призначений для безпечного передавання питань та відповідей між кандидатом та сервером. Усі питання завантажуються в шифрованому вигляді. Усі відповіді перед відправкою підписуються, використовуючи ідентифікатор робочої станції та ідентифікатор користувача. Модуль тестування контролює дату, час та тривалість проходження тесту. Модуль дає можливість зберігати проміжкові відповіді як на серверній так і на клієнтській частині. Під час конструювання тесту «автор» визначає можливість повернення до попередніх відповідей та їх корегування, яка контролюється модулем. Однією з основних задач модулю тестування є підсумок балів, які отримує «кандидат». Кількість балів формується на основі правил, які визначає автор тесту. Процес оцінювання є гнучким та може бути налаштований під різні форми оцінювання питання. Таким чином можна визначити як вагу всього питання так і окремих відповідей. Розроблено можливість створення питань з однією та декількома правильними відповідями, можливість введення довільних відповідей на питання, а також використання мультимедійних матеріалів в запитанні.

Реалізація модуля технологій криптографічного захисту.

Модуль тестування повинен забезпечувати безпеку як самих даних, що зберігаються в системі, так і їх безпечне передавання між клієнтом та сервером. Це досягається за рахунок використання технологій криптографічного захисту. Веб-додатки мають клієнт-серверну архітектуру, що обумовлює передавання даних між запитами до серверу та відповідями клієнту по мережі. Саме дані запитів та відповідей потребують підвищеного захисту, оскільки передаються фактично відкритим каналом. Усі дані тесту шифруються на стороні серверу та передаються по незахищеному каналу даних у зашифрованому вигляді до клієнта. Після отримання даних клієнт їх дешифрує і відображає користувачеві. Під час проходження тесту усі відповіді на запитання зберігаються локально, на робочій станції користувача, в оперативній пам'яті. Після завершення тестування всі відповіді підписуються електронним цифровим підписом та відправляються на сервер. На стороні серверу проводиться контроль відповідності проміжних даних тесту та фінальних, після чого перевіряється підпис.

Реалізація модуля глобального адміністрування.

Модуль глобального адміністрування призначений для контролю облікових записів користувачів та їх ролей. Доступ до модуля адміністрування облікових записів мають лише користувачі ролі «адміністратор». За допомогою даного модуля можна здійснювати створення, редагування та видалення користувачів. Модуль також надає можливість перегляду інформації про використання ідентифікаторів робочих станцій. В модуль також включено конструктор тестів, до якого мають доступ як «адміністратор», так і автори. Процес створення тестів є дуже гнучким та дозволяє створювати різнотипні тести, а також проводити складні обрахунки підсумкових балів. Модуль надає інформацію про статистику правильних відповідей на питання, що дає змогу виключити невдалі питання.

Реалізація модуля контролю та обмеження прав доступу.

Модуль контролю та обмеження прав доступу призначений для розподілення рівнів використання та адміністрування між користувачами веб-ресурсу. Це дозволяє серед усіх користувачів обрати (визначити)

тих, які будуть мати право керувати веб-системою тестування, а саме, створювати облікові записи користувачів, створювати та редагувати тести, переглядати результати тестування. Для користувачів системи необхідно надати доступ лише до самих тестів, а саме до системи тестування, де користувач може обрати тему тестування, та пройти відповідні тести.

Реалізація модуля створення та редагування тестів.

Найголовнішим модулем у веб-системі тестування є модуль створення та редагування тестів. Цей модуль повинен містити різноманітні компоненти, необхідні для створення тестів, оскільки існує велика кількість різних варіантів тестів. Тести можуть бути з одним чи декількома правильними відповідями, з вмістом картинок або системою якихось елементарних розрахунків. Наявність засобів, які дають можливість створювати такі комплексні тести, дозволяє створити якісні та сучасні набори тестів для оцінювання знань у різних сферах.

Реалізація системи зберігання даних.

Майже всі дані, які відображаються у системі веб-тестування зберігаються у базі даних. Це обумовлено тим, що зберігати дані безпосередньо на веб-сторінках є не практичним, оскільки у випадках застосування багатомовності чи використання часто використовуваних термінів або назв (назва організації, ім'я студента) необхідно буде створювати велику кількість подібних веб-сторінок, що, в свою чергу, значно ускладнить процес адміністрування веб-ресурсом. Отже спосіб зберігання контентних даних (дані, що відповідають за наповнення веб-ресурсу) у базі даних є досить зручним і гнучким з точки зору розробки та адміністрування.

Слід зазначити, що наразі існує досить велика кількість різноманітних атак на веб-ресурси, метою яких є неконтрольований доступ до бази даних, що, в свою чергу, дозволяє зчитувати, видаляти, змінювати дані або блокувати доступ до них. Існує багато механізмів протидії таким атакам, але незалежно від їх реалізації можна створити ще один механізм захисту, що побудований на шифруванні даних, які зберігаються в базі даних.

Розробка механізмів захисту бази даних та даних, що передаються в системі тестування.

Розглянемо більш детально можливість розробки механізмів захисту бази даних, що використовується у веб-системі тестування та даних, що передаються в цій системі.

Шифрування є одним з найбільш поширених методів захисту даних. Існує досить багато різних методів шифрування з різними підходами та реалізованих в досить великій кількості різноманітних алгоритмів. При виборі методів та алгоритмів шифрування для розроблювальної веб-системи тестування слід враховувати те, що процес передавання даних – це відносно короткотривалий процес, тому стійкість алгоритмів і ключів, які при цьому використовуються, повинні відповідати тим часовим проміжкам, під час яких зловмисник може використати перехоплені дані. В той же час самі дані, що зберігаються, зберігаються довгостроково в середині системи, тобто на жорсткому диску у вигляді файлів або упорядковано в каталозі бази даних.

Захищати дані на рівні СУБД можна декількома способами. Перш за все можна зашифрувати саму базу даних, яка підлягає захисту, але такий спосіб не захищає від проникнення зловмисника через існуючі підключення, тобто захищає лише на рівні файлової системи. Для більш ефективного захисту слід використовувати шифрування на рівні таблиць, колонок або навіть комірок. Чим менша у базі даних структурна одиниця, яку ми захищаємо, тим складніше реалізувати атаку, але і, відповідно, більша складність механізму захисту та необхідність використання більших процесорних потужностей.

В розроблюваній системі тестування необхідно використовувати додатковий захист для таких даних як набір питань та відповідей, що використовуються при тестуванні. Для підвищення стійкості пропонується здійснювати шифрування з використанням різних ключів для тематичних питань і відповідей на них. Це обмежить зловмисника у доступі до даних, оскільки при реалізації атаки на вміст бази даних з використанням секретного ключа доступ можна буде отримати лише до певних тестів, що значно обмежить об'єм інформації, до якої зловмисник зможе отримати доступ.

Як механізм шифрування для системи тестування можна використовувати як готові, так і власноруч створені механізми. Зокрема, можна використати такі готові механізми шифрування як вбудований механізм EFS [12] шифрування файлової системи Windows, або механізм шифрування даних TrueCrypt [9] в системі Linux, або вбудовані механізми СУБД MS SQL, MySQL, Oracle та ін. Однак недоліком готових вбудованих механізмів шифрування є те, що досить велика кількість зловмисників постійно намагаються знайти в цих системах захисту слабкі місця та можливість використати їх. Найчастіше це відбувається за рахунок виявлення слабких місць в алгоритмах роботи програм, відомих як bug, а також у використанні залишених розробниками BackDoor при тестуванні алгоритмів під час розробки, які випадково їх «залишають» в коді програми після випуску їх фінальних версій. Це є досить суттєвим недоліком, оскільки якщо зловмисники знаходять такі слабкі місця, вони миттєво розповсюджують про ці відомості мережею Інтернет та створюють програмні додатки, які може використовувати будь-який користувач, тим самим значно збільшуючи вірогідність атаки.

Враховуючи сказане, в розроблюваній системі тестування доцільно створювати власний механізм шифрування. Окрім механізму шифрування необхідно розробити власні механізми розподілу секретних ключів, автентифікації сторін взаємодії а також цифрового підписування.

Як вже зазначалось вище, необхідність шифрування даних у системі тестування виникає як на рівні зберігання даних, так і на рівні передавання даних у вигляді питань тестів, що надходять до кандидатів тестування, а також результатів їх відповідей на питання, що передаються від клієнта до сервера. У першому випадку для зберігання даних доцільно використовувати симетричні алгоритми шифрування, у другому випадку, для шифрування даних, що передаються, доцільно використовувати асиметричні алгоритми. Для асиметричного шифрування в системі тестування, що розробляється, пропонується використовувати власний метод шифрування на основі математичного апарату рекурентних V_k –послідовностей [13], який дозволяє підвищувати криптографічну стійкість шифрування.

В разі необхідності розподілу секретних ключів у системі тестування можна використати власний алгоритм розподілу секретних ключів відкритим каналом на основі рекурентних V_k –послідовностей [14], який за певних умов дозволяє підвищувати стійкість криптографічних перетворень під час розподілу ключів.

Необхідність автентифікації у системі тестування виникає у першу чергу на рівні авторизації користувачів та ідентифікації робочих станцій, а також можлива і під час безпосереднього проходження тестування. Для реалізації автентифікації у системі тестування пропонується використовувати власний метод автентифікації сторін взаємодії на основі математичного апарату рекурентних V_k –послідовностей [15], який, окрім забезпечення достатнього рівня криптографічної стійкості, дозволяє значно спрощувати обчислення процедури перевірки автентичності, що може бути корисним під час авторизації або ідентифікації претендентів, особливо на початку тестування навчальних груп користувачів (студентів), коли водночас на сервер може йти одразу багато запитів на проходження тестування.

Необхідність цифрового підписування у системі тестування виникає під час безпосереднього тестування, коли всі відповіді кандидата, що тестується, мають бути підписані цифровим підписом і відправлені на сервер для перевірки. Тут слід зазначити, що при виборі алгоритму цифрового підписування бажано використовувати такий, який вимагав би якомога меншої обчислювальної складності виконання процедури формування цифрового підпису, оскільки кандидати, що проходять тест в системі тестування, можуть використовувати невисокопродуктивні робочі станції (мобільні телефони, планшети і т. і.). Враховуючи це, для цифрового підписування у системі тестування пропонується використовувати один з власних методів цифрового підписування на основі математичного апарату рекурентних V_k –послідовностей, що представлені в роботах [16] та [17], які забезпечують спрощення обчислень під час цифрового підписування.

В результаті розроблено захищену веб-систему тестування, в якій усі технології криптографічного захисту, як то шифрування, ключовий обмін, автентифікація та цифрове підписування є власними і реалізовано на єдиному математичному апараті рекурентних V_k –послідовностей. Це надає додаткові переваги реалізації, оскільки використання єдиного апарату значно спрощує процедури вибору параметрів, а також надає можливість використання проміжних результатів обчислення елементів рекурентних послідовностей для криптографічних застосувань різного призначення, що, в свою чергу, дозволяє підвищити швидкість виконання криптографічних перетворень в цих застосуваннях.

III Висновки

Таким чином запропоновано реалізацію системи тестування з використанням крос-платформних рішень, які надають широкі можливості щодо побудови захищених систем з високим рівнем безпеки, а також веб-технологій з гнучкою системою доступу кандидатів до системи тестування, що дозволяє використовувати систему тестування без прив’язування до конкретної платформи чи операційної системи на будь-якому пристрої, що підключений до глобальної мережі Інтернет або локальної мережі з використанням власного веб-серверу.

Представлено структурну схему захищеної веб-системи тестування а також зв’язки та взаємодію її структурних модулів. Розглянуто особливості розробки кожного програмного модуля веб-системи тестування.

Проведено аналіз проблем забезпечення безпеки системи тестування, а також можливі атаки зловмисника. Розроблено механізми захисту бази даних та даних, що передаються в системі тестування між клієнтами-користувачами та сервером. Усі необхідні технології криптографічного захисту як то шифрування, ключового обміну, автентифікації та цифрового підписування розроблено на основі власних методів на

єдиному математичному апараті рекурентних V_k –послідовностей. Це дозволило не лише підвищити рівень безпеки системи тестування, але й підвищити швидкість виконання в ній криптографічних перетворень.

Список використаної літератури: ***1.*** *Кабанова Т. А., Новиков В. А. Тестирование в современном образовании. Учебное пособие. – М.: Высшая школа, 2010. – 384 с.* ***2.*** *Винник В. К. Обзор дистанционных электронных платформ обучения // Научный поиск. – № 2.5, 2013. – С. 5–7.* ***3.*** *Алексеев А. Н. Дистанционное обучение инженерным специальностям: Монография. – Суммы: ИТД «Универсальная книга», 2005. – 333 с.* ***4.*** *Агапонов С. В., Джалиашвили З. О., Кречман Д. Л. и др. Средства дистанционного обучения. Методика, технология, инструментарий: ред. З. О. Джалиашвили. – СПб.: БХВ-Петербург, 2003. – 336 с.* ***5.*** *Ибрагимов, И. М. Информационные технологии и средства дистанционного обучения: Учебное пособие для студ. вузов. – М. : Академия, 2005. – 336 с.* ***6.*** *Воронов М. В. Дистанционные образовательные технологии и перераспределение функций профессорско-преподавательского состава // Социология образования. – 2009. – № 5. – С. 40-51.* ***7.*** *Батешов Е. А. Основы технологизации компьютерного тестирования. Учебное пособие. – Астана: ТОО «Полиграф-мир», 2011. – 241 с.* ***8.*** *Калюжный, А. С. Компьютерное тестирование как способ контроля знаний студентов // Высшее образование сегодня. – 2009. – № 7. – С. 67–68.* ***9.*** *Брэдфорд Эд.- Може Лу. Кроссплатформенные приложения для Linux и Windows. – СПб: Питер, 2003. – 672 с.* ***10.*** *Уланов А. В. Многоагентное моделирование механизмов защиты от атак "распределенный отказ в обслуживании": дис. ... канд. техн. наук : 05.13.18, 05.13.19 / Уланов А. В. – Санкт-Петербург, 2007. – 165 с.* ***11.*** *Ястребов И. С. Математические модели и реализация контроля доступа на основе ролей и контекста для распределенной системы управления физическим экспериментом: дис. ... канд. техн. наук : 05.13.18 / Ястребов И. С. – Ульяновск. 2010. – 139 с.* ***12.*** *Нортроп Тони, Макин Дж. Microsoft. Учебный курс 70-642. – М: Русская Редакция (Microsoft Press), 2008. – 570 с.* ***13.*** *Яремчук Ю.Є. Метод шифрування інформації з відкритим ключем на основі рекурентних послідовностей // Інформаційна безпека. – №3, 2013. – С. 123–129.* ***14.*** *Яремчук Ю. Є. Метод відкритого розподілу секретних ключів на основі рекурентних послідовностей // Інформаційна безпека. – №2, 2013. – С. 177–183.* ***15.*** *Яремчук Ю. Є. Метод автентифікації суб'єктів (об'єктів) взаємодії на основі рекурентних послідовностей // Вісник Вінницького політехнічного інституту. – №3, 2013. – С. 99–104.* ***16.*** *Яремчук Ю. Є. Можливість цифрового підписування на основі рекурентних послідовностей // Інформатика та математичні методи в моделюванні. – Том 3, №1, 2013. – С. 13–21.* ***17.*** *Яремчук Ю. Є. Можливість формування та перевірки цифрового підпису на основі рекурентних послідовностей // Вісник Вінницького політехнічного інституту. – №5, 2013. – С. 91–95.*

Сергей Егоров
Национальный авиационный университет

УДК 004.056:004.451.1(045)

ЗАЩИТА И МОНИТОРИНГ ПОРТОВ ВВОДА ВЫВОДА В ОПЕРАЦИОННЫХ СИСТЕМАХ LINUX

Аннотация: **Разработаны рекомендации относительно анализа состояния портов операционной системы Linux и обеспечения их защиты.**
Summary: **Recommendations regarding the analysis of the status of ports of the Linux operating system and ensure their protection.**
Ключевые слова: **Операционные системы, порты ввода-вывода, безопасность, информационная безопасность, Linux, Unix администрирование Linux.**

I Постановка задачи

Службы – это основная составляющая сетевой рабочей среды компьютеров. Если бы не было служб, то в компьютере можно было бы запускать не более одной программы, к которой можно было бы подключаться клиентам по одному за раз. В основе работы служб лежит концепция портов. Порт – это специальная добавка к IP-адресу, которая позволяет определить серверу, как именно работать с клиентской частью. Большинство пользователей очень плохо знакомы с портами, не говоря уже о механизме их функционирования. Это связано с тем, что для работы с сетью необходимо знать только IP-адрес, а номер порта знать не обязательно, потому что этот номер запрограммирован внутри программы.