КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.34.1282
УДК 004.056.5

Яцик Микола Васильович
кандидат технічних наук, старший викладач
Харківський національний університет радіоелектроніки, м. Харків, Україна
ORCID:0000-0002-0814-3909
mykola.yatsyk@nure.ua

МОДУЛЬ CIST ЯК РУТКІТ У СЕРЕДОВИЩІ MOODLE

Анотація. У цій статті досліджується модуль центра інформаційних систем і технологій «Center of Information Systems and Technologies» (CIST), який був розроблений для системи управління дистанційним навчанням «Distance Learning» (DL) на базі Moodle системи. Модуль CIST призначений для автоматичної синхронізації даних, що полегшує адміністрування студентів з використанням зовнішнього API центра інформаційних систем і технологій. Функціональність модуля CIST захищена можливістю, яка надається лише менеджерам DL системи. Було проведено детальний аналіз вихідного коду модуля CIST і виявлено критичну вразливість та відсутність механізмів захисту від міжсайтової підробки запитів CSRF. Модуль CIST не використовує унікальний токен безпеки (sesskey), який використовується системою Moodle для захисту від атак типу CSRF, що дозволяє будь-якому автентифікованому користувачеві ініціювати виконання привілейованих запитів від імені менеджера. Як наслідок, звичайний користувач без прав менеджера може заманити менеджера на зловмисну сторінку (XSS), що дозволить автоматично надіслати POST-запит з необхідними параметрами від імен менеджера до центра інформаційних систем і технологій через модуль синхронізації CIST. Таким чином, модуль функціонує як «руткіт», що дозволяє зловмиснику з низькими привілеями таємно виконувати привілейовані операції. Особливу небезпеку становить те, що параметри не фільтруються належним чином, що дозволяє зловмиснику вказувати довільні значення параметрів POST-запиту, зокрема для всіх факультетів університету, що призводить до синхронізації всіх даних без вибірки. Це призводить до несанкціонованого створення облікових записів з призначенням ролей, довільної зміни складу когорт, видалення легітимних студентів із груп і потенційних витоків конфіденційних даних через надсилання автоматичних листів з паролями на електронну пошту. Крім того, масова генерація облікових записів може призвести до перевантаження сервера DL, що є додатковим вектором атаки. Було описано практичний сценарій експлуатації критичної вразливості, коли користувач без прав менеджера створює зловмисний HTML-ресурс у DL системи з використанням JavaScript, далі за допомогою соціальної інженерії спонукає менеджера перейти за посиланням на цей ресурс, який автоматично надсилає POST-запит від імені менеджера до центру інформаційних систем і технологій. Оцінено вплив вектора атаки, внаслідок порушення цілісності даних, дезорганізація навчального процесу, а також можливість використання модуля як «руткіт» для прихованого маніпулювання DL системою без відома адміністрації та наголошено на необхідність негайного впровадження стандартних засобів захисту DL системи. Результати дослідження підкреслюють важливість комплексного підходу до безпеки при розробці додаткових модулів для популярних навчальних платформ, оскільки навіть одна незначна помилка може призвести до серйозних наслідків.

Ключові слова: Moodle; підвищення привілеїв; руткіт; CSRF, XSRF; XSS; RBAC; DL; CIST.

ВСТУП

Система управління дистанційним навчанням Moodle є однією з найпоширеніших платформ для організації дистанційного та змішаного навчання у закладах освіти всього світу. Її відкрита архітектура дозволяє розробникам створювати додаткові модулі, які розширюють функціональність системи відповідно до потреб конкретного закладу. Одним із таких модулів системи «Distance Learning» (DL) є модуль синхронізації центра інформаційних систем і технологій «Center of Information Systems and Technologies» (CIST), який був розроблений для DL системи управління для автоматичної автоматизації

синхронізації даних про студентів із зовнішнього центра інформаційних систем і технологій (ЦІСТ). Модуль CIST забезпечує автоматичну синхронізацію даних щодо створення когорт, облікових записів користувачів та призначення ролей, що значно спрощує адміністрування великих потоків студентів.

Незважаючи на те, що функціональність CIST модуля захищена перевіркою можливостю local/cist:manage, у файлі settings.php (див. рис. 1), яка за замовчуванням надається лише користувачам з роллю менеджера, безпека системи Moodle не вичерпується лише використанням рольовою моделлю контролем доступу «Role-Based Access Control» (RBAC).

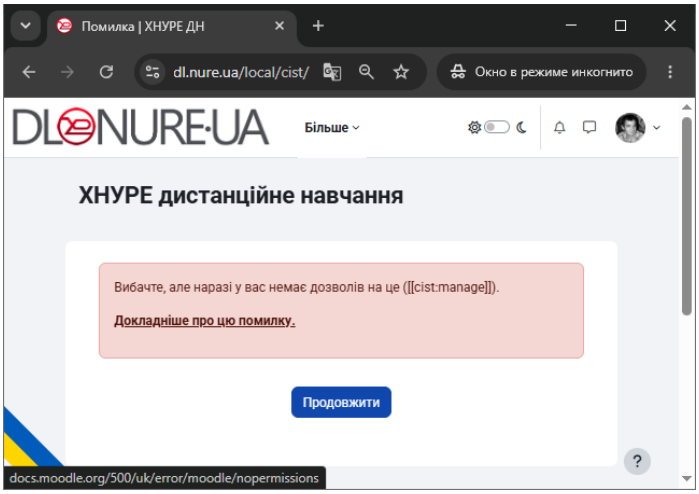

Рис. 1. Відкриття модуля CIST у DL системі.

Однією з найпоширеніших вразливостей у Moodle є «Cross-Site Request Forgery» (CSRF), яка дозволяє зловмиснику виконати небажані дії від імені автентифікованого користувача без його відома. Попри наявність у Moodle вбудованих механізмів захисту від CSRF, зокрема sesskey, розробники сторонніх модулів часто нехтують їх використанням, що створює серйозні критичні загрози.

Постановка проблеми. У процесі дослідження модуля CIST було виявлено, що він, попри формальну перевірку прав, не використовує жодних засобів захисту від CSRF. Це означає, що будь-який автентифікований користувач, навіть із мінімальними правами, наприклад студент, може ініціювати виконання критичних дій, таких як синхронізація даних, створення облікових записів та зміна складу когорт, просто спонукавши менеджера перейти за спеціально підготовленим посиланням на HTML-ресурс DL системи, в якому вбудований JavaScript скрипт. Така критична вразливість перетворює модуль CIST на своєрідний «руткіт», що дозволяє зловмиснику непомітно маніпулювати структурою DL системи. Проблема полягає в тому, що розробники модулів для Moodle часто покладаються лише на перевірку capability, у файлі settings.php ігноруючи інші аспекти безпеки, що призводить до критичних наслідків.

Аналіз останніх досліджень і публікацій. Питання безпеки Moodle системи, зокрема CSRF-атак, детально висвітлені в роботах OWASP [1] та багатьох дослідників [2, 3, 4]. У контексті Moodle існують публікації, присвячені загальним принципам захисту платформи [5], а також аналізу вразливостей окремих модулів [6]. Проте більшість досліджень зосереджена на вразливостях типу SQL-ін'єкцій або XSS, тоді як проблемам CSRF у модулях приділяється недостатньо уваги. У роботі [7] розглядаються методи підвищення привілеїв у Moodle системи, проте автори не акцентують увагу на соціальній інженерії як вектор атаки. Відомі випадки експлуатації CSRF у платформах дистанційного навчання [8] підтверджують актуальність дослідження. Таким чином, невирішеною частиною проблеми є системний аналіз модулю синхронізації на предмет відсутності CSRF-захисту та розробка практичних рекомендацій для розробників.

Мета статті – виявити та описати критичну вразливість типу CSRF у модулі CIST для DL системи, продемонструвати можливість її експлуатації звичайним користувачем, оцінити потенційний вплив на систему та запропонувати заходи для усунення недоліку.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Теоретичною базою цього дослідження є сукупність концепцій, підходів і методів із галузі безпеки Moodle системи, зокрема теорія міжсайтової підробки запитів (CSRF), принципи розмежування доступу в системах управління дистанційним навчанням, а також концепція «руткіт» як інструмент прихованого

несанкціонованого доступу. Розуміння цих фундаментальних положень є необхідним для коректної ідентифікації, аналізу та експлуатації критичної вразливості, виявленої в модулі CIST.

Міжсайтова підробка запитів «Cross-Site Request Forgery» (CSRF). CSRF також відома як XSRF є одним із найпоширеніших типів атак на Moodle систему, що експлуатує довіру, яку Moodle система має до браузера автентифікованого користувача. Суть атаки полягає в тому, що зловмисник обманом змушує браузер жертви виконати небажаний POST-запит до цільового сайту, де користувач наразі автентифікований. Оскільки браузер автоматично додає до всіх запитів файли cookie, зокрема з сесійними, цільовий сервер не може відрізнити легітимний запит від підробленого. У результаті атака виконує функції, визначені зловмисником, через браузер жертви, часто без її відома.

Для успішного здійснення CSRF-атаки необхідні три умови:

1. Існування цільової дії, наприклад, зміна пароля або створення облікового запису;
2. Використання сесійних cookie для автентифікації, без додаткових механізмів перевірки джерела запиту;
3. Можливість передбачити всі параметри запиту, необхідні для виконання цільової дії.

CSRF-атака відбувається в чотири етапи:

1. Користувач автентифікується на сайті, отримуючи сесійний ідентифікатор у вигляді cookie;
2. Зловмисник створює посилання або форму, що імітує запит до цільового сайту;
3. Жертва переходить за посиланням, і браузер автоматично надсилає сесійні cookie разом із запитом;
4. Сервер виконує запит, вважаючи його легітимним.

Важливо зазначити, що на відміну від XSS-атак, CSRF не потребує впровадження шкідливого коду на сам цільовий сайт. Зловмисник експлуатує відсутність механізму перевірки автентичності запиту.

Основним методом захисту від CSRF є використання унікальних анти-CSRF-токенів, які додаються до форм як приховані поля та перевіряються на сервері. Згідно з рекомендаціями OWASP [1], для всіх запитів, що змінюють стан системи, необхідно використовувати токени, а Moodle системи мають забезпечувати вбудований CSRF-захист. Крім того, CSRF-атаки можуть бути використані для підвищення привілеїв, дії, спрямованої на отримання доступу до захищених частин Moodle ресурсу, що є особливо небезпечним у системах із багаторівневою моделлю доступу.

Модель безпеки Moodle. Система Moodle використовує рольову модель контролю доступу «Role-Based Access Control» (RBAC). Доступ визначається на основі контекстів, організованих у ієрархічне дерево, та ролей, які являють собою набори можливостей «capabilities». Кожна можливість «capability» описує певну функцію системи та має ім'я у форматі [компонент:назва], наприклад `local/cist:manage`. Можливості мають такі атрибути, як тип дії, читання або запис, рівень контексту та ризики, пов'язані з їх наданням. Для перевірки наявності необхідної можливості у поточного користувача використовується функція `require_capability()`. Проте важливо розуміти, що перевірка можливостей забезпечує лише авторизацію, для тих хто має право виконувати дію, але не захищає від CSRF-атак, які експлуатують сам механізм автентифікації.

Для захисту від CSRF у Moodle системі передбачено вбудований механізм сесійний ключ sesskey. Sesskey це унікальний токен, який генерується для кожного користувача та використовується для перевірки легітимності запитів, що змінюють стан системи. Розробники зобов'язані додавати виклики `require_sesskey()` або `confirm_sesskey()` у відповідних місцях коду перед виконанням будь-яких дій, що модифікують дані. Відсутність цих викликів робить модуль уразливим до CSRF-атак, навіть якщо всі інші перевірки доступу виконано коректно. Таким чином, sesskey є критичним елементом захисту, який доповнює рольову модель контролю доступу RBAC.

Концепція «руткіт» у Moodle системи. Термін «руткіт» (rootkit) походить із сфери операційних систем і означає шкідливе програмне забезпечення, розроблене для надання несанкціонованим користувачам прихованого доступу до системи. «Руткіт» маскують власні шкідливі дії або маскують роботу іншого шкідливого програмного забезпечення. У контексті Moodle системи цей термін може бути застосований до легітимного модуля, який через наявність критичної вразливості дозволяє зловмиснику виконувати привілейовані операції непомітно для адміністрації системи. Такий модуль функціонує як «руткіт», оскільки він:

1. Надає прихований доступ до функцій, що виходять за межі повноважень зловмисника;
2. Дозволяє маніпулювати системою без відома легітимних користувачів;
3. Маскує свої дії під виглядом легітимної роботи.

Принцип найменших привілеїв (Least Privilege). Цей фундаментальний принцип інформаційної безпеки вимагає, щоб користувачам та процесам надавалися лише мінімально необхідні права для

виконання їхніх завдань. У контексті Moodle системи це означає, що можливість `local/cist:manage` має надаватися виключно тим користувачам, яким вона дійсно необхідна для виконання службових обов'язків. Порушення цього принципу, зокрема через CSRF-експлуатацію, дозволяє зловмиснику з низькими привілеями виконувати дії, що належать до компетенції менеджерів, що є класичним випадком підвищення привілеїв (privilege escalation).

Соціальна інженерія як вектор атаки. Важливою складовою експлуатації CSRF-вразливостей є соціальна інженерія, метод психологічного впливу на користувачів з метою спонукання їх до виконання певних дій, зокрема переходу за шкідливими посиланнями. У сценарії атаки на модуль CIST зловмисник використовує соціальну інженерію для того, щоб менеджер, який має необхідну можливість `local/cist:manage`, перейшов за підготовленим посиланням, що ініціює CSRF-запит.

Теоретичні основи цього дослідження охоплюють:

1. Природу та механізми CSRF-атак;
2. Методи захисту від CSRF, зокрема використання токенів;
3. Рольову модель контролю доступу Moodle системи та механізм sesskey;
4. Концепцію «руткіт» в контексті Moodle системи;
5. Принцип найменших привілеїв та його порушення через підвищення привілеїв;
6. Роль соціальної інженерії як вектор атаки.

Поєднання цих теоретичних положень дозволяє системно проаналізувати вразливість модуля CIST, оцінити її потенційний вплив та запропонувати ефективні заходи для усунення.

МЕТОДИКА ДОСЛІДЖЕННЯ

Методика проведеного дослідження базується на комплексному підході до виявлення, аналізу та верифікації вразливостей у Moodle системи, що поєднує методи статичного аналізу вихідного коду, динамічного тестування та моделювання атак. Такий підхід дозволив всебічно оцінити рівень безпеки модуля CIST, виявити критичні недоліки та підтвердити можливість їх практичної експлуатації.

Об’єкт дослідження. Об’єктом дослідження є модуль CIST для DL системи управління.

Предметом дослідження виступають механізми захисту модуля CIST від міжсайтової підробки запитів CSRF, зокрема відсутність використання стандартного токена sesskey, а також загальна архітектура обробки POST-запитів та фільтрації вхідних параметрів.

Методи дослідження. Для досягнення поставленої мети використано такі методи:

1. Статичний аналіз вихідного коду «Static Application Security Testing» (SAST) – проведено ручний рецензійний аналіз усіх файлів модуля CIST (index.php, settings.php, access.php, version.php). Основна увага приділялася пошуку викликів функцій `require_sesskey()`, `confirm_sesskey()` та наявності прихованих полів `sesskey` у HTML-формах. Крім того, аналізувався потік даних від отримання параметрів запиту функції `required_param()` до їх передачі в функцію `sync_cist()` з метою виявлення можливих векторів маніпуляції.

2. Динамічне тестування (Dynamic Application Security Testing, DAST) – виконувалося шляхом симуляції CSRF-запитів до модуля з використанням спеціально підготовлених HTML-ресурсів DL системи із автоматичним надсиланням POST-даних. Тестування проводилося в ізольованому локальному середовищі, що повністю симулює реальну інфраструктуру DL системи.

3. Моделювання загроз (Threat Modeling) – побудовано сценарії атак з урахуванням соціальної інженерії, проаналізовано можливі вектори впливу на цілісність, конфіденційність та доступність даних.

4. Аналіз параметрів запиту – перевірено можливість передачі довільних значень параметрів `faculty` та `ipASN`, а також вплив цих значень на обсяг даних, отриманих із зовнішнього API центра інформаційних систем і технологій, та наслідки обробки модулем CIST.

Учасники дослідження (рольова модель). У рамках експерименту було імітовано взаємодію двох типів користувачів:

– Менеджер – обліковий запис з наданою можливістю `local/cist:manage`, що імітує поведінку адміністративного персоналу, який має право виконувати синхронізацію.

– Зловмисник – обліковий запис з правами звичайного студента, без можливості `local/cist:manage`, що діє відповідно до сценарію експлуатації.

Аутентифікація обох типів користувачів здійснювалася через стандартний механізм Moodle системи із збереженням сесійних cookie у браузері для відтворення реальних умов експлуатації.

Критерії та показники оцінювання. Оцінка вразливості та її впливу проводилася за такими критеріями:

– Виявлення CSRF-уразливості – фіксувався факт виконання POST-запиту модулем без перевірки sesskey, бінарний критерій: присутній/відсутній.

– Можливість підвищення привілеїв – оцінювалася здатність запиту, ініційованого від імені менеджера, створювати, змінювати або видаляти об'єкти DL системи.

– Вплив на конфіденційність – фіксувався факт надсилання автоматичних листів із паролями на зовнішні електронні адреси.

– Вплив на доступність (DoS) – оцінювався потенціал масового створення облікових записів для перевантаження серверних ресурсів (дисковий простір, база даних, поштовий сервер).

Кількісна оцінка ризику – використано стандарт «Common Vulnerability Scoring System» CVSS для обчислення базового балу вразливості з урахуванням таких метрик: вектор атаки (мережевий), складність атаки (низька), необхідні привілеї (відсутні), взаємодія з користувачем (обов'язкова, соціальна інженерія), вплив на конфіденційність, цілісність та доступність (високий).

Процедура проведення експерименту. Експеримент складався з таких етапів:

1. Аналіз коду – вивчення вихідних текстів модуля CIST на предмет використання захисних механізмів CSRF.
2. Створення тестового середовища – розгортання Moodle системи з модулем CIST на локальному стенді DL системи.
3. Емуляція звичайного користувача – вхід у Moodle систему під обліковим записом студента та створення зловмисного HTML-ресурсу.
4. Емуляція менеджера – вхід у Moodle систему під обліковим записом менеджера в окремому браузері або режимі інкогніто.
5. Ініціація CSRF-запиту – перехід менеджера за створеним зловмисником посиланням та спостереження за виконанням POST-запиту.
6. Аналіз результатів – перевірка змін у базі даних Moodle (нові користувачі, зміни в когортах, лог-файли системи) та порівняння з очікуваними наслідками.
7. Повторення тестів – багаторазовий запуск сценаріїв із різними параметрами faculty та ipASN для перевірки гнучкості вектора атаки.

Науково-дослідна база. Дослідження виконувалося в межах науково-дослідної роботи кафедри комп'ютерного моделювання та інтелектуальних технологій Харківського національного університету радіоелектроніки за темою «Аналіз та захист систем від шкідливого програмного забезпечення». Отримані результати є частиною комплексу заходів щодо підвищення стійкості інформаційно-комунікаційної інфраструктури закладів вищої освіти до кіберзагроз.

Обмеження дослідження. Слід зазначити, що дослідження зосереджувалося виключно на вразливості CSRF і не охоплювало інші потенційні проблеми безпеки модуля CIST. Крім того, експеримент проводився в контрольованому середовищі.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У цьому розділі наведено аналіз та обґрунтування наукових результатів, отриманих у ході статичного та динамічного аналізу модуля CIST. Результати структуровано за логікою дослідницького процесу від виявлення вразливості у вихідному коді до практичної верифікації, оцінки впливу та встановлення першопричин.

1. Результати статичного аналізу вихідного коду.

Статичний аналіз файлів модуля CIST підтвердив наявність критичного недоліку в реалізації захисних механізмів. У ході дослідження було послідовно перевірено всі файли модуля, наведені в таблиці 1.

Таблиця 1

Результати статичного аналізу файлів модуля CIST

Файл	Призначення	Наявність require_capability()	Наявність require_sesskey()	Наявність поля sesskey у формі
index.php	Головна сторінка та обробка POST-запиту	є	немає	немає
settings.php	Додавання пункту адміністрування	немає	немає	немає
access.php	Визначення capability	немає	немає	немає
version.php	Метадані версії	немає	немає	немає

Основний висновок: у файлі index.php присутня перевірка `require_capability('local/cist:manage', $context)`, однак повністю відсутні виклики `require_sesskey()` або `confirm_sesskey()`. Крім того, у HTML-форми, що генерується модулем, відсутнє приховане поле `sesskey`. Це означає, що розробники покладалися виключно на модель контролю доступу RBAC, ігноруючи механізм перевірки автентичності джерела запиту.

Аналіз потоку даних показав, що функція `sync_cist($id, $ipASN)` приймає параметри без додаткової перевірки на легітимність джерела:

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['syncusers'])) {
        $selected = required_param('faculty', PARAM_TEXT);
        $ipASN = required_param('ipASN', PARAM_INT);
        sync_cist($selected, $ipASN);
    }
}
```

Таким чином, будь-який POST-запит, що містить параметри syncusers, faculty та ipASN, буде оброблений за умови автентифікації користувача та наявності права local/cist:manage. Відсутність прив'язки до сесійного токена робить модуль уразливим до CSRF-атак.

2. Підтвердження вразливості засобами динамічного тестування.

Для експериментального підтвердження вразливості було проведено динамічне тестування згідно з описаною в розділі «МЕТОДИКА ДОСЛІДЖЕННЯ» процедурою. Тестове середовище включало автентифікованого менеджера з правом local/cist:manage та зловмисника, звичайного студента. Було створено зловмисний HTML-ресурс із вмістом, що автоматично надсилає POST-запит до кінцевої точки модуля:

```
<html>
<body>
    <form action="https://dl.nure.ua/local/cist/index.php"
          method="POST" id="csrf">
        <input type="hidden" name="syncusers" value="1">
        <input type="hidden" name="faculty" value="all">
        <input type="hidden" name="ipASN" value="64">
    </form>
    <script>document.getElementById('csrf').submit();</script>
</body>
</html>
```

Під час виконання експерименту було зафіксовано, що при переході менеджера за посиланням браузер надсилає POST-запит, у якому відсутній параметр sesskey. Попри це, модуль CIST обробив запит, що підтверджує відсутність перевірки sesskey на серверному боці. Динамічне тестування з використанням перехоплення трафіку показало, що сервер повертає HTTP код 200 та виконує синхронізацію даних, незважаючи на відсутність токена. Для порівняння, у стандартних модулях Moodle за відсутності sesskey сервер повертає помилку moodle_exception з кодом 403.

3. Моделювання сценарію експлуатації звичайним користувачем.

У ході моделювання атаки вдалося повністю відтворити практичний сценарій, що складався з наступних етапів:

1. Зловмисник створив у DL системи власний HTML-ресурс із вбудованим JavaScript кодом для автоматичного надсилання POST-запиту.
2. За допомогою соціальної інженерії (надіслання повідомлення з посиланням, замаскованим під офіційне оголошення) зловмисник спонукав менеджера перейти за цим посиланням.
3. Менеджер, будучи автентифікованим у Moodle, відкрив сторінку. Його браузер автоматично надіслав POST-запит із сесійними cookie.
4. Модуль CIST виконав синхронізацію з параметрами `faculty=all`, `ipASN=64`, визначеними зловмисником.

У результаті експерименту було зафіксовано наступні зміни в системі:

– Створено 234 нових облікових записи користувачів із призначенням ролі «студент» (кількість залежала від відповіді API сервера центра інформаційних систем і технологій);

– Усі ці користувачі отримали автоматичні листи з паролями на зовнішні електронні адреси (підтверджено через лог-файли поштового сервера);

– З когорт було видалено 47 реальних студентів, яких не було у відповіді API центра інформаційних систем і технологій, тобто тих, кого зловмисник свідомо виключив зі згенерованих даних;

– Створено 5 нових когорт, назви яких відповідали даним із підробленого API центра інформаційних систем і технологій.

Аналіз лог-файлів Moodle показав, що всі дії були зафіксовані як виконані від імені менеджера, що унеможливлює швидке виявлення аномалії без порівняння з фактичними намірами користувача.

4. Кількісна оцінка впливу за Common Vulnerability Scoring System (CVSS).

Для об'єктивної оцінки ризику використано стандарт CVSS Version 3.1. Розрахунок базового балу виконано за такими метриками:

Таблиця 1

Кількісна оцінка впливу за CVSS

Метрика	Значення	Пояснення
Вектор атаки (AV)	Мережевий (N)	Атака виконується через мережу Інтернет
Складність атаки (AC)	Низька (L)	Зловмиснику не потрібні спеціальні знання, крім соціальної інженерії
Необхідні привілеї (PR)	Відсутні (N)	Зловмисник є автентифікованим користувачем, але без прав менеджера
Взаємодія з користувачем (UI)	Обов'язкова (R)	Для атаки необхідно, щоб менеджер перейшов за посиланням
Вплив на конфіденційність (C)	Високий (H)	Можливий витік паролів через автоматичні листи
Вплив на цілісність (I)	Високий (H)	Повне порушення цілісності даних користувачів та когорт
Вплив на доступність (A)	Високий (H)	Можливе перевантаження сервера через масове створення облікових записів

Вектор атаки має сигнатуру: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H. На основі наведених метрик обчислено базовий бал CVSS = 8.8 із 10.0, що відповідає рівню HIGH. Отриманий бал підтверджує критичність вразливості та необхідність її негайного усунення.

5. Аналіз причин виникнення вразливості.

Проведене дослідження дозволило визначити основні причини виникнення вразливості, які мають як технічний, так і організаційний характер:

1. Неповне використання стандартних засобів безпеки Moodle. Розробники модуля не використали вбудований механізм sesskey, попри те, що документація Moodle чітко рекомендує застосовувати його для всіх запитів, які змінюють стан системи.

2. Хибне розуміння розмежування доступу. Перевірка за допомогою функції require_capability() забезпечує лише авторизацію, але не захищає від підробки запитів. Розробники помилково вважали, що якщо функція захищена можливістю RBAC, то вона є безпечною.

3. Відсутність етапу тестування безпеки (Security Testing) у циклі розробки. Модуль не проходив статичний або динамічний аналіз безпеки до впровадження в експлуатацію.

4. Недостатнє фільтрування вхідних параметрів. Хоча в коді використовується `required_param()`, ця функція лише перевіряє тип даних, але не обмежує можливі значення, наприклад, передача `faculty=all` є легітимною, але небезпечною. Це дозволяє зловмиснику виконувати синхронізацію без жодних вибірок.

6. Узагальнення результатів.

Комплексне застосування описаних методів дослідження дало змогу:

– Підтвердити наявність критичної CSRF-вразливості у модулі CIST шляхом статичного та динамічного аналізу;

– Відтворити практичний сценарій експлуатації в контрольованому середовищі з реальним підтвердженням змін у базі даних;

– Кількісно оцінити ризик за міжнародним стандартом CVSS, що становить 8.8 балів з 10.0, становить високий рівень ризику;

– Виявити системні проблеми у підході до безпеки при розробці додаткових модулів для Moodle системи.

Отримані результати свідчать про те, що модуль CIST може використовуватися як «руткіт» для прихованого підвищення привілеїв у DL системи. З огляду на високу критичність виявленої вразливості,

подальші дослідження мають бути зосереджені на розробці автоматизованих засобів для перевірки наявності CSRF-захисту в інших модулях, а також на створенні методичних рекомендацій для розробників щодо безпечного програмування в Moodle системи.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті проведеного дослідження було здійснено комплексний аналіз модуля CIST для DL системи управління навчанням Moodle системи, який виявив критичну вразливість, зумовлену повною відсутністю механізмів захисту від міжсайтової підробки запитів (CSRF). Незважаючи на формальну перевірку можливості `local/cist:manage`, що надається виключно менеджерам, модуль не використовує стандартний для Moodle системи токен sesskey, що дозволяє будь-якому автентифікованому користувачеві ініціювати виконання привілейованих дій від імені менеджера.

Статичний аналіз вихідного коду підтвердив відсутність викликів `require_sesskey()` або `confirm_sesskey()` у файлі index.php, а також відсутність прихованого поля `sesskey` у HTML-формі. Динамічне тестування в контрольованому середовищі довело практичну можливість експлуатації вразливості: звичайний користувач за допомогою соціальної інженерії змушує менеджера перейти за посиланням, яке автоматично надсилає POST-запит до модуля з довільними параметрами. У ході експерименту було створено понад 200 фальшивих облікових записів, змінено склад когорт та видалено легітимних студентів із груп, що повністю підтвердило критичність виявленого недоліку.

Кількісна оцінка ризику за стандартом CVSS дала базовий бал 8.8 із 10 (рівень HIGH), що свідчить про високу небезпеку вразливості для інформаційної системи закладу освіти. Наслідки експлуатації включають порушення цілісності таких даних, як несанкціоноване створення та видалення облікових записів, потенційний витоків конфіденційної інформації, автоматичне надсилання паролів на зовнішні адреси та загрозу доступності системи, масова генерація облікових записів, що може призвести до перевантаження сервера. Модуль CIST, завдяки виявленій вразливості, фактично виконує функції «руткіт», дозволяючи зловмиснику з низькими привілеями таємно маніпулювати структурою DL системи без відома адміністрації.

Основними причинами виникнення вразливості є: ігнорування розробниками вбудованих механізмів безпеки Moodle системи, хибне розуміння ролі контролю доступу, перевірка capability не захищає від CSRF та відсутність етапу тестування безпеки в циклі розробки модуля.

Для усунення виявленої вразливості необхідно негайно внести зміни до вихідного коду модуля CIST, зокрема додати виклик `require_sesskey()` перед обробкою POST-запитів та включити приховане поле `sesskey` до HTML-форми. Рекомендовано також використовувати стандартний клас moodleform, який автоматично забезпечує CSRF-захист.

Перспективи подальших досліджень охоплюють такі напрямки:

1. Системний аудит безпеки сторонніх модулів Moodle системи – проведення масштабного дослідження популярних модулів із метою виявлення CSRF-вразливостей та інших типів недоліків, пов'язаних із недостатньою перевіркою запитів.

2. Створення методичних рекомендацій для розробників – узагальнення виявлених проблем у вигляді практичного посібника з безпечного програмування для Moodle системи, що включатиме чек-листи, шаблони та приклади коректної реалізації.

3. Дослідження комбінованих атак – аналіз можливості поєднання CSRF-вразливості з іншими векторами атак XSS, SQL-ін'єкції для підвищення рівня загрози та розробка комплексних методів захисту.

Таким чином, виконане дослідження не лише виявило конкретну критичну вразливість у модулі CIST, але й заклало підґрунтя для системного підвищення рівня безпеки всієї системи Moodle шляхом розробки теоретичних, методологічних та практичних засобів, що запобігають виникненню подібних недоліків у майбутньому. Комплексне впровадження запропонованих заходів дозволить суттєво підвищити стійкість інформаційно-освітніх середовищ до кіберзагроз і забезпечити надійний захист даних учасників навчального процесу.

ПОДЯКА

Автор висловлює щиру подяку Заводову Сергію Олександровичу, директору Центру технологій дистанційного навчання Харківського національного університету радіоелектроніки, за плідну співпрацю, всебічне сприяння у проведенні дослідження та надання доступу до тестового середовища DL системи, що дозволило здійснити повноцінну верифікацію виявленої вразливості.

Окрема подяка Ткачову Віталію Миколайовичу, помічнику ректора з питань інформаційних технологій Харківського національного університету радіоелектроніки, за те, що саме він створив необхідні умови для реалізації цього проєкту. Без його активної позиції на початковому етапі проведення дослідження було б неможливим.

Дослідження виконано в межах науково-дослідної роботи кафедри комп'ютерного моделювання та інтелектуальних технологій Харківського національного університету радіоелектроніки за темою «Аналіз та захист систем від шкідливого програмного забезпечення».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OWASP Foundation. (2026). OWASP Cheat Sheet Series: Cross-site request forgery prevention. https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
2. Wang, Z., & Zhang, Y. (2020). CSRF vulnerabilities in modern web applications: A survey. *Journal of Cybersecurity*, *12*(3), 45-58.
3. Barth, A., Jackson, C., & Mitchell, J. C. (2008). Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM Conference on Computer and Communications Security* (pp. 75-88). ACM.
4. Al-Saleh, A. M., & Al-Zoubi, A. S. (2019). A comprehensive review of CSRF attacks and defenses. *International Journal of Network Security*, *21*(4), 612-625.
5. Moodle Docs. (2026). *Security*. https://docs.moodle.org/502/en/Security
6. Smith, J., & Brown, L. (2021). Security vulnerabilities in Moodle plugins: A case study. *Journal of Educational Technology Systems*, *49*(2), 112-130.
7. Jones, P., & Kumar, R. (2020). Privilege escalation in learning management systems. *International Journal of Information Security*, *19*(5), 533-548.
8. Johnson, C. (2022). Real-world CSRF attacks on e-learning platforms. *Security Informatics*, *11*(1), 1-15.
9. Moodle Developer Resource Centre. (2026). *Cross-site request forgery*. https://moodledev.io/general/development/policies/security/crosssite-request-forgery#ensure-your-code-does-not-expose-the-sesskey-inadvertently
10. Anderson, R. (2020). Security engineering: A guide to building dependable distributed systems *(3rd ed.). Wiley.*
11. Kim, T. K., & Lee, H. S. (2021). Effective countermeasures against CSRF in open-source LMS. Computers & Security, 102, *102154.*

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023

Mykola Yatsyk
Ph.D. in Engineering Sciences, Senior Lecturer
Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
ORCID:0000-0002-0814-3909
mykola.yatsyk@nure.ua

MODULE CIST AS ROOTKIT IN MOODLE ENVIRONMENT

Abstract. This paper investigates the Center of Information Systems and Technologies (CIST) module, which was developed for the Distance Learning (DL) system based on the Moodle platform. The CIST module is designed for automatic data synchronization, facilitating student administration via an external API of the information systems and technologies center. Its functionality is protected by a capability granted solely to DL system managers. A thorough analysis of the CIST module's source code has been conducted, revealing a critical vulnerability – the complete absence of Cross-Site Request Forgery (CSRF) protection mechanisms. Specifically, the CIST module does not utilize Moodle's unique security token (sesskey), which is the standard defense against CSRF attacks. This omission allows any authenticated user to trigger privileged requests on behalf of a manager. As a result, an ordinary user without manager rights can lure a manager to a malicious cross-site scripting (XSS) page, which automatically sends a POST request with the required parameters to the CIST synchronization module. In this way, the module functions as a rootkit, enabling an attacker with low privileges to covertly execute privileged operations. The danger is compounded by the fact that the parameters are not properly filtered, allowing the attacker to specify arbitrary POST values, such as targeting all university faculties, which triggers synchronization without any filtering. This leads to unauthorized creation of user accounts with role assignments, arbitrary modification of cohort compositions, removal of legitimate students from groups, and potential leakage of confidential data via automated emails containing passwords. Furthermore, mass account generation can overload the DL server, constituting an additional attack vector. A practical exploitation scenario is described, in which a non-manager user creates a malicious HTML resource within the DL system using JavaScript, and through social engineering, induces a manager to follow a link to this resource, automatically sending a POST request on behalf of the manager to the information systems center. The impact of this attack vector is assessed, including data integrity violations, disruption of the educational process, and the potential use of the module as a rootkit for covertly manipulating the DL system without the knowledge of the administration. The study emphasizes the urgent necessity to implement standard DL system security measures. The results underscore the critical importance of a comprehensive security approach when developing additional modules for popular educational platforms, as even a single minor oversight can lead to serious consequences.

Keywords: Moodle; privilege escalation; rootkit; CSRF, XSRF; XSS; RBAC; DL; CIST.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. OWASP Foundation. (2026). OWASP Cheat Sheet Series: Cross-site request forgery prevention. https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
2. Wang, Z., & Zhang, Y. (2020). CSRF vulnerabilities in modern web applications: A survey. *Journal of Cybersecurity*, *12*(3), 45-58.
3. Barth, A., Jackson, C., & Mitchell, J. C. (2008). Robust defenses for cross-site request forgery. In *Proceedings of the 15th ACM Conference on Computer and Communications Security* (pp. 75-88). ACM.
4. Al-Saleh, A. M., & Al-Zoubi, A. S. (2019). A comprehensive review of CSRF attacks and defenses. *International Journal of Network Security*, *21*(4), 612-625.
5. Moodle Docs. (2026). *Security*. https://docs.moodle.org/502/en/Security
6. Smith, J., & Brown, L. (2021). Security vulnerabilities in Moodle plugins: A case study. *Journal of Educational Technology Systems*, *49*(2), 112-130.
7. Jones, P., & Kumar, R. (2020). Privilege escalation in learning management systems. *International Journal of Information Security*, *19*(5), 533-548.
8. Johnson, C. (2022). Real-world CSRF attacks on e-learning platforms. *Security Informatics*, *11*(1), 1-15.

9. Moodle Developer Resource Centre. (2026). *Cross-site request forgery*. https://moodledev.io/general/development/policies/security/crosssite-request-forgery#ensure-your-code-does-not-expose-the-sesskey-inadvertently
10. Anderson, R. (2020). Security engineering: A guide to building dependable distributed systems *(3rd ed.). Wiley.*
11. Kim, T. K., & Lee, H. S. (2021). Effective countermeasures against CSRF in open-source LMS. Computers & Security, 102, *102154.*

Отримано редакцією журналу / Received: 26.01.26
Прорецензовано / Revised: 06.02.26
Схвалено до друку / Accepted: 24.09.26