УДК 004.056.53, 004.588

Лабунець Павло Юрійович
студент
Київського національного університету імені Тараса Шевченка
Лабунец Павел Юрьевич
студент
Киевского национального университета имени Тараса Шевченко
Labunets Pavlo
Student of the
Taras Shevchenko National University of Kyiv

Слюсар Євген Андрійович
кандидат технічних наук, асистент,
асистент кафедри комп'ютерної інженерії
Київський національний університет імені Тараса Шевченка
Слюсар Евгений Андреевич
кандидат технических наук, ассистент,
ассистент кафедры компьютерной инженерии
Киевский национальный университет имени Тараса Шевченко
Sliusar Ievgen
Candidate of Technical Sciences, Assistant Lecturer
Taras Shevchenko National University of Kyiv

DOI: 10.25313/2520-2057-2021-4-7058

ПЛАТФОРМА АВТОМАТИЗАЦІЇ ДЛЯ ПРАКТИКУМУ З ВИВЧЕННЯ БІНАРНИХ ВРАЗЛИВОСТЕЙ КОМП'ЮТЕРНИХ СИСТЕМ

ПЛАТФОРМА АВТОМАТИЗАЦИИ ДЛЯ ПРАКТИКУМА ПО ИЗУЧЕНИЮ БИНАРНЫХ УЯЗВИМОСТЕЙ КОМПЬЮТЕРНЫХ СИСТЕМ

AN AUTOMATION PLATFORM FOR THE PRACTICUM TO STUDY BINARY VULNERABILITIES OF COMPUTER SYSTEMS

Анотація. *Розроблено платформу-практикум для вивчення бінарних вразливостей з використанням хмарних технологій та з метою застосування в якості лабораторного стенду в вищих навчальних закладах.*
Ключові слова: *вразливість, експлойт, автоматизація, хмарні технології, CTF-випробування.*

Аннотация. *Разработана платформа-практикум для изучения бинарных уязвимостей с использованием облачных технологий для использования в качестве лабораторного стенда в высших учебных заведениях.*
Ключевые слова: *уязвимость, эксплойт, автоматизация, облачные технологии, CTF-испытания.*

Summary. *The practicum-platform for learning binary vulnerabilities has been developed, applying cloud technologies for the purpose of using it as a laboratory stand in higher educational institutions' learning process.*
Key words: *vulnerability, exploit, automation, cloud technologies, CTFchallenge.*

Вступ. Проблема інформаційної безпеки, захисту інформації та надійності програм є одним з основних питань в сучасному світі інформаційних технологій.

Найбільшу небезпеку для комп'ютерних систем на сьогоднішній день становлять бінарні властивості. Так звуться критичні помилки в тих програмних застосунках, які компілюються в машинний код. Експлуатація таких вразливостей дозволяє змусити програму виконати ті дії, можливість яких не була закладена в неї при розробці (аж до виконання довільного набору інструкцій), просто відповідним чином сформувавши вхідні дані. Програмне забезпечення та послідовності дій, що використовують вразливості з метою отримати певну користь називають експлойтами.

Теоретичні знання про вразливості та техніки їх експлуатації, підкріплені практичним досвідом, є важливими не тільки для профільних спеціалістів з інформаційної безпеки, але і для розробників, особливо для тих, що використовують низькорівневі мови програмування.

Проблематика теми та вирішення поставлених задач. Опанувати знання про вразливості та основні техніки їх експлуатації можна за допомогою таких ресурсів, як CTF-випробування [1] (англ. Capture the Flag Challenges). Найпопулярнішими прикладами останніх є веб-ресурси pwnable.kr [2] та hackthebox.eu [3]. Вони пропонують користувачам набори задач, сенс яких полягає в експлуатації спеціально підготовлених вразливих програм задля отримання можливості підвищення привілеїв та читання секретного значення (того самого «флагу») із захищеного файлу. Ці ресурси є доволі корисними з точки зору звичайного користувача, який хоче отримати практичний досвід, проте застосовування їх в якості лабораторного практикуму є недоцільним. Головною причиною є те, що вони пропонують обмежений набір задач, які є ідентичними для всіх користувачів, в той час, як реалізація лабораторного практикуму має забезпечити кожного відносно унікальною задачею задля гарантії дотримання академічної доброчесності та оцінювання саме самостійної роботи студентів.

Крім вищенаведених ресурсів, які, власне, є збірниками задач, на сьогодні доступні і платформи для розгортки власних CTF-випробовувань (наприклад, CTFd [4] та його аналоги). Проте при дослідженні їх можливостей, було зроблено висновок, що вони не мають всіх функцій, які повинна мати описана в даній статті платформа, а саме:

- підтримка автоматичної генерації однотипних, але, водночас, унікальних для кожного користувача (студента) задач;
- автоматизована розгортка середовища за запитом для проведення студентом досліджень та виконання задачі;
- надання як звичайним користувачам, так і адміністратору курсу зручного веб-інтерфейсу;
- можливість розширення набору та побудови власних шаблонів випробувань з підтримкою майбутньої генерації на їх основі унікальних задач;
- функціонал для інтеграції в інфраструктуру Університету (так як дана платформа реалізується в першу чергу для Київського національного університету імені Тараса Шевченка, то мова йде саме про його інфраструктуру: хмарні сервіси на основі OpenStack [5] та система управління курсами Moodle [6]).

Схожа проблематика була розглянута в роботі [1]: там теж було розглянуто реалізацію CTF-випробування в якості лабораторного практикуму та піднято питання забезпечення кожного учасника унікальною задачею. Проте в вищенаведеній статті не пропонується рішень щодо інтеграції з системами хмарної інфраструктури та управління курсами, а запропоновані варіанти генерації унікальних завдань лише посилаються на інші розробки, які є прив'язаними до конкретних продуктів та реалізовані за доволі громіздкими схемами, що можуть значно ускладнити експлуатацію платформи.

Розроблена платформа являє собою веб-додаток, написаний на мові програмування Python з використанням фреймворку Django [7]. Вона реалізує інтерфейс звичайного користувача (студента) та адміністратора курсу (викладача).

Авторизувавшись, студент може переглядати список доступних випробувань, власний прогрес, активувати собі задачі та здавати завдання шляхом введення у веб-форму секретного ключа, який він отримує так само, як отримують «флаг» в звичайних CTF-випробуваннях.

Адміністратор курсу через веб-інтерфейс може переглядати та модифікувати списки студентів, активованих задач і шаблонів завдань, а також підтверджувати факт виконання задачі студентом. Крім того, для зручності роботи було реалізовано взаємодію платформи з системою Moodle через REST API. Це дозволяє імпортувати список користувачів з Moodle до бази даних вебдодатку шляхом простого введення в форму посилання на відповідний курс та реквізитів для авторизації в системі Moodle та автоматично виставляти оцінки при підтвердженні виконання задачі.

Генерацію унікальних задач реалізовано шляхом параметризації вихідних кодів програм з вразливостями та декларування в окремому файлі параметрів, яким присвоюються випадкові значення з вказаного набору чи діапазону та які потім використовуються при конфігурації користувацького середовища. Загалом, шаблон випробування — це директорія, в якій наявні вихідні коди вразливої програми, описи конфігурації середовища у форматі Ansible Playbook та файл, що описує параметри та їх можливі значення.

Генерацію завдання для кожного студенти описати наступною послідовністю дій (схематично процес зображено на рисунку 1):

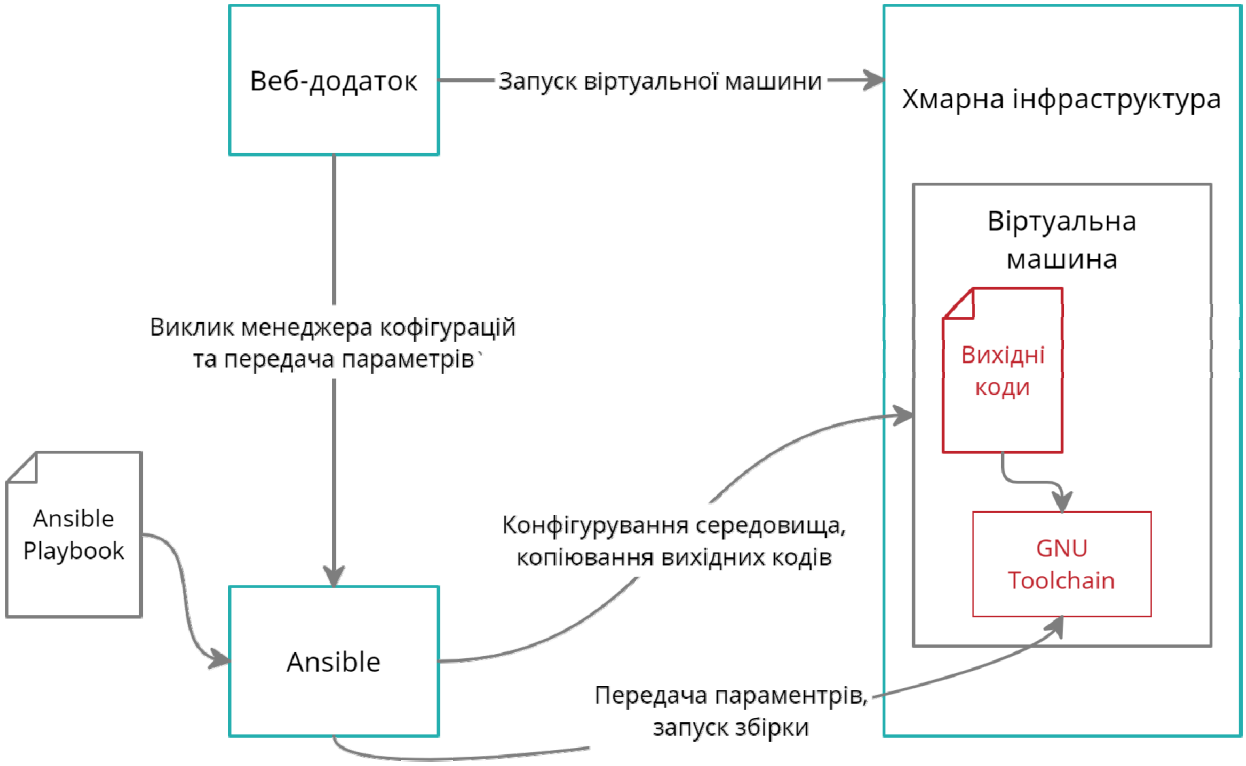

Рис. 1. Процес генерації завдання

1) Студент натискає на кнопку «Активувати задачу».
2) В хмарній інфраструктурі створюється віртуальна машина, яка і буде середовищем для виконання завдання.
3) Випадковим чином встановлюються значення параметрів, що відповідають за унікальність задачі.
4) За допомогою програми Ansible відбувається конфігурування середовища на основі опису в шаблоні проекту та переданих параметрів.
5) З метою компіляції вразливої програми Ansible, в свою чергу, використовує засоби автоматизації збірки GNU Toolchain, також передаючи їм згенеровані параметри.
6) Після успішного виконання всіх дій користувачу надається можливість для підключення до віртуальної машини за протоколом SSH.

Крім самої платформи, було також розроблено 3 види випробувань різного рівня складності. Задачі першого типу включали в себе експлуатацію однієї з чотирьох вразливостей: переповнення буфера в сегменті стеку [8], вразливість форматного рядка [8], вразливість звернення до звільненої пам'яті [9] та подвійне звільнення пам'яті [10] в умовах відсутності будь-яких механізмів захисту. В випробуваннях другого та третього типів було поставлено задачу експлуатації вразливості переповнення буфера або звернення до звільненої пам'яті в умовах дії механізмів DEP [9] та повноцінного набору механізмів захисту ОС Linux (ASLR [9], DEP, стекові індикатори [9]) відповідно. Шляхом параметризації вихідних кодів було забезпечено можливість появи різних вразливостей та різних умов (адрес сегментів пам'яті, довжин буферів, тощо) для кожного конкретного випадку.

Висновки. Аналіз сучасних платформ для реалізації CTF-випробувань показав наявність технічних обмежень існуючих рішень з точки зору їх застосування в навчальному процесі вищих навчальних закладів та дозволив сформулювати вимоги до архітектури програмного комплексу, яка шляхом застосування хмарних технологій OpenStack, засобів автоматизації збірки GNU Toolchain та програми-менеджера конфігурацій Ansible дозволяє усунути обмеження.

Згідно зі сформульованими вимогами було реалізовано систему, яка автоматизує практикум для вивчення бінарних вразливостей та реалізує можливість інтеграції в інфраструктуру Університету, надає зручний інтерфейс для користувачів та забезпечує можливість генерації унікальних задач за шаблонами.

Платформу розроблено з оглядом на подальше розширення функціоналу та масштабування як за кількістю користувачів, так і за набором доступних задач. Це гарантує зручність в її використанні як викладачами, так і студентами.

Література

1. Vykopal Jan & Švábenský Valdemar & Chang Ee-Chien. (2020). Benefits and Pitfalls of Using Capture the Flag Games in University Courses. P. 752–758. doi: 10.1145/3328778.3366893.
2. Pwnable.kr. URL: pwnable.kr
3. Hack The Box. URL: hackthebox.eu
4. Karagiannis S., Maragkos-Belmpas E., Magkos E. (2020). An Analysis and Evaluation of Open Source Capture the Flag Platforms as Cybersecurity e-Learning Tools. In: Drevin L., Von Solms S., Theocharidou M. (eds) Information Security Education. Information Security in Action. WISE 2020. IFIP Advances in Information and Communication Technology. Vol 579. Springer, Cham. doi: 10.1007/978-3-030-59291-2_5
5. Tiago Rosado and Jorge Bernardino. (2014). An overview of openstack architecture. In Proceedings of the 18th International Database Engineering & Applications Symposium (IDEAS '14). Association for Computing Machinery, New York, NY, USA. P. 366–367. doi: 10.1145/2628194.2628195
6. Kautsar I. A., Musashi Y., Kubota S. and Sugitani K. (2015). Synchronizing learning material on Moodle and lecture based supportive tool: The REST based approach // International Conference on Information & Communication Technology and Systems (ICTS), Surabaya, Indonesia. P. 187–192. doi: 10.1109/ICTS.2015.7379896
7. Django. URL: djangoproject.com
8. Erickson J. Hacking: The Art of Exploitation Second Edition / Jon Erickson. No Starch Press, 2008. 488 p.
9. Tanenbaum A. Modern Operating Systems / Andrew Tanenbaum, Herbert Boss. Prentice Hall, 2014. 1101 p.
10. Chae S., Jin H., Park M. C. and Lee D. H. (2020). «HS-Pilot: Heap Security Evaluation Tool Model Based on Atomic Heap Interaction» in IEEE Access. Vol. 8. P. 201914–201924. doi: 10.1109/ACCESS.2020.3036118