Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji
Sprawozdanie z projektu

HONEYPOTY W ŚRODOWISKACH CHMUROWYCH

KONFIGURACJA ŚRODOWISKA T-POT

Autorzy:
Tomasz Gudyka, Kamil Kałużny

Kwiecień 2025

Spis treści

1 Wprowadzenie

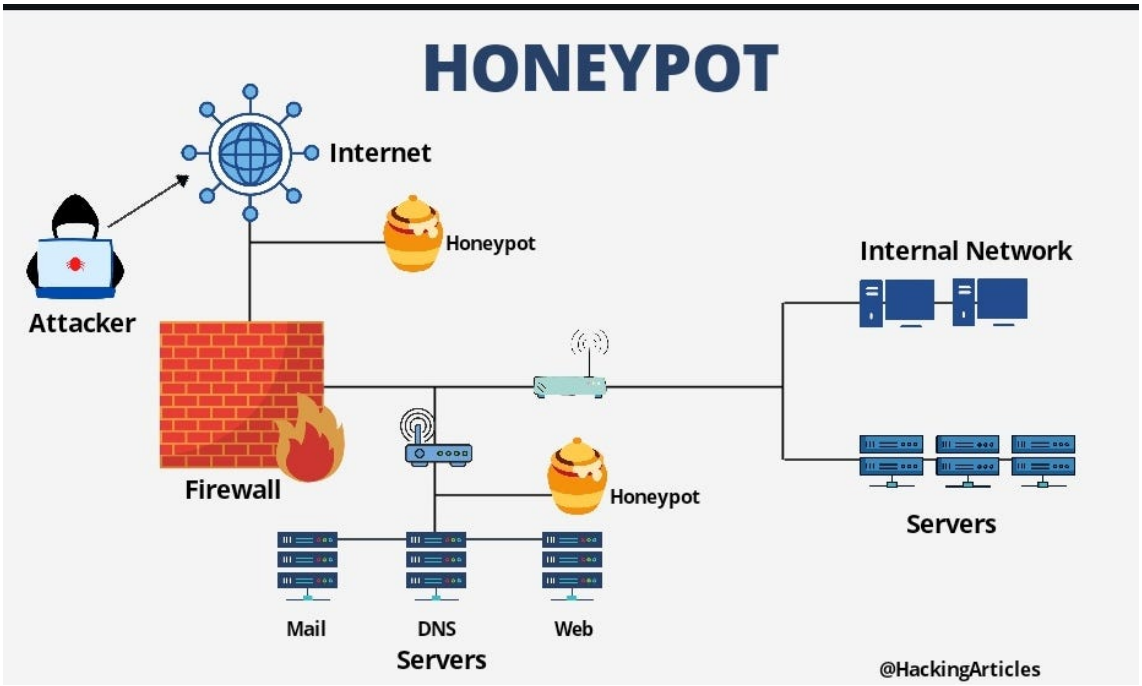
W dobie narastających zagrożeń cybernetycznych organizacje poszukują kreatywnych metod obrony i detekcji ataków. Jednym z zaawansowanych narzędzi w arsenale obrońców jest honeypot – system-pułapka zaprojektowany do zwabiania atakujących. Celem niniejszego sprawozdania jest szczegółowe omówienie koncepcji honeypotów, ich rodzajów i zastosowań, ze szczególnym uwzględnieniem środowisk chmurowych (takich jak AWS, Azure, GCP).

W dalszej części pracy skupimy się na platformie T-Pot, będącej rozbudowanym, zintegrowanym środowiskiem honeypotów. Opiszemy charakterystykę T-Pot, jej funkcje, wymagania, zalety i wady, a także przedstawimy proces wdrożenia z wykorzystaniem Dockera krok po kroku. Zaprezentowane zostaną przykłady rzeczywistych ataków wychwyconych przez T-Pot (np. brute-force, exploity, malware) oraz sposób, w jaki system ten gromadzi i wizualizuje logi za pomocą stosu ELK (Elasticsearch, Logstash, Kibana).

2 Pojęcie honeypota - definicja

Honeypot (z ang. „garnek miodu") to w kontekście bezpieczeństwa IT celowo wystawiony system lub zasób, który imituje podatne usługi lub dane w celu przyciągnięcia potencjalnych napastników[1]. Innymi słowy, jest to wabiąca pułapka: z pozoru atrakcyjny cel, który ma odwrócić uwagę atakującego od prawdziwych systemów produkcyjnych i umożliwić administratorom sieci obserwację działań intruza[2]. Honeypot wygląda dla hakera jak normalny serwer czy aplikacja (np. baza danych, serwer WWW, maszyna z usługą SSH), jednak został specjalnie przygotowany do monitorowania i rejestrowania wszelkich prób włamania. Dzięki temu, gdy atakujący podejmie interakcję z honeypotem, jego aktywność jest śledzona i analizowana w kontrolowanym środowisku.

Warto zauważyć, że koncepcja honeypota wywodzi się z terminologii szpiegowskiej – „honeypot" lub „honeytrap" oznaczały w żargonie służb pułapkę z udziałem przynęty (np. agentki), mającą skompromitować wroga[3]. W cyberbezpieczeństwie termin ten został zaadaptowany do opisania systemów pełniących rolę przynęty na cyberprzestępców. Honeypot jest więc fikcyjnym słabym ogniwem umyślnie wprowadzonym do infrastruktury, aby wykrywać ataki i badać techniki hakerskie bez narażania właściwych zasobów.



Rysunek 1: Schemat działania honeypota w przykładowym środowisku

3 Klasyfikacja honeypotów

Honeypoty można podzielić według różnych kryteriów. Dwie najważniejsze osie klasyfikacji to: poziom interakcji oferowany atakującemu oraz przeznaczenie honeypota (środowisko produkcyjne vs badawcze).

3.1 Poziom interakcji (niski vs wysoki)

Jednym z podstawowych podziałów jest rozróżnienie na honeypoty niskiej interakcji i wysokiej interakcji. Podział ten opisuje, jak bardzo realistyczną i rozbudowaną interakcję z systemem oferujemy atakującemu[1][2]:

- **Honeypot niskiego poziomu interakcji** – to zazwyczaj symulacja wybranych usług lub protokołów. Są one stosunkowo łatwe do skonfigurowania i utrzymania, ale zapewniają ograniczony zakres interakcji z intruzem. Emulowane są tylko pewne odpowiedzi na zapytania (np. prosty serwer FTP lub HTTP udający podatną usługę). Taki honeypot nie posiada pełnego systemu operacyjnego w tle – imituje jedynie powierzchowne zachowanie usługi. Zaletą jest niski narzut zasobów i mniejsze ryzyko, natomiast minusem ograniczone informacje, jakie można zebrać o atakującym (napastnik szybko odkryje, że środowisko jest sztuczne lub nie oferuje głębszej możliwości penetracji).
- **Honeypot wysokiego poziomu interakcji** - to przeciwieństwo powyższego: oferuje intruzowi pełne, prawdziwe środowisko operacyjne do interakcji. Może to być rzeczywisty system (lub wirtualna maszyna) z działającym systemem operacyjnym i usługami, które atakujący może próbować kompromitować. Taki honeypot jest znacznie bardziej wymagający w utrzymaniu (wymaga więcej zasobów oraz ciągłego nadzoru), ale w zamian pozwala zaobserwować pełen

wektor ataku – od wykorzystania podatności po działania po włamaniu. Honeypoty wysokiej interakcji potrafią zatrzymać atakującego na dłużej, dając zespołowi bezpieczeństwa więcej czasu na obserwację jego technik oraz ewentualną reakcję. Istnieje też pojęcie honeypota „czystego" (pure honeypot), który oznacza pełnowartościowy system produkcyjny specjalnie naszpikowany czujnikami monitorującymi (np. prawdziwy serwer z pozornie ważnymi danymi, służący wyłącznie jako przynęta). W praktyce jednak od tworzenia czystych honeypotów częściej stosuje się wyizolowane środowiska wysokiej interakcji (np. odseparowane VM) ze względów bezpieczeństwa.

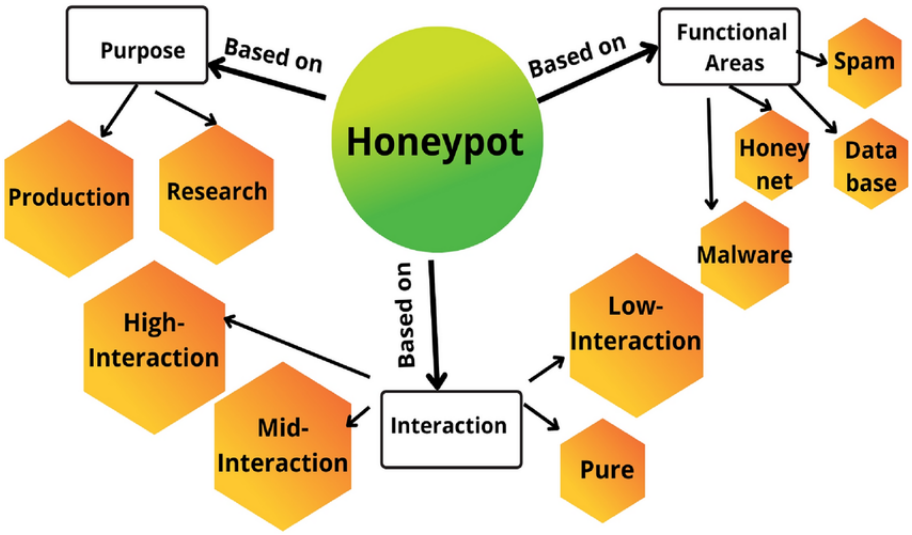
Spotyka się również pojęcie honeypotów średniej interakcji – stanowią one kompromis, emulując niektóre aspekty systemu (zwłaszcza warstwy aplikacyjnej) bez pełnego systemu operacyjnego. Ich celem jest np. wprowadzenie pewnego opóźnienia czy zamieszania dla atakującego (udawanie bardziej złożonego systemu), przy mniejszym ryzyku niż w przypadku pełnych honeypotów.

3.2 Przeznaczenie (produkcyjne vs badawcze)

Drugim ważnym kryterium jest przeznaczenie honeypota – wyróżnia się honeypoty produkcyjne oraz badawcze[2]:

- **Honeypot produkcyjny** - wykorzystywany jest jako dodatkowa warstwa obrony w sieci firmowej. Taki honeypot stawia się obok prawdziwych serwerów produkcyjnych (np. w tej samej podsieci) i uruchamia się na nim te same usługi co na produkcji. Jego głównym celem jest detekcja i odwrócenie uwagi: jeśli intruz zaatakuje sieć, honeypot produkcyjny ma za zadanie przyciągnąć go i umożliwić wykrycie włamania zanim zostaną naruszone krytyczne systemy. Krótko mówiąc, honeypot produkcyjny pełni rolę „alarmu w sieci" – sygnalizuje obecność intruza i zwodzi go, by tracił czas na fałszywym celu, podczas gdy prawdziwe systemy pozostają nienaruszone.

- **Honeypot badawczy** - służy głównie do celów analitycznych i naukowych. Tego typu honeypoty wystawia się często w internecie (np. w DMZ lub chmurze), aby przyciągały ataki z całego świata. Ich zadaniem nie jest ochrona konkretnej usługi, lecz zbieranie informacji o metodach działania napastników w rzeczywistych warunkach. Honeypoty badawcze pozwalają obserwować ataki w skali globalnej i dostarczają danych, które pomagają analitykom bezpieczeństwa opracowywać lepsze mechanizmy obronne (np. tworzyć sygnatury dla IDS/IPS, priorytetyzować poprawki bezpieczeństwa na podstawie faktycznie wykorzystywanych exploitów itp.). Często są elementem projektów badawczych, prowadzonych np. przez uczelnie lub firmy bezpieczeństwa, i bywają skoordynowane w rozległe sieci honeypotów (honeynety) w wielu lokalizacjach.

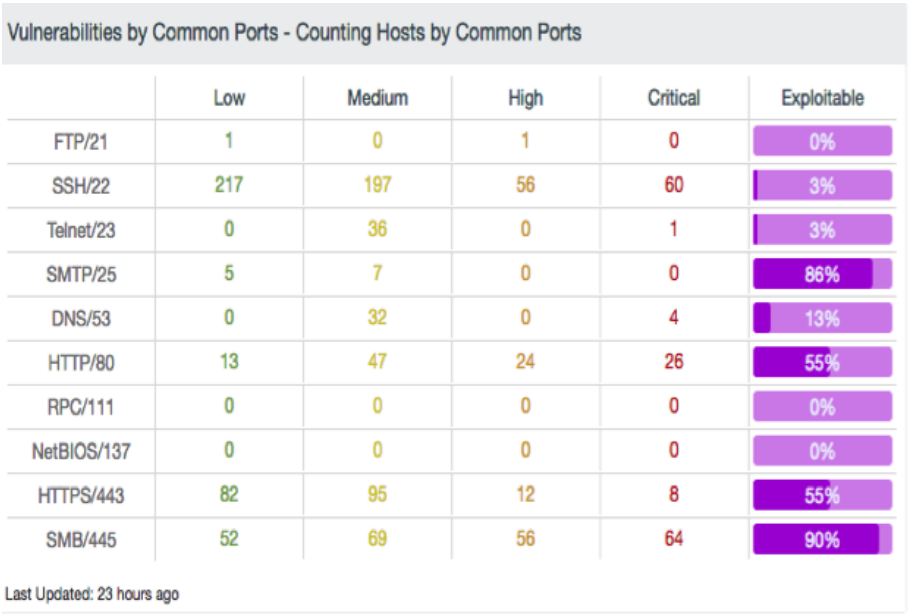


Rysunek 2: Schemat klasyfikacji honeypotów

4 Cele i zastosowania honeypotów

Głównym celem stosowania honeypotów jest uzyskanie przewagi informacyjnej nad atakującymi. Poprzez zwabienie hakera do kontrolowanego środowiska, obrońcy mogą zbierać cenne dane wywiadowcze o nowych zagrożeniach, technikach i narzędziach używanych przez napastników. Honeypot działa jak laboratoryjne stanowisko badawcze: rejestruje kroki intruza, co pozwala specjalistom bezpieczeństwa lepiej zrozumieć taktyki i techniki atakujących. Dzięki temu możliwe jest opracowanie skuteczniejszych metod obrony w przyszłości oraz szybkie załatanie wykrytych słabości (jeśli honeypot symulował np. konkretną aplikację, którą firma używa, i zaobserwowano na nim nowy exploit).[1]

Innym istotnym zastosowaniem honeypotów jest wykrywanie włamań i alarmowanie. Ponieważ systemy-pułapki nie pełnią produkcyjnych ról, każda próba komunikacji z honeypotem może być traktowana jako podejrzana. Honeypot pełni zatem rolę sensora intrusion detection: w momencie ataku może powiadomić administratorów o naruszeniu bezpieczeństwa. Działa to szczególnie dobrze w przypadku honeypotów produkcyjnych w sieci wewnętrznej – jeśli ktoś próbuje się do nich dostać, najpewniej oznacza to, że omija zabezpieczenia lub to osoba z wewnątrz o nieczystych zamiarach.



Vulnerabilities by Common Ports - Counting Hosts by Common Ports

	Low	Medium	High	Critical	Exploitable
FTP/21	1	0	1	0	0%
SSH/22	217	197	56	60	3%
Telnet/23	0	36	0	1	3%
SMTP/25	5	7	0	0	86%
DNS/53	0	32	0	4	13%
HTTP/80	13	47	24	26	55%
RPC/111	0	0	0	0	0%
NetBIOS/137	0	0	0	0	0%
HTTPS/443	82	95	12	8	55%
SMB/445	52	69	56	64	90%

Last Updated: 23 hours ago

Rysunek 3: Popularne usługi pod względem podatności

Honeypoty mogą także odciążać inne systemy bezpieczeństwa. Atakujący, skupiając się na łatwym celu, jakim jest honeypot, traci czas i zasoby na bezskuteczne ataki, podczas gdy systemy produkcyjne działają niezakłócone. W pewnym sensie honeypot „kupuje czas" zespołowi obronnemu – pozwala na wykrycie ataku we wczesnej fazie oraz potencjalnie opóźnia postęp intruza w głąb sieci.[2]

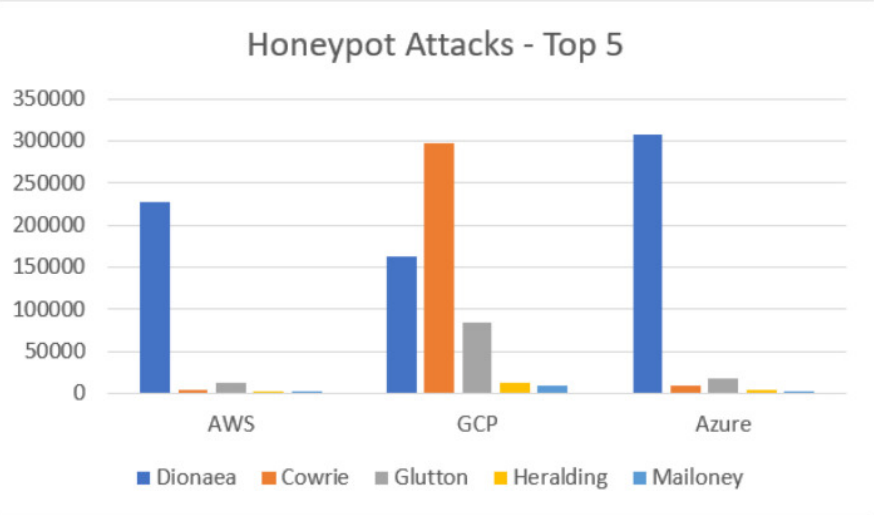
Należy jednak pamiętać, że honeypoty nie zastępują tradycyjnych zabezpieczeń, a raczej je uzupełniają. Ich skuteczność zależy od właściwej konfiguracji i ciągłego nadzoru. Źle zabezpieczony honeypot może zostać przez atakującego wykorzystany jako platforma do atakowania innych systemów. Istnieje też ryzyko, że zaawansowany przeciwnik zorientuje się, iż ma do czynienia z pułapką – dlatego honeypot powinien być możliwie wiarygodny.

5 Honeypoty w środowiskach chmurowych (AWS, Azure, GCP)

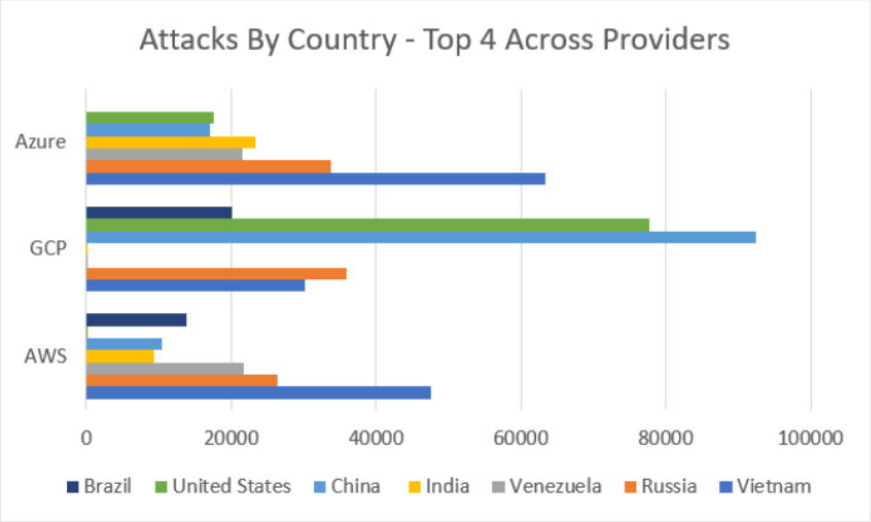
Migracja infrastruktury do chmury nie eliminuje zagrożeń – przeciwnie, uwidacznia nowe wektory ataku. W związku z tym honeypoty znalazły zastosowanie także w środowiskach chmurowych. Chmura publiczna (np. Amazon Web Services, Microsoft Azure czy Google Cloud Platform) oferuje łatwość wdrożenia serwera honeypot bez potrzeby inwestycji w sprzęt. Można szybko uruchomić maszynę wirtualną z odpowiednio skonfigurowanym systemem pułapkowym i wystawić ją na adres publiczny, uzyskując globalny „wabik" na cyberprzestępców. Takie podejście jest często wykorzystywane w projektach badawczych do obserwacji aktualnych trendów ataków w internecie.

Badania porównawcze wskazują, że ataki na honeypoty mogą różnić się w zależności od platformy chmurowej. W jednym z eksperymentów jednocześnie wdrożono

identyczne honeypoty w kilku regionach na AWS, Azure i GCP, zbierając logi przez 3 tygodnie. Analiza wykazała znaczące zróżnicowanie aktywności atakujących w zależności od dostawcy: różnice występowały zarówno w wolumenie i pochodzeniu geograficznym ruchu, jak i w najczęściej atakowanych usługach oraz wykorzystanych podatnościach. Przykładowo, zaobserwowano wzmożone próby ataków na popularne usługi zdalnego dostępu (np. protokół RDP) – co przypisano zwiększonej zależności od tych usług w okresie pracy zdalnej podczas pandemii COVID-19[4]. Źródła ataków pochodziły zarówno z krajów tradycyjnie kojarzonych z aktywnością botnetów (Chiny, Rosja, USA), jak i z mniej oczywistych lokalizacji (np. Wietnam, Indie, Wenezuela). Wniosek płynący z tych obserwacji jest taki, że honeypoty w chmurze dostarczają unikalnego wglądu w ekosystem zagrożeń skierowanych na infrastrukturę cloud – różni aktorzy mogą wybierać różne cele w zależności od platformy.



(a) Honeypot Attacks Top 5 Comparison



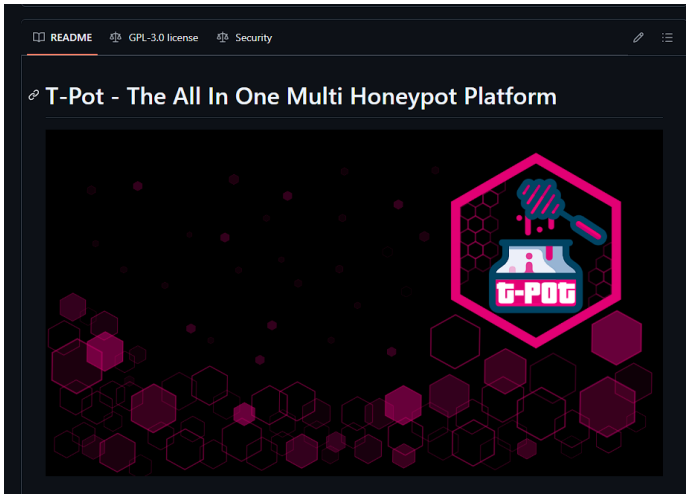
(b) Attacks by Country Top 4 Comparison

Rysunek 4: Porównanie usługodawców chmurowych pod względem ataków i krajów

Przy wdrażaniu honeypota w chmurze należy pamiętać o bezpieczeństwie konfiguracji. Ważne jest zastosowanie odpowiednich reguł firewall (Security Groups/Network Security Groups), by kontrolować ruch sieciowy – zwykle honeypot ma akceptować połączenia przychodzące na różne porty, ale warto ograniczyć ruch wychodzący (aby zapobiec ewentualnemu wykorzystaniu honeypota do ataków DDoS czy spamowania, gdyby został on kompletnie przejęty przez wroga). Ponadto honeypot w chmurze powinien być odizolowany od produkcyjnych zasobów (np. w oddzielnej sieci VPC/VNet), aby nawet w razie kompromitacji nie stanowił zagrożenia dla reszty infrastruktury. Cloudowe honeypoty dostarczają ogromu danych, ale generują też koszty (opłaty za czas pracy VM, transfer danych, przechowywanie logów). Mimo to, dla wielu organizacji i badaczy, uruchomienie honeypota w AWS/Azure/GCP jest szybkim sposobem na zdobycie informacji o aktualnych zagrożeniach, jakie czyhają na ich chmurowe zasoby.

6 Platforma T-Pot – All-in-One Honeypot

6.1 Charakterystyka i funkcjonalności T-Pot



Rysunek 5: Repozytorium t-pot'a na platformie github

T-Pot to rozbudowana platforma honeypot typu „wszystko w jednym" (all-in-one), rozwijana jako projekt open-source przez zespół Deutsche Telekom Security. Jej celem jest zintegrowanie wielu różnych honeypotów w ramach jednej infrastruktury, co umożliwia jednoczesne monitorowanie rozmaitych wektorów ataku i scentralizowaną analizę zebranych danych. T-Pot można określić jako honeynet w pudełku – pakiet ponad 20 honeypotów obsługiwanych w jednym systemie, wyposażonym dodatkowo w narzędzia wizualizacji i analizy ataków[6][10]. Kluczowe cechy i funkcje platformy T-Pot to m.in.:

- **Integracja wielu typów honeypotów:** W ramach T-Pot równolegle działają moduły emulujące różne usługi i protokoły. Są tam honeypoty niskiej i średniej interakcji (np. Dionaea łapiąca malware przez usługi SMB, FTP, HTTP; ElasticPot udający API Elasticsearch; Cowrie udający powłokę SSH/Telnet; Conpot symulujący urządzenia SCADA/ICS; Mailoney udający serwer

SMTP i inne) oraz honeypoty wysokiej interakcji (np. moduł Honeytrap mogący dynamicznie uruchamiać różne usługi). T-Pot integruje też popularne honeypoty wymieniane w literaturze, takie jak Kippo/Cowrie (SSH), Dionaea (wieloprotokołowy malware honeypot) czy Glastopf (pułapka webowa) – wszystko to działa jednocześnie, stanowiąc szerokie spektrum przynęt dla atakujących. Dzięki temu pojedyncza instancja T-Pot może rejestrować bardzo różnorodne ataki, od prób brute-force na SSH po exploity aplikacyjne czy ataki na usługi IoT.

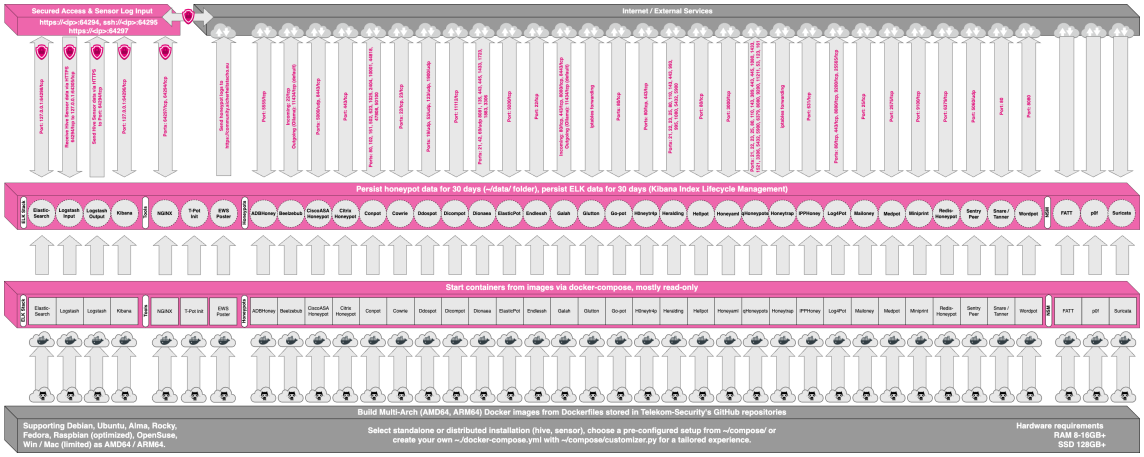
- **Wbudowany system detekcji i analizy:** T-Pot zawiera zintegrowany IDS/IPS Suricata, który monitoruje ruch sieciowy wokół honeypotów i generuje alerty bezpieczeństwa (np. zidentyfikowane sygnatury ataków, CVE exploitów) widoczne w konsoli i dashboardach. Ponadto T-Pot wykorzystuje mechanizm p0f i inne narzędzia fingerprintingu do pasywnego rozpoznawania atakującego (np. systemu operacyjnego napastnika). Platforma umożliwia także automatyczne przechwytywanie złośliwych plików (sample malware) dostarczanych przez intruzów – np. honeypot Dionaea zapisuje binaria malware pobierane podczas ataku, co daje możliwość późniejszej analizy tych plików.

- **Scentralizowane logowanie i wizualizacja (stack ELK):** Jedną z największych zalet T-Pot jest gotowa integracja z Elastic Stack (ELK). Wszystkie honeypoty logują zdarzenia, które są zbierane i przetwarzane (przez Logstash) oraz składowane w bazie Elasticsearch, a następnie udostępniane do przeglądania w interfejsie Kibana. Użytkownik otrzymuje bogaty zestaw predefiniowanych dashboardów i narzędzi analitycznych – od interaktywnych wykresów i map ataków po szczegółowe logi z poszczególnych usług. T-Pot udostępnia np. panel ogólny z mapą świata pokazującą geolokację źródeł ataków w czasie rzeczywistym, statystyki najczęstszych celów i technik ataku, jak również osobne dashboardy dla konkretnych honeypotów (np. szczegółowa analiza ataków na ElasticPot czy Cowrie).

- **Łatwa dystrybucja i skalowalność dzięki konteneryzacji:** Architektura T-Pot opiera się na technologii Docker – każdy honeypot oraz usługa to odrębny kontener Docker, zarządzany wspólnie poprzez pliki konfiguracyjne (docker-compose). Takie podejście zapewnia izolację między komponentami oraz ułatwia wdrażanie i aktualizacje. Dzięki konteneryzacji T-Pot może działać na różnych dystrybucjach Linuksa (Ubuntu, Debian, Fedora/AlmaLinux, openSUSE itp.), a nawet w ograniczonym zakresie na Windows czy macOS (np. w Docker Desktop) – choć natywne wsparcie dla systemów innych niż Linux jest ograniczone. Docker pozwala również na skalowanie: T-Pot może pracować na pojedynczej maszynie jako instancja standalone lub zostać rozproszony na wiele węzłów (np. instancje sensorów przesyłające zebrane logi do centralnego serwera Elastic).

- **Narzędzia dodatkowe dla analityków:** Poza Kibaną, T-Pot zawiera kilka przydatnych narzędzi w interfejsie webowym, m.in. CyberChef (interaktywne narzędzie do dekodowania/analizy danych), Spiderfoot (platforma do OSINT i profilowania zagrożeń) czy ElasticVue (frontend do eksploracji bazy Elasticsearch). Do dyspozycji jest również „attack map” (mapa ataków) i inne wizu-

alizacje czyniące z T-Pot nie tylko zbieracz danych, ale i wygodne środowisko dla analityka.



Rysunek 6: Usługi (wraz z numerami portów) dostępne w środowisku t-pot

6.2 Wymagania systemowe T-Pot

Ze względu na swoją kompleksowość, T-Pot ma dość znaczące wymagania sprzętowe i środowiskowe. Twórcy zalecają, aby system przeznaczony pod T-Pot dysponował minimum 8 GB pamięci RAM (zalecane 16 GB) oraz co najmniej 128 GB wolnej przestrzeni dyskowej. Wynika to z faktu, że uruchomionych będzie kilkadziesiąt kontenerów Docker – każdy honeypot to osobny kontener, do tego dochodzą kontenery stacku ELK (Elasticsearch potrafi zużywać dużo RAM) oraz innych usług pomocniczych. Na dysku przechowywane są zarówno logi (w bazie Elasticsearch), jak i próbki malware pobrane przez honeypoty, stąd wymóg odpowiednio dużej przestrzeni.[10]

Jeśli chodzi o system operacyjny, T-Pot jest dystrybuowany w formie skryptów instalacyjnych kompatybilnych z wieloma dystrybucjami Linuksa. Oficjalnie wspierane są m.in. Ubuntu (LTS), Debian, AlmaLinux (klon Red Hat), openSUSE.

T-Pot może działać na sprzęcie fizycznym, maszynach wirtualnych, a także w chmurze – byle spełnione były powyższe parametry. Należy jednak pamiętać, że wydajność i stabilność rozwiązania będzie lepsza na dedykowanym sprzęcie lub wysokiej klasy VM. Twórcy wskazują, że w przypadku chmury mogą być potrzebne pewne dodatkowe dostosowania (np. inne limity obsługi połączeń sieciowych).

6.3 Zalety i wady T-Pot

Zalety T-Pot: Platforma T-Pot zyskała popularność dzięki licznym korzyściom dla użytkowników chcących prowadzić honeypot w profesjonalny sposób:

1. **Kompleksowość i różnorodność:** Największą zaletą jest integracja wielu honeypotów w jednym rozwiązaniu. Użytkownik otrzymuje „pakiet" kilkunastu–kilkudziesięciu pułapek, co pozwala monitorować różne rodzaje ataków jednocześnie z jednego systemu.

2. **Gotowa infrastruktura analityczna (ELK):** T-Pot oferuje „od ręki" zaawansowaną analitykę zebranych danych. Dzięki wbudowanej integracji z ELK Stack, użytkownik ma dostęp do czytelnych dashboardów, wykresów i map ataków niemal od razu po instalacji.

3. **Łatwa instalacja i aktualizacja:** Mimo rozbudowania, T-Pot jest stosunkowo prosty we wdrożeniu – dostarczany jest z automatycznym instalatorem. Proces instalacji sprowadza się do uruchomienia jednego skryptu, który pobiera wszystkie potrzebne komponenty i konfiguruje środowisko. Nie jest wymagana specjalistyczna wiedza o Dockerze czy o poszczególnych honeypotach – domyślna konfiguracja działa od razu po instalacji. Aktualizacje platformy również są zautomatyzowane (skrypt update). T-Pot jest projektem open-source (dostępny bezpłatnie), co oznacza brak kosztów licencji – Community Edition może być używana swobodnie.

4. **Możliwość customowej konfiguracji:** Platforma T-Pot charakteryzuje się wysoką elastycznością konfiguracji, umożliwiając uruchamianie wybranych modułów honeypot zgodnie z potrzebami użytkownika. W ramach skryptu Python możliwe jest precyzyjne określenie, które pułapki (np. HTTP, SSH, Telnet, moduły IoT) mają zostać załadowane, co pozwala na optymalizację zużycia zasobów systemowych.

Wady T-Pot: Pomimo wielu zalet, platforma T-Pot ma również pewne ograniczenia i wady, o których warto wspomnieć:

1. **Duże zapotrzebowanie na zasoby:** Jak już omówiono w wymaganiach, T-Pot wymaga dość mocnej maszyny. Ponadto, w miarę gromadzenia logów, baza Elasticsearch może zajmować coraz więcej miejsca – administracja T-Pot wymaga więc monitorowania wykorzystania dysku i ewentualnego archiwizowania starszych danych.

2. **Ograniczone wsparcie wieloplatformowe:** Choć T-Pot może działać na różnych OS, de facto pełna funkcjonalność dostępna jest tylko na Linuxie. Wersje na Windows/macOS (np. poprzez Docker Desktop) są okrojone – nie wszystkie honeypoty zadziałają poprawnie.

3. **Złożoność i potencjalne błędy:** Mimo że instalacja jest uproszczona, T-Pot pozostaje złożonym systemem. Istnieje więcej punktów potencjalnej awarii – np. problemy ze startem któregoś kontenera, konflikty portów, obciążenie powodujące niestabilność ElasticSearch itp. Użytkownik powinien mieć przynajmniej podstawową wiedzę o Dockerze, by diagnozować ewentualne problemy.

4. **Ryzyko wykrycia przez atakującego:** To bardziej teoretyczna wada – jednak warto zauważyć, że zestaw T-Pot ma pewne znaki szczególne, które mogą zostać rozpoznane przez zaawansowanego intruza. Jeśli ktoś specjalnie poszukuje honeypotów może np. poznać po typowych otwartych portach lub sygnaturach bannerów, że ma do czynienia z T-Pot. Dodatkowo, fakt uruchomienia tak wielu usług na jednej maszynie może wzbudzić podejrzenia (rzadko kiedy prawdziwy serwer oferuje jednocześnie otwarty MongoDB, SSH, SMB, HTTP, SMTP, ICS itp.). Oczywiście, dla masowego skanera botnetowego nie

ma to znaczenia – ale jeśli badamy aktywnie grupy APT, one mogą potrafić zidentyfikować honeypot i unikać ujawniania swoich najcenniejszych exploitów.

6.4 Implementacja T-Pot z Dockerem – krok po kroku

Jednym z założeń projektowych T-Pot jest łatwość wdrożenia. Poniżej przedstawiono krok po kroku proces instalacji i uruchomienia:

1. **Przygotowanie środowiska:** Najpierw należy przygotować czysty system Linux spełniający wymagania (np. Ubuntu Server 22.04 LTS x64, 8 GB RAM, 128 GB dysku). W środowisku chmurowym tworzymy nową instancję o odpowiednich parametrach i konfigurujemy reguły firewall, by otworzyć niezbędne porty honeypotów na świat. W rzeczywistości na maszynie posiadającej 8GB RAM'u możliwe jest uruchomienie pełnej wersji T-Pot. Pojemność dysku wpływa na jedynie na możliwą ilość przechowywanych logów - już 60GB jest wystarczające na przechowanie miesięcznych logów.

2. **Instalacja zależności:** T-Pot instalowany jest poprzez skrypt bash, który sam zadba o instalację Dockera i innych komponentów. Niemniej jednak, dobrze jest zaktualizować wszystkie pakiety systemowe, np. na Ubuntu/Debian:

```
sudo apt-get update && sudo apt-get install -y curl git.
```

3. **Pobranie kodu T-Pot:** Prostsza metoda polega na pobraniu repozytorium z github i wypakowwaniu go:

```
git clone https://github.com/telekom-security/tpotce
```

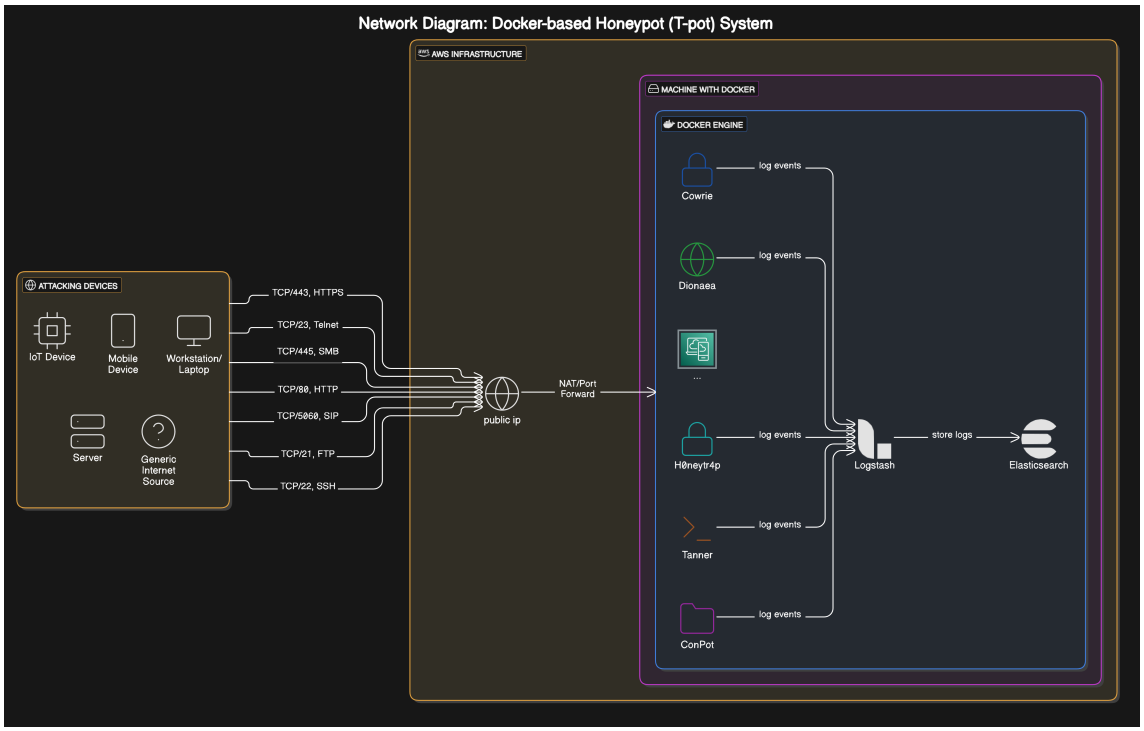
4. **Przebieg instalacji i konfiguracja:** Po starcie skryptu **install.sh** zobaczymy interaktywne menu tekstowe. Instalator sprawdzi środowisko (wolne zasoby, obecność pakietów) i poprosi o potwierdzenie kontynuacji. Następnie pojawi się pytanie o typ instalacji T-Pot – wybieramy zazwyczaj „hive" jako pełną instalację na pojedynczym węźle. Istnieją też inne tryby (np. sensor lub mini).

5. **Utworzenie kont administracyjnych:** Instalator następnie poprosi o podanie nazwy użytkownika i hasła dla interfejsu administracyjnego. Warto wybrać silne hasło, ponieważ interfejs ten będzie dostępny przez internet (na porcie 64297 HTTPS). Oczywiście można ustawić w odpowiedni sposób Security Group (w przypadku AWS), tak aby akceptowany był jedynie ruch z zaufanych adresów IP.

6. **Automatyczna instalacja usług (Docker):** Po zebraniu powyższych danych, instalator przystąpi do właściwego procesu instalacji. W tym kroku skrypt pobiera wszystkie wymagane obrazy Docker dla honeypotów i komponentów T-Pot.

7. **Restart i uruchomienie T-Pot:** Po instalacji proszeni jesteśmy o restart serwera/maszyny. Po ponownym uruchomieniu, wszystkie komponenty T-Pot startują jako usługa systemowa (za pomocą skryptów Docker Compose).

8. **Dostęp do interfejsu web (Kibana):** Kolejnym krokiem jest skorzystanie z interfejsu przeglądarkowego. Domyślnie T-Pot wystawia panel dostępny pod adresem: https://<adres_IP_serwera>:64297. Należy otworzyć przeglądarkę i połączyć się pod ten URL. Ukaże się ekran logowania – tutaj podajemy nazwę użytkownika i hasło, które ustawiliśmy w trakcie instalacji. Po zalogowaniu zobaczymy stronę powitalną T-Pot.



Rysunek 7: Schemat sieciowy implementacji T-Pot

6.5 Zbieranie logów, agregacja i wizualizacja (ELK, Kibana)

Jednym z największych atutów T-Pot jest jego rozbudowany system zbierania i prezentacji logów. W tradycyjnym honeypocie analityk często musi ręcznie przeglądać pliki logów tekstowych z pojedynczej usługi. W T-Pot natomiast wszystkie dane są automatycznie agregowane i udostępnione poprzez wygodne dashboardy.

Architektura logowania

Każdy honeypot (kontener) generuje logi swoich zdarzeń – np. Cowrie zapisuje próby logowań i polecenia, Dionaea loguje informacje o pobieranych plikach i połączeniach, Suricata generuje alerty IDS, itp. T-Pot uruchamia usługę Logstash (składnik stosu ELK), która zbiera te wszystkie strumienie logów z poszczególnych kontenerów. Logstash pełni funkcję agregatora i transformatora – parsuje logi na odpowiednie pola (np. rozbija wpis Cowrie na pola: IP źródłowy, użyty login, hasło, komenda; wpis Suricaty na: sygnatura ataku, adresy, porty, protokół, itd.) i zapisuje dane w centralnej bazie Elasticsearch. Elasticsearch to wydajna, skalowalna baza wyszukiwania, która przechowuje logi w formie dokumentów JSON, umożliwiając

szybkie filtrowanie i agregacje. Co ważne, T-Pot centralizuje logi w czasie rzeczywistym – czyli gdy tylko honeypot coś zarejestruje, niemal od razu pojawi się to w Elasticsearch, a przez to jest dostępne na dashboardach Kibany.

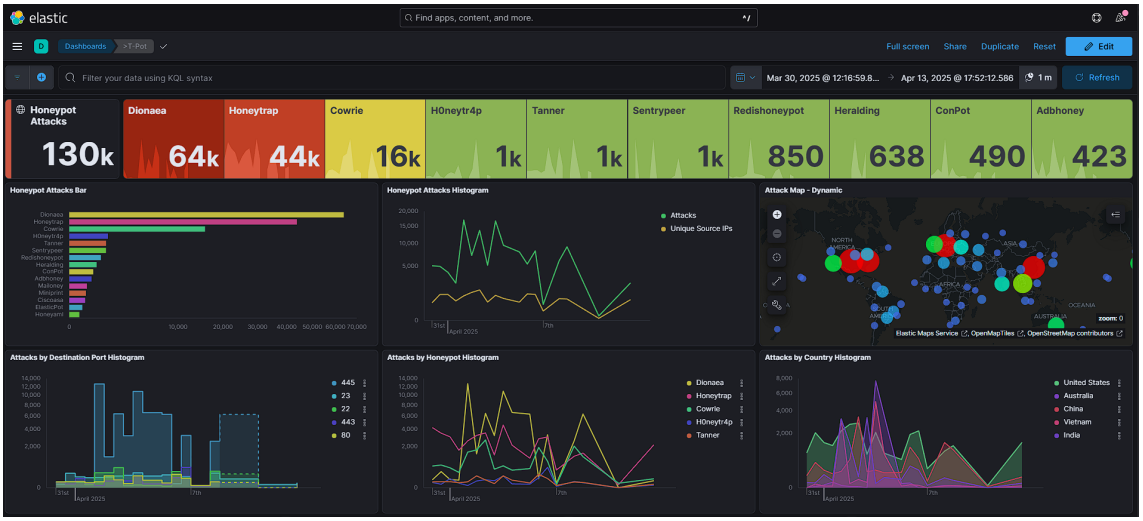
Elasticsearch + Kibana

Jak wspomniano, T-Pot dostarcza gotową instancję Kibana – webowej aplikacji do wizualizacji danych z Elasticsearch. Po zalogowaniu się do Kibany, użytkownik ma dostęp do wielu dashboardów i widżetów przygotowanych specjalnie pod honeypoty. Przykłady dostępnych dashboardów to:

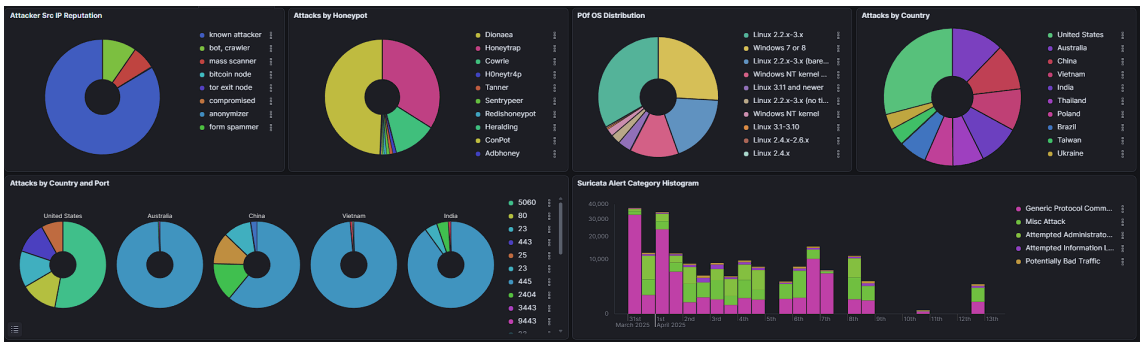
- Dashboard ogólny T-Pot
- Dashboards specyficzne dla honeypotów
- Mapa ataków (Attack Map)
- Narzędzia analityczne

Warto wspomnieć, że T-Pot domyślnie zachowuje logi lokalnie, ale nic nie stoi na przeszkodzie, by np. skonfigurować eksport wybranych danych do zewnętrznego SIEM-a czy bazy. Ponadto, dane z Elasticsearch można wykorzystać do automatycznego generowania raportów. Dla przykładu, można ustawić w Kibanie harmonogram generowania raportu PDF z podsumowaniem ataków co tydzień – co przydaje się w środowisku biznesowym do informowania kierownictwa o stanie bezpieczeństwa.

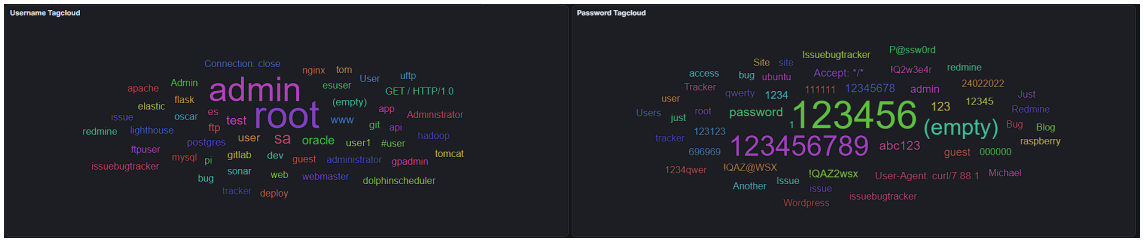
Poniżej przedstawione są wizualizacje naszych pomiarów w narzędziu Kibana:



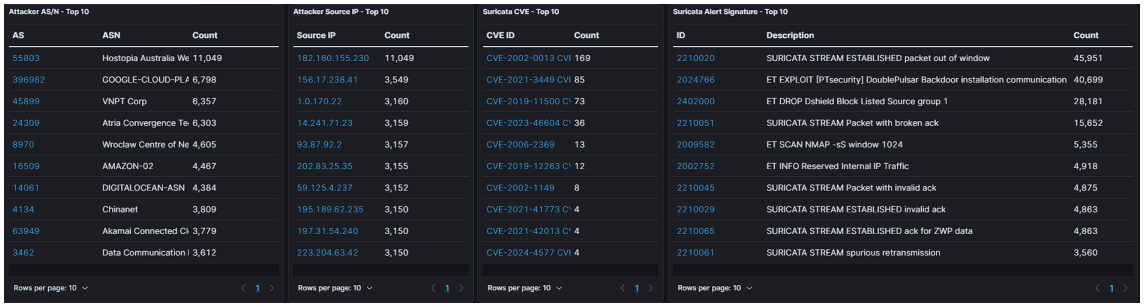
Rysunek 8: Podsumowanie ogólne (wraz z histogramem i mapą ataków)



Rysunek 9: Podsumowanie źródeł ataków oraz systemów operacyjnych



Rysunek 10: Najczęstsze wykorzystywane loginy i hasła do ataków



Rysunek 11: Statystyki atakujących, podatności i sygnatur ataków

6.6 Analiza i obserwacje wyników wdrożenia własnej instancji T-Pot

Jednym z najciekawszych aspektów korzystania z honeypota jest obserwacja realnych ataków, jakie do nas spływają. Dzięki różnorodności usług udostępnianych przez T-Pot, możemy zaobserwować szerokie spektrum technik używanych przez napastników. Poniżej przedstawiamy wyniki przeprowadzonych przez nas analiz naszej instancji t-pot'a na platformie AWS w przedziale czasowym 30 marca 2025 - 13 kwietnia 2025:

Najaktywniejsze honeypoty

Sumarycznie przeprowadzono 141 tysięcy ataków na maszynę. Poniższa tabela przedstawia udział poszczególnych honeypotów w liczbie odnotowanych ataków.

Honeypot	Liczba ataków
Dionaea	64k
Honeytrap	44k
Cowrie	26k
H0neytr4p	1k
Tanner	1k
SentryPeer	1k
Redishoneypot	853
Heralding	638
ConPot	490
ADBHoney	425

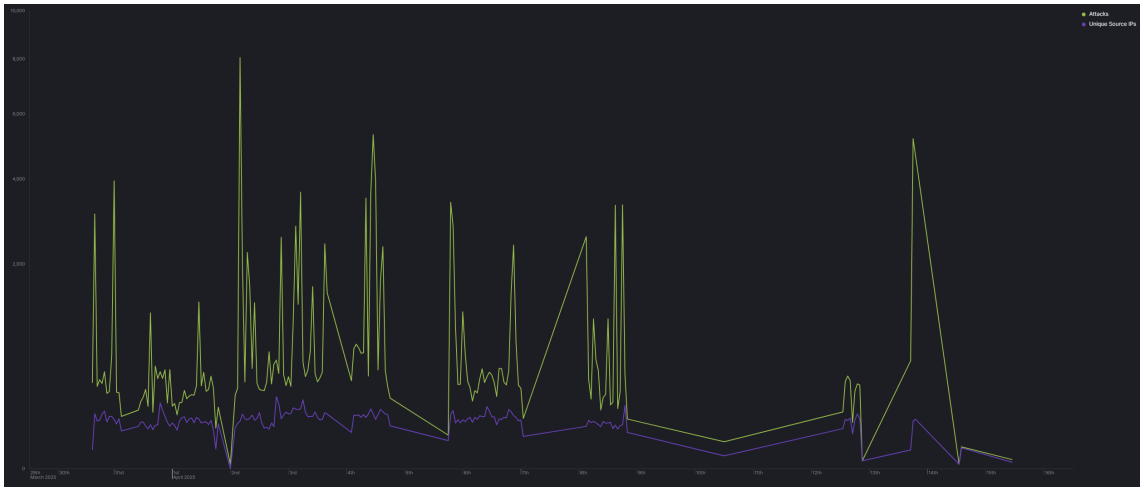
Tabela 1: Liczba ataków przeprowadzonych na konkretne honeypoty

- Dionanea - Niskointeraktywny honeypot do wykrywania malware i exploitów, emuluje wiele protokołów (FTP, HTTP, SMB, MSSQL itp.). Bardzo popularny do zbierania złośliwego oprogramowania i analizowania ataków sieciowych.
- Honeytrap - Niskointeraktywny honeypot, który emuluje różne usługi sieciowe, służy do wykrywania i logowania prób ataków na różne protokoły, głównie połączenia TCP. Dobry do szybkiego wykrywania aktywności atakujących.
- Cowrie - Średnio- i wysokointeraktywny honeypot SSH/Telnet, symuluje shell systemu Linux, pozwala na rejestrowanie sesji i pobranych plików. Używany do analizy ataków brute force i malware.
- H0neytr4p - Odmiana honeypota low-interaction, mniej popularny, ale skuteczny w wykrywaniu prostych ataków na usługi sieciowe.
- Tanner - Honeypot SSH/SCP służący do analizy botów próbujących przesyłać pliki i wykonać zdalny kod.
- SentryPeer - Honeypot o niskiej interakcji, detekcja nieautoryzowanego użycia VoIP/SIP (np. ataki na centrale telefoniczne).
- Redishoneypot - Honeypot specjalizujący się w emulacji serwera Redis, wykrywający próby wykorzystania luk w tym popularnym systemie bazodanowym.
- Heralding - Honeypot protokołów uwierzytelniających (RDP, VNC, FTP itp.).
- ConPot - Honeypot ICS/SCADA, symuluje urządzenia przemysłowe, wykorzystywany do analizy ataków na infrastrukturę krytyczną.
- ADBHoney - Honeypot emulujący usługi ADB (Android Debug Bridge), wykrywający próby ataków na urządzenia z Androidem.

Liczba ataków a pora dnia

Nie zaobserwowano wyraźnej korelacji pomiędzy porą dnia a liczbą wykrytych ataków. Współczesne ataki na usługi sieciowe są często realizowane przez zautomatyzowane skanery i botnety, które działają 24/7, niezależnie od strefy czasowej. Atakujące systemy rozproszone na całym świecie wysyłają próby ataku w różnych porach

dnia i nocy, co powoduje równomierne rozłożenie aktywności. Ponadto ataki mogą pochodzić z różnych regionów świata, co dodatkowo wyrównuje aktywność w czasie. Gdy w jednej strefie czasowej jest noc, w innej jest dzień i atakujący mogą działać aktywnie. Wyraźne są natomiast pojedyncze peaki. Wynikają one prawdopodobnie z targetowania honeypota przez zautomatyzowany skaner wysyłający dużą liczbę requestów.



Rysunek 12: Liczba ataków w kolejnych dniach

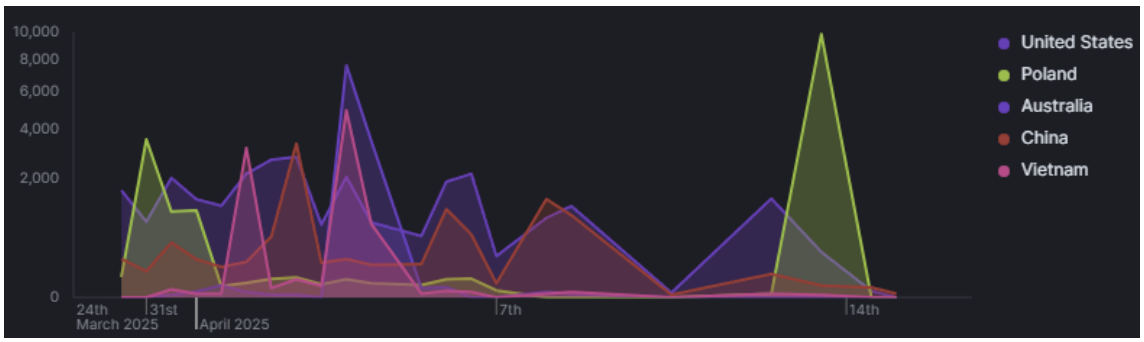
Źródła ataków

Ataki pochodzą praktycznie ze wszystkich kontynentów, co potwierdza skalę działania automatycznych botów.

Rysunek 13: Udział państw w atakach

W okresie badania najwięcej ataków pochodziło z:

- Stany Zjednoczone - około 27% ataków
- Polska - około 15% ataków
- Australia - około 11% ataków
- Chiny, Wietnam, Indie, Tajlandia, Brazylia - kolejne miejsca

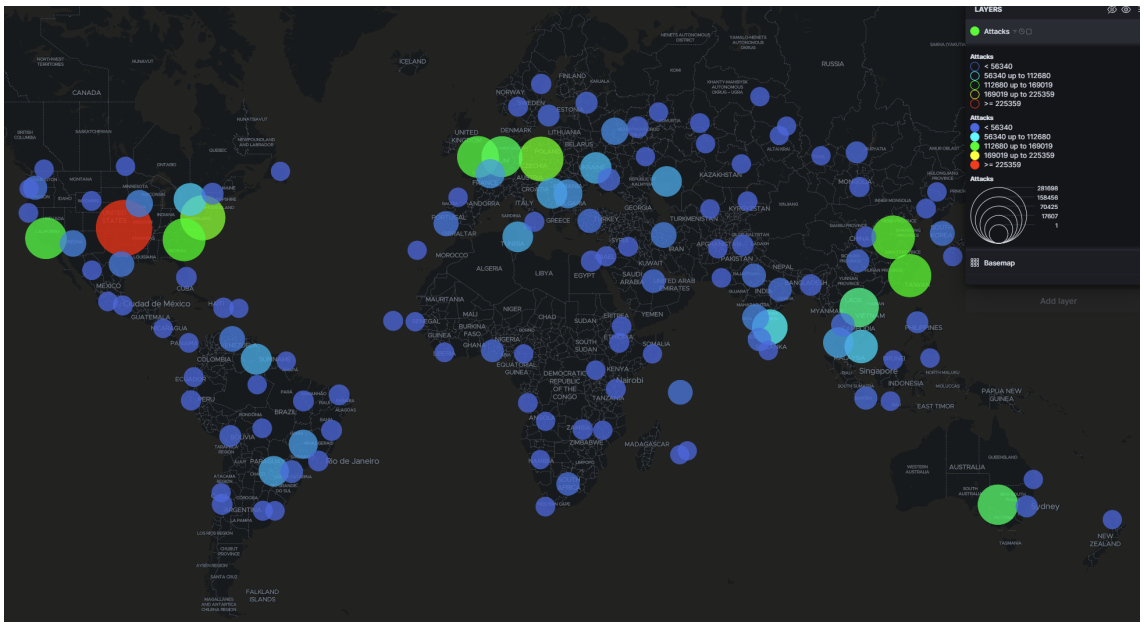


Rysunek 14: Ilość ataków z najaktywniejszych państw w kolejnych dniach

Wysoka liczba ataków pochodzących z rozwiniętych krajów wskazuje na wykorzystywanie słabo zabezpieczonych instancji w chmurze, które często stanowią łatwy cel dla cyberprzestępców. W takich państwach, gdzie infrastruktura chmurowa jest powszechna i rozbudowana, atakujący mogą łatwo znaleźć podatne serwery lub usługi, które następnie wykorzystują do prowadzenia działań ofensywnych.

Z kolei obecność krajów takich jak Wietnam, Indie czy Brazylia wśród źródeł ataków sugeruje udział zainfekowanych urządzeń konsumenckich, które są częścią rozproszonych botnetów. W tych regionach często notuje się mniejszą świadomość bezpieczeństwa oraz ograniczone zabezpieczenia indywidualnych urządzeń, co sprzyja masowym infekcjom i wykorzystywaniu ich do przeprowadzania ataków.

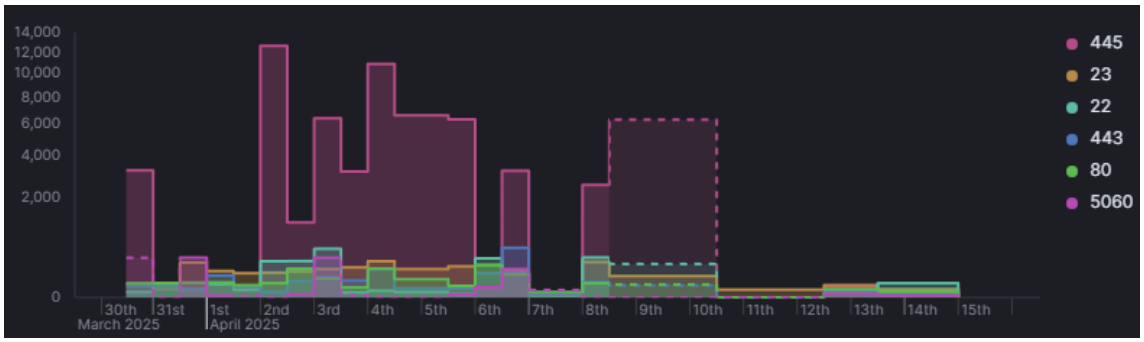
Wysoka pozycja Polski w zestawieniu wynika natomiast z faktu, że część zarejestrowanych prób ataków to działania testowe przeprowadzane przez nas samych w celach badawczych i diagnostycznych.



Rysunek 15: Mapa lokacji, z których były przeprowadzane ataki

Chociaż IP można fałszować (VPN, proxy), mapa nadal daje dobry pogląd na to 'skąd' przychodzą aktywności sieciowe.

Najczęściej atakowane porty



Rysunek 16: Najczęściej atakowane porty

Port	Protokół/Usługa	Kolor na wykresie	Opis i typowe zagrożenia
445	SMB	Fioletowy (ciemny)	Najczęściej atakowany port. Często wykorzystywany do rozprzestrzeniania ransomware (np. WannaCry, EternalBlue).
5060	SIP	Różowy (jasny)	Używany do ataków na systemy VoIP – próby rejestracji, floodów, brute-force kont SIP.
23	Telnet	Żółty	Niezabezpieczony protokół komunikacyjny, popularny cel ataków brute-force (zwłaszcza w IoT).
22	SSH	Turkusowy	Cel ataków słownikowych na logowanie do serwerów. Bardzo często celem Cowrie.
443	HTTPS	Niebieski	Mniej liczny ruch – możliwe skanowanie serwisów webowych z błędami (np. Log4Shell, CVE).
80	HTTP	Zielony	Ataki typu directory traversal, XSS, RCE, skanowanie znanych podatności webowych.

Tabela 2: Najczęściej atakowane porty

Po przeprowadzeniu szczegółowej analizy jesteśmy w stanie przedstawić rekomendacje zabezpieczenia najczęściej atakowanych portów:

- Port 445 (SMB), będący głównym wektorem ransomware, powinien być zawsze blokowany na styku sieci z Internetem. SMB powinno być dostępne wyłącznie wewnątrz sieci lokalnej lub przez bezpieczne tunele VPN. Jeśli wymagany jest dostęp zdalny do zasobów SMB, rekomenduje się stosowanie zabezpieczonych rozwiązań takich jak VPN z silnym uwierzytelnianiem wieloskładnikowym (MFA).

- Port 5060 (SIP), ze względu na charakterystykę wymaga zaawansowanej ochrony. Zaleca się stosowanie firewalli aplikacyjnych dedykowanych dla VoIP, które analizują i filtrują ruch SIP na poziomie sesji, wykrywając anomalie i blokując nieautoryzowane próby rejestracji. Należy także ograniczyć dostęp do serwerów SIP tylko do zaufanych adresów IP oraz stosować mechanizmy rate-limitowania prób logowania.

- Port 23 (Telnet) jest uznawany za bardzo niebezpieczny, gdyż protokół Telnet przesyła dane w postaci jawnego tekstu. Najlepszą praktyką jest całkowite wyłączenie Telnetu i zastąpienie go bezpiecznym protokołem SSH. Jeżeli Telnet musi być używany (np. w starszych urządzeniach IoT), zaleca się ograniczenie

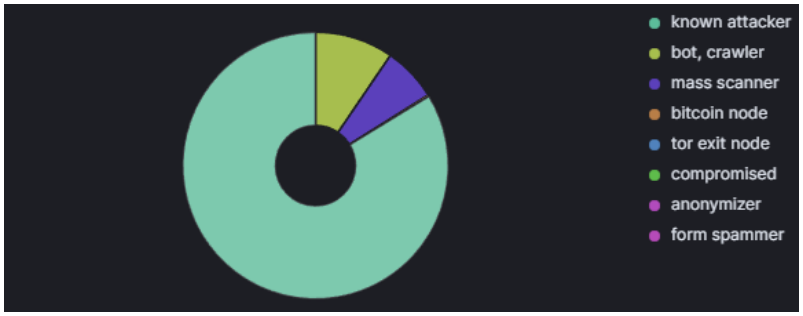
dostępu do niego wyłącznie w ramach wydzielonych podsieci lub przez VPN oraz stosowanie silnych, unikalnych haseł.

- W przypadku portu 22 (SSH), rekomenduje się kilka warstw ochrony. Po pierwsze, zmiana domyślnego portu SSH na nietypowy (tzw. "security through obscurity") może ograniczyć liczbę automatycznych skanów. Dobrą praktyką jest także wdrożenie uwierzytelniania kluczem publicznym zamiast hasła oraz włączenie ograniczeń dostępu na podstawie adresów IP. Warto także wdrożyć mechanizmy blokowania po wielokrotnych nieudanych próbach logowania (np. Fail2Ban) oraz regularnie aktualizować oprogramowanie SSH.

- Dla portów 443 (HTTPS) i 80 (HTTP), które obsługują usługi webowe, najczęstszym zagrożeniem są exploity podatności aplikacyjnych (np. XSS, związane z ciasteczkami, CORS itp). Rekomendujemy zastosowanie Web Application Firewall (WAF) analizującego ruch HTTP/HTTPS w czasie rzeczywistym, zabezpieczenie aplikacji za pomocą testów penetracyjnych oraz aktualizowanie wszystkich komponentów webowych (serwery, frameworki, biblioteki) na bieżąco.

Reputacja atakujących

T-pot pozwala także na weryfikację reputacji adresów IP atakujących. Dzięki integracji z bazami danych reputacyjnych oraz mechanizmom klasyfikacji, T-Pot umożliwia identyfikację charakteru i potencjalnego zagrożenia ze strony źródeł ataków.



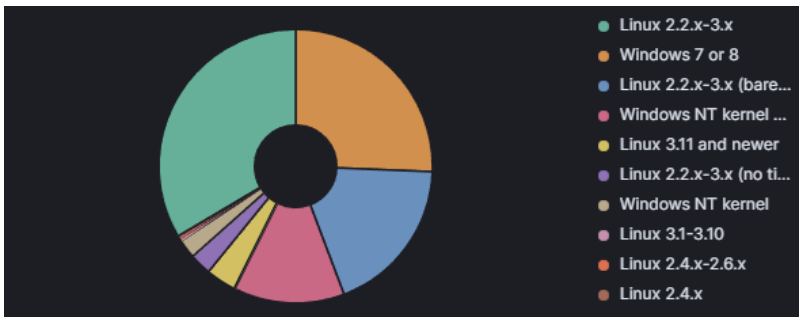
Rysunek 17: Reputacja atakujących

Kategoria	Liczba zdarzeń	Opis
known attacker	29 641	Adresy IP zidentyfikowane jako znani agresorzy, często powiązani z wcześniejszymi atakami lub działaniami złośliwymi. Stanowią największą grupę, co wskazuje na powtarzalność i uporczywość atakujących.
bot, crawler	3 377	Automatyczne boty i crawlery, które skanują sieć w celu zbierania informacji lub wyszukiwania podatności. Choć nie zawsze złośliwe, ich aktywność może być wstępem do późniejszych ataków.
mass scanner	2 327	Systemy przeprowadzające masowe skanowanie adresów IP i portów w celu wykrycia otwartych usług i potencjalnych celów ataku. Są to często narzędzia wykorzystywane przez cyberprzestępców do rozpoznania infrastruktury.
bitcoin node	39	Adresy IP należące do węzłów sieci Bitcoin. Ich obecność może wskazywać na wykorzystanie infrastruktury kryptowalutowej lub po prostu na przypadkowe próby połączeń.
tor exit node	20	Węzły wyjściowe sieci Tor, które umożliwiają anonimowy dostęp do internetu. Ataki z takich adresów są trudniejsze do powiązania z konkretnymi osobami, co utrudnia śledzenie sprawców.
compromised	6	Adresy IP zainfekowanych urządzeń, które zostały przejęte przez atakujących i wykorzystywane jako proxy lub do dalszych ataków. Stanowią realne zagrożenie, gdyż działają jako „żółte lampki" w łańcuchu ataku.
anonymizer	2	Usługi maskujące prawdziwy adres IP użytkownika, takie jak VPN-y lub proxy. Utrudniają identyfikację źródła ataku.
form spammer	2	Adresy IP wykorzystywane do automatycznego wysyłania spamu przez formularze internetowe. Choć mniej groźne niż ataki bezpośrednie, mogą zakłócać działanie serwisów i obciążać zasoby.

Tabela 3: Reputacja atakujących

Dzięki funkcjom T-Pot umożliwiającym weryfikację reputacji atakujących, możliwe jest nie tylko wykrycie samego ataku, ale także jego kontekst i potencjalne zagrożenie. Największą grupę stanowią znani atakujący, co potwierdza istnienie aktywnych i powtarzających się zagrożeń. Obecność botów, masowych skanerów oraz anonimowych źródeł (Tor, anonymizery) wskazuje na złożoność i różnorodność technik stosowanych przez cyberprzestępców.

Atakujące systemy



Rysunek 18: Rodzaje atakujących systemów

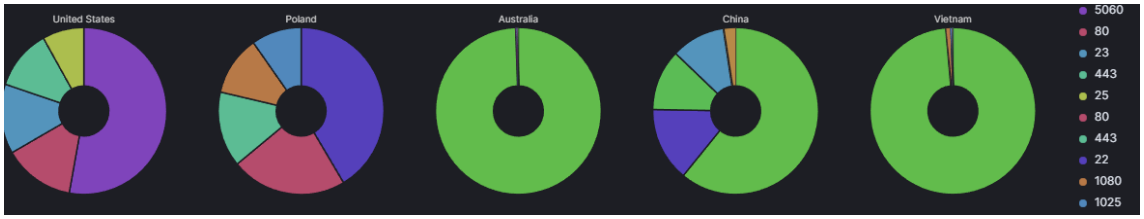
Poniższa tabela przedstawia zestawienie najczęściej wykrywanych systemów operacyjnych (OS) na podstawie analizowanych prób ataków w honeypotach

System operacyjny	Liczba wykryć	Opis
Linux 2.2.x-3.x	82 441	Starsze i średnio nowoczesne wersje jądra Linux, często wykorzystywane przez serwery i urządzenia sieciowe. Popularne w urządzeniach IoT i systemach embedded. Może to wskazywać na kompromitację urządzeń z nieaktualnym oprogramowaniem bądź spoofing sygnatur OS.
Windows 7 or 8	63 071	Systemy operacyjne Microsoft Windows, szeroko rozpowszechnione na komputerach osobistych i firmowych. Często celem ataków ze względu na dużą bazę użytkowników.
Linux 2.2.x-3.x (barebone)	45 788	Minimalistyczne wersje jądra Linux, pozbawione dodatkowych funkcji i timestampów, wykorzystywane w lekkich systemach i urządzeniach wbudowanych.
Windows NT kernel 5.x	32 118	Wersje jądra Windows NT używane w starszych systemach Windows (np. Windows 2000, XP). Infrastruktura legacy wciąż działa w sieci (np. bankomaty, systemy przemysłowe, starsze środowiska korporacyjne). Botnety aktywnie wykorzystują luki w nieaktualnych systemach Windows.
Linux 3.11 and newer	8 806	Nowsze wersje jądra Linux, stosowane w nowoczesnych dystrybucjach i serwerach. Świadczy o obecności ataków na aktualne systemy.
Linux 2.2.x-3.x (no timestamps)	6 328	Wersje jądra Linux, w których wyłączono zapisywanie znaczników czasu, co może utrudniać analizę zdarzeń i ataków.
Windows NT kernel	5 494	Ogólna kategoria starszych wersji jądra Windows NT, wykorzystywanych w różnych systemach legacy.
Linux 3.1-3.10	797	Wersje jądra Linux z przedziału 3.1 do 3.10, stosowane w średnio nowoczesnych systemach.
Linux 2.4.x-2.6.x	666	Bardzo stare wersje jądra Linux, które mogą występować w przestarzałych urządzeniach lub systemach.
Linux 2.4.x	486	Jedne z najstarszych wersji jądra Linux, rzadko spotykane, ale wciąż obecne w niektórych legacy systemach.

Tabela 4: Systemy operacyjne atakujących

Ze względu na fakt, iż znaczna część ataków została przeprowadzona prawdopodobnie z przejętych wcześniej maszyn bądź serwerów, dominacja starszych systemów w atakach na honeypot odzwierciedla globalny problem utrzymywania nieaktualnej infrastruktury. Podczas gdy nowoczesne systemy są względnie bezpieczne, luki w legacy systemach oraz IoT stanowią krytyczne wektory ataków. Organizacje muszą priorytetowo traktować aktualizacje, segmentację sieci oraz zaawansowane mechanizmy detekcji, aby neutralizować te zagrożenia.

Lokalizacja atakującego a atakowane porty (usługi)



Rysunek 19: Lokalizacja atakującego a atakowane porty (usługi)

- Stany Zjednoczone
  - Dominujące porty: 5060 (SIP) – VoIP – największy udział, 443, 25, 23, 80 – klasyczne wektory ataków.
  - Szerokie spektrum atakowanych usług, w tym VoIP (SIP), co może wskazywać na wykorzystanie zasobów chmurowych z USA do automatycznych ataków na systemy VoIP oraz klasyczne serwisy jak SSH, HTTP i Telnet. Obecność SMTP może oznaczać próbę rozsyłania spamu lub ataków na serwery mailowe.
- Polska

  - Dominujące porty: 22 (SSH) – największy udział, 443, 1080 (SOCKS Proxy), 1025 (Dynamicznie alokowany przez Windowsa, wykorzystywany przez wiele trojanów), 80.
  - Wysoka aktywność z Polski również ukierunkowana SSH (atakowanie za pomocą narzędzia Hydra), ale równocześnie klasyczne porty webowe i IoT.

- Australia
  - Dominujące porty: 445 (SMB), 2404, 3443, 9443 - niestandardowe HTTPS i serwisy webowe
  - Australia prowadzi bardzo intensywne ataki na port SMB (445), który był wektorem np. dla EternalBlue (WannaCry). Pozostałe porty mogą wskazywać na próby obejścia klasycznych reguł firewalli.

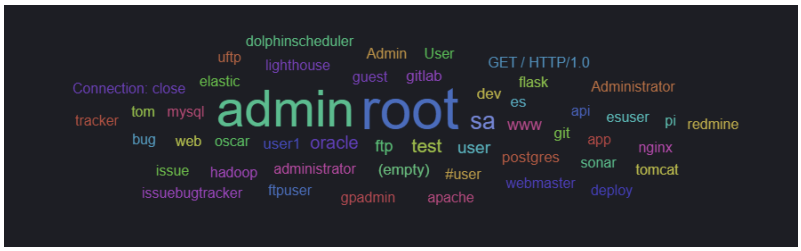
- Chiny
  - Dominujące porty: 445 (SMB), 22 (SSH), 6379 (Redis DB – często niezabezpieczony), 23 (Telnet), 1433 (MS SQL Server)
  - Chiny prowadzą szeroko zakrojone ataki skanujące i eksploatujące

- Wietnam
  - Dominujące porty: 445 (SMB), 1080 (SOCKS Proxy), 9100, 8081, 7170 – drukarki, webapki, IoT
  - Ruch z Wietnamu silnie koncentruje się na SMB, ale także sugeruje skanowanie nietypowych portów wykorzystywanych przez IoT oraz proxy.

Loginy i hasła

W ramach obserwacji działań wymierzonych w honeypot, wykryto również liczne próby ataków typu brute force, których celem było uzyskanie nieautoryzowanego dostępu do usług – głównie SSH i HTTP Basic Auth. Analiza logów wskazuje jednoznacznie, że atakujący posługiwali się zestawami powszechnie znanych i łatwych do odgadnięcia loginów oraz haseł, znajdujących się na popularnych listach typu Top 100 passwords oraz Top usernames (np. rockyou.txt, SecLists, Hashes.org).

Poniższe dane odzwierciedlają Top 50 prób uwierzytelnienia z wykorzystaniem najczęściej wpisywanych nazw użytkowników:



Rysunek 20: Najpopularniejsze loginy

Te nazwy użytkowników należą do najczęściej używanych domyślnych kont w:

- systemach Linux/Unix (root, user, pi, postgres)
- aplikacjach webowych (admin, test, web, gitlab)
- bazach danych (mysql, oracle, sa)
- środowiskach CI/CD i devops (git, deploy, dev)

Zidentyfikowane próby obejmują również bardzo słabe i generyczne hasła:

Rysunek 21: Najpopularniejsze hasła

Część haseł (np. 123456, admin, password) należy do pierwszej dziesiątki najgorszych haseł w zestawieniach takich jak NordPass Top 200 czy HaveIBeenPwned common passwords list.

Atakujące AS

Poniżej przedstawiono analizę najaktywniejszych dostawców usług sieciowych (AS – Autonomous Systems), z których inicjowane były ataki na honeypot.

AS	ASN	Liczba ataków
Wroclaw Centre of Networking and Supercomputing	8970	14,453
Hostopia Australia Web Pty Ltd	55803	11,049
GOOGLE-CLOUD-PLATFORM	396982	6,906
VNPT Corp (Wietnam)	45899	6,357
Atria Convergence Technologies (Indie)	24309	6,303
AMAZON-02 (AWS)	16509	4,603
DIGITALOCEAN-ASN	14061	4,405
Chinanet	4134	3,824
Akamai Connected Cloud	63949	3,802
Data Communication Business Group (Tajwan)	3462	3,612

Tabela 5: Atakujące AS

Największym źródłem okazało się Wrocławskie Centrum Sieciowo - Superkomputerowe (WCSS), ze względu na to, że studiują tam najwybitniejsi oraz najbardziej aktywni hackerzy na całym świecie. W innym przypadku mogłoby to oznaczać np. kompromitację systemów w infrastrukturze uczelnianej (np. zainfekowane VM-y, kontenery lub serwery).

Infrastruktura chmurowa była często stosowana jako główne źródło ataków - Google Cloud (396982), AWS (16509) oraz DigitalOcean (14061) pojawiają się w top 10. Często wykorzystywana jest ona do ataków automatycznych i rozproszonych (DDoS, bruteforce), hostowania botów lub serwisów C2 (command and control) czy testów przez red teamy i pentesterów (w tym bez wiedzy administracji).

Można było również zauważyć ISP z krajów o wysokiej aktywności botnetowej. VNPT (Wietnam), Atria (Indie), Chinanet (Chiny), Data Comm. (Tajwan) to operatorzy, z których adresów IP pochodziły tysiące ataków. Może to świadczyć o braku odpowiedniego nadzoru nad bezpieczeństwem wśród użytkowników indywidualnych czy powszechnym wykorzystaniu ich klientów jako część zainfekowanych botnetów.

Zaskakująco dużą liczbę ataków odnotowano z Hostopia Australia Web. Jako firma świadcząca usługi hostingowe, może być źródłem skryptów skanujących uruchamianych przez klientów bądź kompromitowanych instancji webowych lub serwerów usług.

Rodzaje wykorzystywanych podatności (CVE)

Analiza danych z systemu wykrywania zagrożeń Suricata ujawnia istotne trendy w zakresie najczęściej wykrywanych podatności (CVE), co pozwala lepiej zrozumieć taktyki i cele atakujących. Szczególną uwagę zwraca różnorodność wykrywanych podatności - od bardzo starych po całkiem nowe - co wskazuje na szerokie spektrum stosowanych narzędzi ataku.

CVE	Liczba detekcji	Opis skrócony
CVE-2002-0013, CVE-2002-0012	169	Apache chunked encoding overflow – bardzo stara, ale nadal skanowana podatność w serwerach Apache.
CVE-2021-3449	85	OpenSSL denial-of-service – błąd obsługi renegocjacji TLS, powodujący crash serwera (DoS).
CVE-2019-11500	73	Exim RCE – podatność umożliwiająca wykonanie zdalnego kodu na serwerach pocztowych Exim.
CVE-2023-46604	36	Apache ActiveMQ RCE – podatność wykorzystywana w kampaniach malware, umożliwiająca RCE przez manipulację serializacją klas Javy.
CVE-2006-2369	13	PHP-CGI Remote Code Execution – historyczna podatność PHP w kontekście CGI.
CVE-2019-12263, -12261, -12260, -12255	12	Zestaw podatności Exim – różne błędy w protokole SMTP i jego obsłudze.
CVE-2002-1149	8	OpenSSL buffer overflow – kolejna podatność z początków internetu, nadal skanowana.
CVE-2021-41773	4	Apache Path Traversal – zdalny dostęp do plików spoza katalogu głównego na Apache 2.4.49.
CVE-2021-42013	4	Remote Code Execution w Apache 2.4.50 – eskalacja poprzedniej podatności.
CVE-2024-4577, CVE-2002-0953	4	Nowa podatność w PHP i stara w mod_ssl.

Tabela 6: Rodzaje wykorzystywanych podatności (CVE)

Jednym z najbardziej uderzających wniosków jest obecność bardzo starych podatności z lat 2002–2006, takich jak CVE-2002-0013, CVE-2002-0012 czy CVE-2006-2369. Atakujący nadal aktywnie skanują pod kątem tych błędów, co może świadczyć o używaniu generycznych zestawów exploitów, które nie były aktualizowane od wielu lat. Może to również wskazywać na działalność prostych botnetów IoT lub skryptów malware, które wciąż próbują wykrywać zapomniane, niezałatane systemy legacy. W praktyce oznacza to, że nawet bardzo stare podatności mogą nadal stanowić zagrożenie, jeżeli są wykorzystywane w infrastrukturze.

Z drugiej strony, widoczna jest również obecność nowoczesnych wektorów zdalnego wykonania kodu (RCE), szczególnie z ostatnich lat (2021–2024). Przykłady takie jak CVE-2023-46604 (Apache ActiveMQ RCE) oraz CVE-2024-4577 (najnowsza podatność PHP) są typowymi celami kampanii malware, w tym ransomware, koparek kryptowalut oraz działań tzw. initial access brokerów. Często są one wykorzystywane do uzyskania pierwszego dostępu do infrastruktury, który następnie może zostać odsprzedany innym grupom. Ponadto, CVE-2021-41773 i CVE-2021-42013 świadczą o ciągłych próbach eksploatacji popularnego serwera Apache HTTP, często występującego w środowiskach produkcyjnych.

Kolejnym istotnym aspektem jest popularność serwera pocztowego Exim jako celu ataków. Kilka z analizowanych CVE dotyczy właśnie Exima (np. CVE-2019-11500 oraz grupa CVE-2019-122xx), co wskazuje, że atakujący nadal chętnie wykorzystują go jako wektor infekcji. Exim bywa domyślnym komponentem w wielu dystrybucjach Linuksa, jak Debian, co czyni go atrakcyjnym celem zarówno dla botnetów, jak i grup APT prowadzących kampanie spamowe czy wykorzystywanych jako malware droppery.

Rodzaje ataków i techniki

Typowa interakcja atakującego to: próba połączenia z Telnet/SSH → logowanie słownikowe → próba pobrania skryptu przez wget lub curl. W logach Suricata zaobserwowano też wiele eksploitów i prób komunikacji z malware, m.in.:

- ET EXPLOIT DoublePulsar Backdoor – 40 699 razy
- SURICATA STREAM ESTABLISHED packet out of window – 45 951 razy
- ET SCAN NMAP -sS – klasyczne rozpoznanie sieci

Zarejestrowano również alerty dotyczące komunikacji C2 i pakietów „złamanych" TCP – co wskazuje na ruch testowy, skanerski lub aktywność malware.

Ataki typu exploit (wykorzystanie podatności)

T-Pot wystawia kilka usług podatnych „z definicji" lub symulujących podatności, np. stary serwer SMB, API Elasticsearch, usługi SCADA, bazy danych itp. Przykładowo ElasticPot nasłuchujący na porcie 9200 udaje niezabezpieczony serwer Elasticsearch – w logach tego modułu można zobaczyć próby wywołania znanych endpointów API, które służą do exploitów (np. _search?size=... z wstrzykniętym kodem). Inny przykład to honeypot Conpot (ICS/SCADA) – może on zostać zaatakowany przez skrypty szukające urządzeń przemysłowych; wtedy loguje komendy protokołów jak Modbus czy S7comm.

Ataki malware/botnet – pobieranie i instalacja złośliwego oprogramowania

Bardzo wartościową cechą honeypotów jest zdolność do przechwytywania plików malware dostarczanych przez atakujących. W T-Pot głównym „łowcą" malware jest honeypot Dionaea. Jego zadaniem jest udawać różne usługi (SMB, FTP, HTTP,

SQL itp.) i akceptować pliki od atakującego, zapisując je na dysku. W ten sposób można zebrać próbki ransomware, trojanów bankowych czy komponentów botnetu, który próbował nas zaatakować. T-Pot dzięki Suricacie rozpoznaje znane rodziny malware po sygnaturach w ruchu.

Inne obserwacje

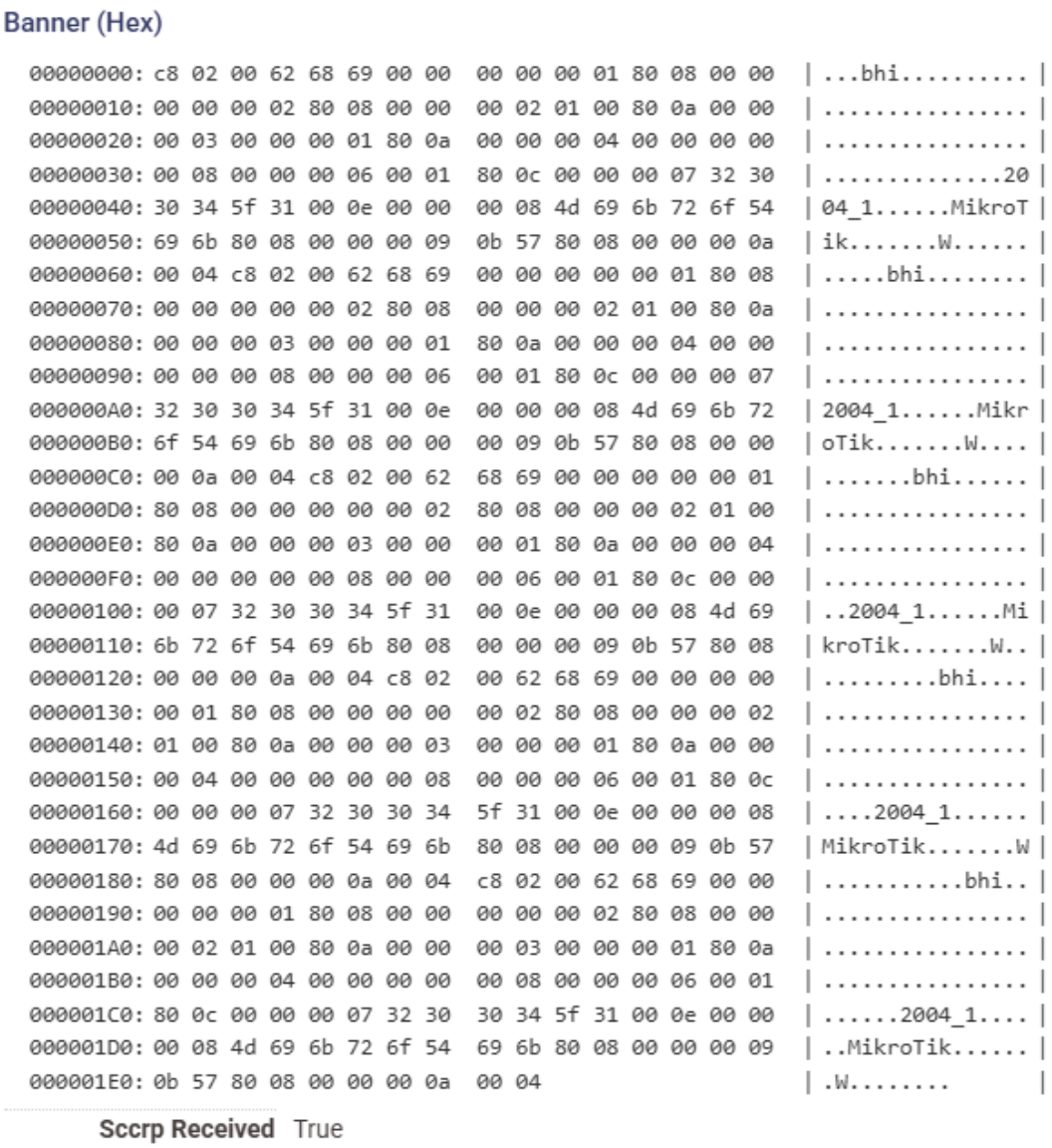
Poza powyższymi, T-Pot rejestruje również skanowanie i rekonesans (np. moduł p0f zanotuje skan portów i spróbuje określić system operacyjny skanującego po TTL pakietów; Suricata wygeneruje alerty o skanowaniu), próby exfiltracji danych (na honeypotach typu database/web można zobaczyć nietypowe zapytania wskazujące na chęć wyciągnięcia informacji, choć w honeypocie nie ma prawdziwych danych), czy ataki typu DoS wymierzone w rzekome usługi (np. ktoś może spróbować zalać fałszywy serwer HTTP ruchem – co odnotuje Suricata). T-Pot z racji bycia „fabryką danych" pozwala też na statystyczne spojrzenie: możemy np. dowiedzieć się jakie protokoły są najczęściej atakowane w ciągu miesiąca, z jakich krajów pochodzą najaktywniejsi intruzi, jakie hasła są najpopularniejsze w brute-force itp. Analiza takich danych to kopalnia wiedzy dla zespołów threat intelligence.

6.7 Przykładowa analiza dwóch atakujących adresów IP

Analiza została przeprowadzona na podstawie danych zebranych z serwisów Censys oraz Shodan.

Adres 38.156.75.17

- OS: Ubuntu Linux
- Services (13): 22/SSH, 53/DNS, 443/HTTP, 1022/SSH, 1701/L2TP, 2000/MIKROTIK_BW, 2206/SSH, 2225/SSH, 5222/XMPP, 8006/HTTP, 13333/SSH, 30080/HTTP, 50443/HTTP
- SSH 22/TCP - Ubuntu Linux, OpenBSD OpenSSH 8.9p1
- HTTP 443/TCP - Status 400 Bad Request
- SSH 1022/TCP - Ubuntu Linux 20.04, OpenBSD OpenSSH 8.2
- L2TP 1701/UDP



Rysunek 22: Banner Mikrotik na porcie 1701

- 2000/TCP - MikroTik bandwidth-test server
- SSH 2206/TCP
- SSH 2225/TCP - Ubuntu Linux, OpenBSD OpenSSH 8.9p1
- 3478/UDP - STUN, Server IP: 38.156.75.17
- XMPP 5222/TCP, Banner - «?xml version=’1.0’?><stream:stream xmlns:stream=’http://etherx.jabber.org/streams’ xmlns=’jabber:client’ xml:lang=’en’ id=”><stream:error><improper-addressing xmlns=’urn:ietf:params:xml:ns:xmpp-stream"
- HTTP 8006/TCP - Proxmox, Proxmox Virtual Environment
- SSH 13333/TCP - Ubuntu Linux, OpenBSD OpenSSH 8.9p1

- HTTP 30080/TCP - nginx 1.18.0, Status 404 Not Found
- HTTP 50443/TCP - nginx 1.18.0, Status 200 OK, HTML Title: Welcome to nginx!

Serwer wygląda na komponent infrastruktury C2 (Command and Control) lub węzeł pośredniczący botnetu, służący do kierowania atakami, tunelowania oraz komunikacji.

Liczne instancje SSH na różnych portach mogą wskazywać na:

- środowisko testowe lub sterujące
- próbę ukrycia dostępu przed skanerami
- możliwość zastosowania tunelowania lub backdoora

Obecność wielu interfejsów HTTP/HTTPS sugeruje:

- obecność dashboardu zarządzającego (Proxmox)
- możliwe interfejsy C2 lub GUI dla operatora
- serwer pełni rolę infrastruktury zarządzającej maszynami wirtualnymi lub kontenerami (np. do uruchamiania złośliwego oprogramowania)

MikroTik Bandwidth Test (2000/TCP)

- Banner: MikroTik bandwidth-test server.
- Znaczenie: Typowe dla urządzeń sieciowych MikroTik.
- Wnioski: Host może emulować MikroTik w celu ukrycia rzeczywistej tożsamości lub być skompromitowanym routerem MikroTik.

L2TP (1701/UDP)

- Wskazuje na tunel VPN – może służyć do tunelowania komunikacji z innymi węzłami lub umożliwienia zdalnego dostępu do sieci botnetu.
- Obecność może sugerować próbę stworzenia stealth VPN C2.

XMPP (5222/TCP)

- Banner XML pokazuje typową sesję startową XMPP.
- stream:error – ale z odpowiedzią – znaczy, że serwer działa.
- XMPP używane jest często jako protokół C2 (szyfrowana, trudna do detekcji komunikacja, często wykorzystywane przez botnety APT (np. PlugX, Ares)).

STUN (3478/UDP)

- Umożliwia penetrację NAT – niezbędna dla wielu C2/botów.
- Pozwala określić zewnętrzny adres IP i port.
- Popularne w APT i malware P2P jako komponent wykrywania sieci.

Podsumowując, prawdopodobnie jest to infrastruktura klasy Red Team / APT / Botnet-as-a-Service.

Adres 200.7.124.35

Services (6): 21/FTP, 23/TELNET, 53/DNS, 80/HTTP, 443/HTTP, 1900/UPnP

- Services (6): 21/FTP, 23/TELNET, 53/DNS, 80/HTTP, 443/HTTP, 1900/UPnP
- FTP 21/TCP – Banner: `220 Ftp firmware update utility`, brak obsługi AUTH TLS/SSL (komendy zwracają błędy 500) – wskazuje na brak szyfrowania.
- TELNET 23/TCP – klasyczny interfejs logowania `Login:`, sugeruje starsze urządzenie sieciowe lub brak odpowiedniego hardeningu.
- DNS 53/UDP – serwer typu forwarding, odpowiedzi z kodem SUCCESS .
- HTTP 80/TCP – Strona zwraca kod `200 OK`, identyczna treść jak HTTPS (identyczny hash SHA1).
- HTTPS 443/TCP – Certyfikat samopodpisany, dane wskazujące na firmę Broadcom:
  - `CN=Daniel, O=Broadcom, email=kiding@broadcom.com`
  - TLSv1.3 z szyfrowaniem `CHACHA20_POLY1305_SHA256`
  - JA3S fingerprint: `475c9302dc42b2751db9edcac3b74891`
- UPnP 1900/UDP – odpowiedź HTTP 200 z lokalizacją do XML konfiguracji urządzenia WPS: `http://192.168.1.1:49152/wps_device.xml`, nagłówek `SERVER: Unspecified, UPnP/1.0`, UUID wskazuje na unikalne urządzenie (możliwy router, IoT)

Serwer prawdopodobnie reprezentuje urządzenie sieciowe (router, modem lub bramka IoT), najpewniej dostępne publicznie i nieodpowiednio zabezpieczone.

Główne przesłanki mogące świadczyć o nieautoryzowanym udostępnieniu urządzenia lub słabej konfiguracji bezpieczeństwa:

- FTP i TELNET dostępne z zewnątrz – wskazują na nieaktualne praktyki bezpieczeństwa; mogą umożliwiać nieautoryzowany dostęp.
- Brak szyfrowania na FTP – podatność na przechwycenie danych logowania i manipulację aktualizacjami.
- Samopodpisany certyfikat HTTPS – brak walidacji tożsamości serwera, potencjalne ryzyko MITM.
- UPnP aktywne publicznie – bardzo ryzykowne, umożliwia zdalną konfigurację NAT/firewalla, często wykorzystywane przez malware (np. Mirai) do lateral movement lub eksploitacji.

7 Podsumowanie

Honeypoty stanowią unikalną technikę obrony i analizy w cyberbezpieczeństwie – działając jako kontrolowane pułapki, pozwalają na wykrycie włamań oraz zgromadzenie bezcennych danych o metodach działania przestępców. Centralnym punktem sprawozdania było przedstawienie platformy T-Pot, która jest kompleksowym rozwiązaniem integrującym kilkadziesiąt honeypotów różnych typów w jedną platformę badawczą. T-Pot łączy w sobie zalety wielu pojedynczych pułapek i dostarcza zaawansowane narzędzia analityczne (ELK, Kibana, mapy ataków), co czyni go potężnym narzędziem zarówno dla praktyków (SOC, blue team) jak i badaczy akademickich. Przedstawione zostały kluczowe funkcjonalności T-Pot:

- obsługa wielu protokołów
- architektura oparta na Dockerze
- integracja z IDS – oraz wymagania systemowe

Omówione zostały także zalety (kompleksowość, darmowa dostępność, elastyczność) i wady (duży apetyt na zasoby, złożoność). Opisano krok po kroku proces wdrażania T-Pot w środowisku linuksowym z użyciem Dockera. Z ilustrowanych przykładów ataków wynika, że honeypoty T-Pot rejestrują szerokie spektrum zagrożeń – od masowych ataków brute-force (np. botnety Mirai) po zaawansowane exploity i próby dostarczenia malware – dostarczając bogatego materiału do analiz.

Podkreślilono rolę systemu logowania i wizualizacji w T-Pot: scentralizowane zbieranie logów z wszystkich kontenerów i prezentacja poprzez gotowe panele Kibana znacznie ułatwia pracę z danymi co czyni platformę praktycznym narzędziem do threat intelligence.

Na koniec przedstawiono szczegółową analizę pozyskanych danych, między innymi: źródła ataków, lokalizacje atakujących, najczęściej wykorzystywane dane logowania, rodzaje i techniki ataków czy najczęściej atakowane usługi. Podjęto się także analizy dwóch przykładowych adresów IPv4 atakujących.

Reasumując, honeypoty – a w szczególności tak rozwinięte środowiska jak T-Pot – to ważny element współczesnych strategii cyberobrony. Pozwalają nie tylko wykrywać i opóźniać ataki, ale przede wszystkim uczyć się przeciwnika. W kontekście akademickim, budowa i analiza honeypota w chmurze (np. z użyciem T-Pot na AWS/Azure) stanowi cenne doświadczenie, łączące wiedzę z zakresu bezpieczeństwa sieci, administracji systemami oraz analizy danych. Honeypoty dostarczają nam przysłowiowej „wędki” – zamiast biernie czekać na atak, aktywnie go prowokujemy i wyciągamy z niego lekcje. Dzięki temu społeczność bezpieczeństwa może proaktywnie reagować na pojawiające się zagrożenia, a same honeypoty pozostają żywym i rozwijającym się obszarem badań naukowych i zastosowań praktycznych w cyberbezpieczeństwie.

Bibliografia

[1] https://www.pcguard.pl/bezpieczenstwo/honeypot

[2] https://www.fortinet.com/resources/cyberglossary/what-is-honeypot

[3] https://bibliotekanauki.pl/articles/419075.pdf

[4] https://pmc.ncbi.nlm.nih.gov/articles/PMC8036602/

[5] https://salientengineering.com/setups-guides/how-to-deploy-and-setup-honeydrop-honeypot-in-aws

[6] https://medium.com/@jiumael/trap-the-hackers-building-and-analyzing-a-t-pot-honeypot-b15f3b6c5ea2

[7] https://bevijaygupta.medium.com/honeypots-101-a-beginners-guide-to-honeypots-f05eab2b3d17

[8] https://www.researchgate.net/figure/Categorization-of-honeypot-systems_fig1_367413267

[9] https://pmc.ncbi.nlm.nih.gov/articles/PMC8036602/

[10] https://github.com/telekom-security/tpotce