Attack Surface Intelligence

Narzedzie do cyklicznego skanowania rozległych sieci (powyżej 10 000 adresów IP). Rozwiązanie ma wykrywać aktywne hosty, identyfikować wystawione usługi, technologie webowe, CMS-y, charakterystyczne nagłówki, ciasteczka oraz rejestrować zmiany widoczne w czasie.

1. Wprowadzenie - opis organizacji, sytuacji i stanu aktualnego

Duże organizacje posiadają rozległą infrastrukturę sieciową, w której część systemów może być widoczna sieciowo tylko okresowo lub odpowiadać na skany sieciowe inaczej w różnych oknach czasowych. Tradycyjne jednorazowe skanowanie nie zawsze pozwala uchwycić pełny obraz powierzchni ataku.

Dla zespołów bezpieczeństwa istotne jest nie tylko wykrycie otwartych portów, ale również zrozumienie, co faktycznie działa pod danym adresem: czy jest tam aplikacja webowa, panel administracyjny, CMS, środowisko testowe, nietypowy baner, interesujące ciasteczko, stary komponent lub technologia potencjalnie wymagająca dalszej analizy.

Problemy:

- Brak konkretnych narzędzi mogących wykonywać takie zadania i integrować się z platformami do zarządzania podatnościami jak DefectDojo (https://defectdojo.com/).
- Wśród podobnych projektów, niespełniających jednak w pełni wymagań, wymienić można SecureCodeBox (https://www.securecodebox.io/).
- Główne wyzwania stojące przed takim rozwiązaniem to rozległa adresacja sieci (ponad 10000 adresów), oraz możliwość tego, że dany adres w zależności od pory dnia może odpowiadać lub nie (w jaki sposób raportować zmianę? Jak określić pewność wyniku identyfikacji technologii? Po jakim czasie braku odpowiedzi w sukcesywnych skanach raportować host jako nieaktywny?)

O Jednostce:

Zamawiającym jest Centrum Bezpieczeństwa Informacji Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie. Jednostka zajmuje się bezpieczeństwem systemów i usług informacji cyfrowej, niektóre z zadań wykonywanych to: monitorowanie sieci, obsługa incydentów, wykonywanie testów penetracyjnych.

---

2. Wyzwanie

Zadaniem uczestników jest stworzenie prototypu systemu do discovery scan dużej infrastruktury.

Rozwiązanie powinno umożliwiać:

1. Wczytanie listy adresów IP, zakresów CIDR lub domen.
2. Wykonanie bezpiecznego skanu discovery.
    - wielowątkowy skan infrastruktury narzędziami jak nmap, które mogą integrować się z Defect Dojo,
    - powinno mieć funkcjonalność sprawdzenia czasu trwania skanu,
    - podział adresów źródłowych do skanowania na optymalne podgrupy.

3. Wykrycie aktywnych hostów.
4. Identyfikację podstawowych usług.
5. Pobranie podstawowych informacji o usługach HTTP/HTTPS, takich jak:
    - tytuł strony,
    - kod odpowiedzi HTTP,
    - nagłówki,
    - ciasteczka,
    - certyfikat TLS,
    - przekierowania.
6. Proste wykrywanie technologii, np.:
    - CMS,
    - framework,
    - serwer aplikacyjny,
    - urządzenie IoT,
7. Dodatkowe rozpoznawanie elementów/konfiguracji usług:
    - środowisko testowe lub staging.
    - panel administracyjny,
8. Porównanie wyników między kolejnymi przebiegami skanu.
9. Oznaczenie nowych, znikających i zmienionych usług.
10. Umożliwienie tworzenia własnych sygnatur detekcyjnych.
11. Rozwiązanie musi mieć formularz, w którym można wprowadzić dane:
    - Te dane muszą również podlegać wyżej wymienionym wymaganiom, tzn. być procesowane tak samo
    - Konieczna jest opcja wprowadzenia ręcznego, a także za pomocą pliku w formacie CSV i/lub JSON
12. Rozwiązanie musi mieć zaimplementowany RBAC, tak aby możliwe było tworzenie i obsługa różnych ról
13. Rozwiązanie powinno mieć zaimplementowane konkretne role:
    - Administracja.
    - Dodawanie, usuwanie, modyfikacja.
    - Modyfikacja i odczyt.
    - Odczyt.
14. Rozwiązanie musi posiadać opcję edycji wprowadzonych wartości niezależnie od stanu skanu i zachowania tych zmian przy kolejnych skanach.
15. Rozwiązanie musi posiadać czytelny interfejs dla użytkownika z dowolną rolą oraz główny dashboard informujący o najważniejszych znaleziskach.

16. Rozwiązanie powinno korzystać z bazy danych, która zapewnia mechanizm szyfrowanie danych w spoczynku (*Encryption At Rest)* dla wszystkich pól posiadających dane wrażliwe.

Moduł sygnatur powinien być projektowany ostrożnie. Ma on służyć przede wszystkim do bezpiecznej detekcji obecności cech lub komponentów, a nie do eksploatowania podatności.

Przykłady bezpiecznych sygnatur:

- obecność konkretnego nagłówka,
- konkretna nazwa ciasteczka,
- hash favicony,
- charakterystyczny fragment HTML,
- ścieżka typu `/login`, `/admin`, `/version`, ale bez prób logowania,
- wykrycie wersji w stopce,
- fingerprint popularnego CMS-a.

Skan infrastruktury powinien wykonywać się wielowątkowo, z użyciem narzędzi, takich jak nmap, oraz mogącymi się integrować z narzędziem DefectDojo. Aplikacja powinna mieć funkcjonalność sprawdzania czasu trwania skanu i podziału adresów na optymalne podgrupy, w zależności od szybkości skanowania. Powinna mieć możliwość ustawiania konkretnych przedziałów czasowych uruchomienia oraz częstotliwości wykonywania skanów. W przypadku bardzo długiego czasu skanowania, jeśli przekroczone zostanie okno czasowe, w którym powinien zostać wykonany, skan powinien zostać zatrzymany i uruchomiony ponownie w kolejnym oknie.

---

3. Oczekiwany rezultat

Oczekiwanym rezultatem jest działający prototyp narzędzia dla zespołu bezpieczeństwa, który pozwala prowadzić bezpieczne discovery scan dużej infrastruktury oraz analizować zmiany w czasie, a także będzie bezpiecznie przechowywać dane i umożliwiać ich edycję.

Minimalny oczekiwany zakres funkcjonalny:

- import listy hostów lub zakresów,
- import wyników skanów w formie JSON lub CSV,
- obsługa dużych zakresów, np. ponad 10 000 IP,
- konfiguracja skanu - okno czasowe i częstotliwość uruchamiania,
- ograniczanie tempa zapytań,
- wykrywanie aktywnych hostów,
- podstawowa identyfikacja usług,
- zapis wyników do bazy lub pliku,
- porównanie dwóch skanów,
- dashboard lub raport pokazujący najważniejsze zmiany,
- eksport wyników do JSON lub CSV,
- rozgraniczenie funkcjonalności na role,

- możliwość edycji danych istniejących już skanów, wyników i konfiguracji,
- ochrona danych z wykonanych skanów (Szyfrowanie)

Mile widziany dodatkowy zakres:

- harmonogram cyklicznych skanów,
- wykrywanie różnic zależnych od pory dnia,
- ocena „ciekawości" lub priorytetu analitycznego,
- powtarzanie pomiarów dla hostów niestabilnych,
- moduł HTTP/HTTPS fingerprinting,
- proste wykrywanie CMS-ów lub technologii,
- grupowanie podobnych usług,
- obsługa tagów,
- możliwość eksportu wybranych obserwacji do DefectDojo jako finding typu informacyjnego lub jako evidence do dalszej analizy,
- wizualizacja zmian w czasie,
- integracja z pasywnymi źródłami danych, jeżeli zostaną udostępnione.

---

4. Wymagania formalne

Projekt powinien zawierać:

1. Repozytorium kodu źródłowego
2. Dokumentację techniczną
3. Prezentację do 10 slajdów obejmującą:
    - Działające demo, uruchomione lokalnie lub w przygotowanym środowisku testowym.

---

5. Wymagania techniczne

Dozwolone technologie

Uczestnicy mogą użyć dowolnych bezpłatnych, open-source oraz innych, niewymagających licencji technologii. Sugerowanym rozwiązaniem jest wykorzystanie języków programowania charakteryzujących się niskim użyciem zasobów, takie jak Go, Rust, C++.

---

6. Sposób testowania i/lub walidacji

Rozwiązania zostaną przetestowane w przygotowanym środowisku laboratoryjnym, zawierającym zestaw hostów i usług imitujących zróżnicowaną infrastrukturę.

Test będzie obejmował co najmniej dwa przebiegi skanu, aby ocenić wykrywanie zmian.

Test obejmie również możliwość edycji znalezisk, ich import, a także rozgraniczenie ról.

Weryfikowane będą:

1. Poprawność wykrycia aktywnych hostów.
2. Poprawność identyfikacji usług.
3. Trafność fingerprintingu HTTP/HTTPS.
4. Umiejętność wykrycia CMS-ów lub technologii.
5. Wykrywanie ciekawych nagłówków i ciasteczek.
6. Działanie customowych sygnatur.
7. Porównanie wyników między skanami.
8. Obsługa hostów niestabilnych lub zmieniających odpowiedź.
9. Bezpieczeństwo i ostrożność działania skanera.
10. Czytelność dashboardu lub raportu.
11. Import oraz export danych.
12. Rozgraniczenie funkcjonalności między role.

---

7. Dostępne zasoby

Uczestnikom zostaną udostępnione:

1. Konfiguracja do uruchomienia kontenerów, służących jako środowisko testowe.
2. Opis zasad bezpiecznego skanowania, w tym limity ruchu, jeśli istnieją.
3. Listę dozwolonych portów do testowania.
4. Przykładowy zbiór wyników referencyjnych dla części infrastruktury.
5. Zestaw przykładowych sygnatur do modułu customowych sygnatur.
6. Przykładowy format eksportu/importu JSON/CSV.
7. Opcjonalnie:
    - testową instancję DefectDojo,
    - kontener Docker z przykładowymi usługami,
    - przykładowe domeny i subdomeny,
    - listę „ciekawych artefaktów”, które narzędzie powinno wykryć.

Zasoby zostaną udostępnione uczestnikom poprzez kanał na Discordzie lub stacjonarnie na stanowisku AGH.

Dodatkowe źródła:

API: https://demo.defectdojo.org/api/v2/oa3/swagger-ui/

Dokumentacja Nmap: https://nmap.org/docs.html

Dokumentacja SecureCodeBox: https://www.securecodebox.io/docs/getting-started/installation

---

8. Kryteria oceny

Kryteria oceny:

- Skuteczność discovery i poprawność identyfikacji usług w dużej infrastrukturze - 30%
- Wykrywanie zmian w czasie, obsługa niestabilnych hostów i użyteczność raportowania - 15%
- Jakość techniczna, dokumentacja, prezentacja, łatwość uruchomienia i importu/eksportu danych oraz integracja z innymi usługami (DefectDojo) - 20%
- Bezpieczeństwo działania, kontrola intensywności skanowania i przejrzystość prezentacji wyników w dashboardzie - 15%
- Jakość fingerprintingu, wykrywania technologii, CMS-ów, nagłówków, ciasteczek i sygnatur - 10%
- Rozgraniczenie funkcjonalności na role i odpowiednie zabezpieczenie danych - 10%

---

9. Dodatkowe uwagi / kontekst wdrożeniowy

Najlepsze rozwiązanie będzie wdrożone i aktywnie rozwijane w ramach działalności Centrum Bezpieczeństwa Informacji i przyczyniać się do podnoszenia dojrzałość procesów bezpieczeństwa na Akademii Górniczo-Hutniczej. Współpraca AGH z innymi uczelniami i partnerami, którzy mierzą się z podobnymi wyzwaniami, może wpłynąć na upowszechnianie praktycznej wiedzy, a także narzędzi opracowanych w ramach niniejszego projektu.

---

10. Kontakt

Mentorzy:

Kacper Łagodowski - klagodowski@agh.edu.pl

Paweł Skalny – skalny@agh.edu.pl

Tomasz Nal – naltomasz@agh.edu.pl
