Miłosz Kutyła (318427), Jakub Ossowski (318435),
Jan Walczak (318456), Patryk Jankowicz (318422)

Politechnika Warszawska

Sprawozdanie z realizacji projektu BEKOM nr 1

5 grudnia 2023

Spis treści

Oświadczenie

Niniejszy dokument to sprawozdanie z realizacji projektu w ramach przedmiotu BEKOM. Oświadczamy, że ta praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu BEKOM, została wykonana przez nas samodzielnie.

1. Wstęp

1.1. Scenariusz

Firma, dla której pracuje zespół WOJK, poprosiła nas o pomoc z zabezpieczeniem nowej aplikacji www. Aplikacja składa się z:

- serwera aplikacyjnego, który serwuje usługę,
- bazy danych, która gromadzi dane powiązane z działaniem aplikacji.

Zadaniem zespołu jest przedstawienie architektury sieciowej, która będzie uwzględniała najlepsze praktyki bezpieczeństwa m.in.:

- zastosowanie szyfrowanego połączenia tam, gdzie jest to wymagane,
- zastosowanie filtrowania ruchu w kontekście zdefiniowanego ruchu,
- zastosowania weryfikacji bezpieczeństwa danych użytkowników wprowadzanych do aplikacji,
- zbieranie logów z nieprawidłowych zdarzeń w sieci,
- odpowiednie przygotowanie maszyn, na których będzie działać aplikacja www.

1.2. Wymagania – projekt

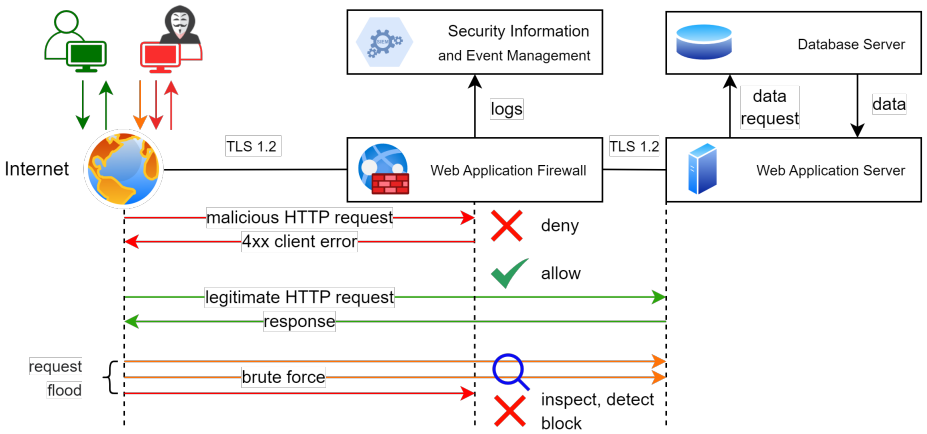
Projekt architektury powinien uwzględnić kwestie bezpieczeństwa związane z separacją poszczególnych komponentów usługi www oraz kwestie bezpieczeństwa związane z siecią i serwerami.

2. Design wysokopoziomowy

Docelowa architektura sieci powinna składać się z następujących komponentów:

- serwera aplikacyjnego,
- serwera bazodanowego,
- Web Application Firewalla dedykowanego dla aplikacji,
- rozwiązania SIEM.

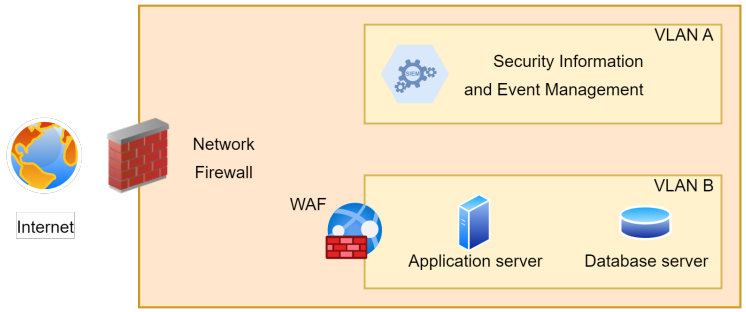
W celu zobrazowania komunikacji pomiędzy komponentami docelowej sieci, przygotowaliśmy diagram połączeń między systemami. Przedstawia on typy danych i wiadomości wymienianych pomiędzy: systemami oraz klientami a systemami. Ruch powinien być szyfrowany end-to-end pomiędzy Internetem i WAFem, a nastepnie pomiędzy WAFem i serwerem aplikacyjnym. Diagram połączeń przedstawia rysunek 1.



Rysunek 1: Diagram połączeń między systemami

Poszczególne komponenty powinny zostać odseparowane poprzez umieszczenie ich w różnych segmentach. Ze względu na wzajemną zależność WAF, serwer aplikacyjny i serwer bazodanowy powinny zostać umieszczone w jednym segmencie. Podobnie, ze względu na brak bezpośrednich powiązań z isnnymi systemami, SIEM powinien zostać umieszczony w osobnym segmencie. W celu zwiększenia bezpieczeństwa, rekomendowane byłoby dodatkowo odseparowanie serwera bazodanowego poprzez umieszczenie go za firewallem bazodanowym. Ze względu na dodatkowe koszty implementacyjne, nie zdecydowaliśmy się na dodatkowe odseparowanie serwera bazodanowego.

W celu implementacji odpowiedniej segmentacji, zdecydowaliśmy się na dodanie dodatkowego komponentu – routera z funkcją firewalla na styku z siecią Internet. Zasugerowaną segmentację (na schemacie wysokopoziomowym) przedstawia rysunek 2.



Rysunek 2: Sugerowana topologia sieci: schemat wysokopoziomowy

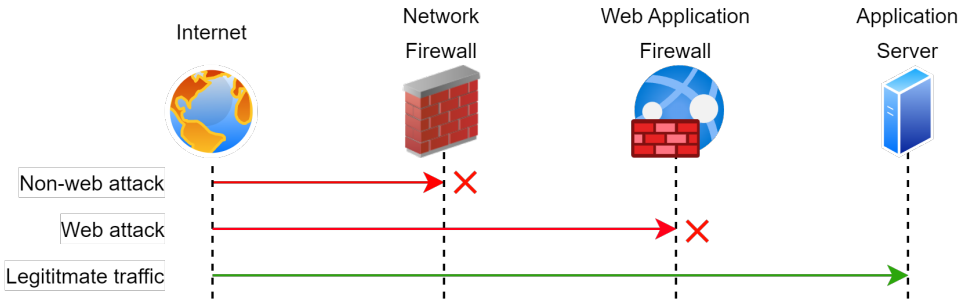
Firewall na brzegowym routerze powinien mieć reguły, które:

- zezwalają na ruch TCP (TLS v1.2) z Internetu (klienta) do WAFa. Ograniczenie przez: adres IP (docelowy: 192.168.0.102), port (443), protokół.
- zezwalają na ruch TCP (TLS v1.2) z WAFa do Internetu (klienta). Ograniczenie przez: adres IP (źródłowy: 192.168.0.102), port (443), protokół.
- zezwalają na ruch TCP z WAFa do SIEMa (logi). Ograniczenie przez: adresy IP (źródłowy, docelowy), porty, protokół.
- blokują każdy inny ruch.

Wprowadzenie dodatkowego routera i firewalla umożliwia:

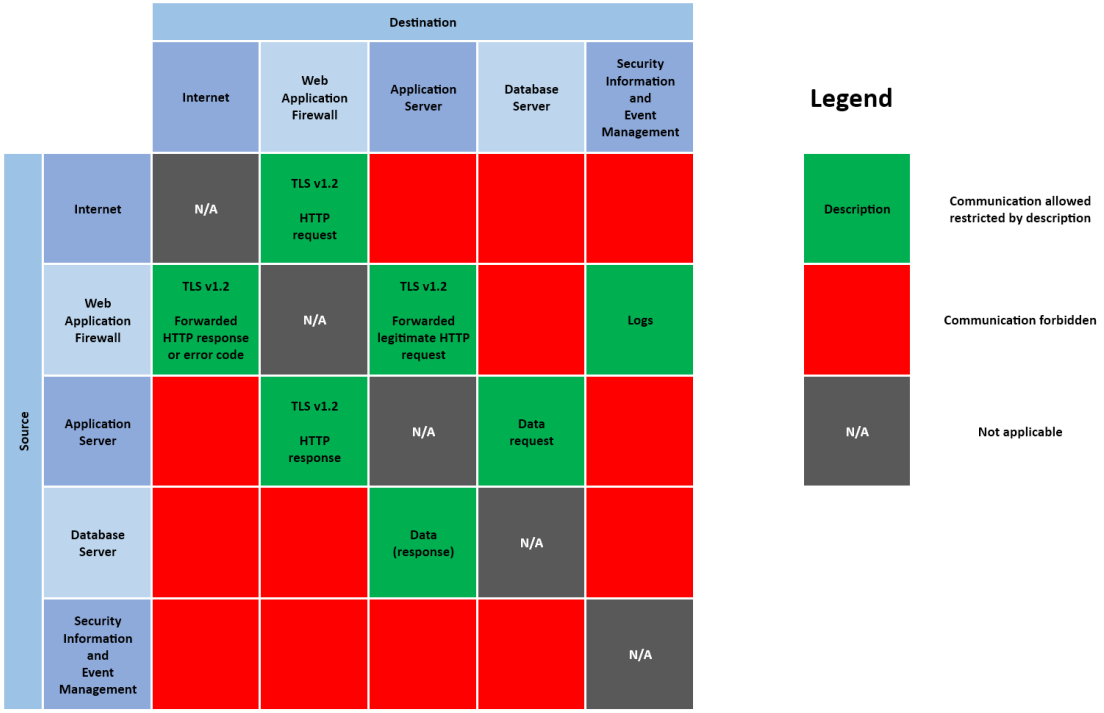
- wprowadzenie routingu między VLANami: dla ruchu z WAFa do SIEMa,
- wprowadzenie dodatkowej warstwy ochrony sieci,
- zmniejszenie objętości niedozwolonego ruchu trafiającego do sieci wewnętrznej,
- zmniejszenie objętości złośliwego ruchu trafiającego do WAF.

Odróżnienie funkcjonalności firewalla sieciowego od aplikacyjnego przedstawia rysunek 3.



Rysunek 3: Odróżnienie funkcjonalności Network FW od WAF

Zobrazowanie połączeń między systemami umożliwiło nam utworzenie macierzy komunikacji. Macierz komunikacji dla komponentów sieci przedstawia rysunek 4.



Rysunek 4: Macierz komunikacji

3. Wybór rozwiązań technologicznych

3.1. Konfiguracja hostów

Całość oprogramowania zostanie zainstalowana na systemie operacyjnym typu Linux w wersji 22.04. Hosty WAF, App Server oraz Database Server zostaną postawione na wersji serwerowej Ubuntu, natomiast host SIEM zostanie postawiony na wersji Desktop. Taka konfiguracja pozwoli ujednolicić strukturę naszej sieci i ułatwić zarządzanie. Administrator systemu może być specjalistą od danego rodzaju systemu. Dodatkowo taka jednolitość przyspieszy procedurę utwardzania hostów poprzez wykorzystanie tych samych regulacji CIS dla wszystkich hostów.

3.2. Security Information and Event Management

Wybranym przez nas oprogramowaniem typu SIEM jest `Wazuh`. Jest to oprogramowanie typu open-source z przejrzystym i łatwym w użytkowaniu interfejsem graficznym umożliwiającym efektywne analizowanie logów systemowych, zarządzanie zdarzeniami oraz detekcją włamań. Zapewnia on możliwość integracji z narzędziami `Elastick Stack` oraz komunikację z agentem zainstalowanym na Web Application Firewall.

3.3. Web Application Firewall

Do implementacji aplikacyjnego firewall'a działającego jako reverse proxy zdecydowaliśmy się na wykorzystanie serwera `nginx`. Nasz wybór kierowany był głównie popularnością tego silnika, powszechnością dokumentacji oraz społeczności Internetowych, w których będziemy mogli szukać rozwiązania potencjalnie napotkanych problemów na drodze implementacji. Dodatkowym atutem jest licencja BSD, na której jest wydawany silnik `nginx`, co sprawia, że jest on odpowiedni do rozwiązań akademickich. Natomiast do implementacji samego firewall'a, który będzie uruchomiony na powyższym serwerze, wykorzystamy open-source'owe rozwiązanie `Mod Security`. Umożliwi nam to łatwe podpięcie WAFa do silnika serwera proxy dzięki istnieniu modułów, które takie połączanie znacznie upraszczają. Kolejnym czynnikiem, którym się sugerowaliśmy, jest OWASP ModSecurity Core Rule Set. Istnienie takiego zestawu reguł proponowanych przez organizację OWASP utwierdziło nas w przekonaniu, że jest to zarówno popularne, jak i bezpieczne rozwiązanie przy odpowiedniej konfiguracji.

3.4. Application Server

Aplikację napiszemy sami, wykorzystując darmowy framework `Flask` w języku Python. Jest to popularna biblioteka, której przeznaczeniem jest tworzenie aplikacji webowych. Owa aplikacja, umożliwi nawiązanie połączenia z serwerem bazodanowym opartym na `PostgreSQL`. W szczególności umożliwi nam: wywołanie zapytań SQL, zdefiniowanie działań w zależności od podstrony, a także pobranie/wyświetlenie danych wpisanych przez użytkownika lub zwróconych przez bazę danych. Usługę wystawimy na porcie 443 (szyfrowany kanał) używając `Apache2`. Do uwierzytelnienia użytkownika, w celu zapewnienia weryfikacji bezpieczeństwa danych użytkowników wprowadzanych do aplikacji, zastosujemy Basic.

3.5. Database Server

Baza danych będzie obsługiwana przez jedną z najpopularniejszych, open-source'owych usług – `PostgreSQL`. Dzięki temu dostępnych jest wiele wartościowych materiałów, więc konfiguracja powinna się udać bez większych problemów. Dodatkowym atutem `PostgreSQL` jest dostępność biblioteki `psycopg2` języka `Python` pozwalającej na prostą integrację z serwerem aplikacyjnym.

4. Design niskopoziomowy

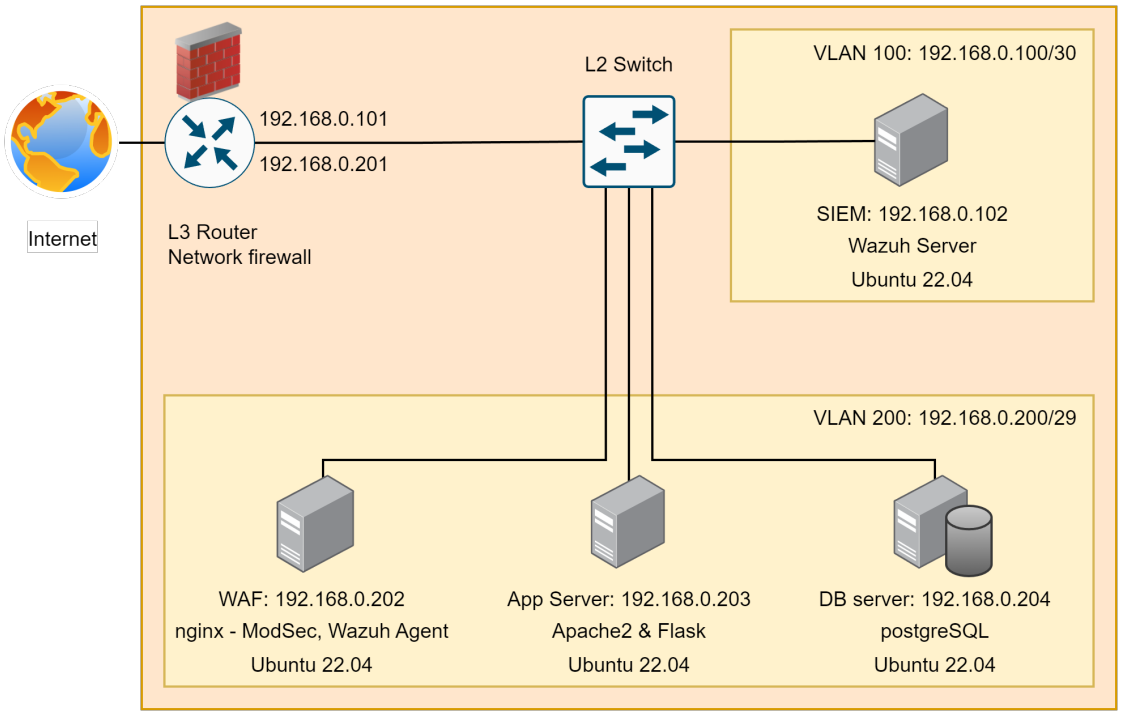
Zgodnie z designem wysokopoziomowym, komponenty sieci zostały odseparowane poprzez umieszczenie ich w odpowiednich segmentach – osobnych VLANach. Dzięki odpowiedniemu podzieleniu komponentów sieci, jesteśmy w stanie zasugerować wstępną adresację sieci:

- `192.168.0.100/30` dla VLAN 100 przeznaczonego dla SIEMa. Wykorzystamy pełną pulę adresową (adresacja SIEM, routera).

- `192.168.0.200/29` dla VLAN 200 przeznaczonego dla WAF, serwera aplikacyjnego i bazodanowego. Wykorzystamy możliwie największą pulę adresową (adresacja WAF, serwera aplikacyjnego, serwera bazodanowego, routera).

Ze względu na mały rozmiar sieci, zdecydowaliśmy się wprowadzić dwupoziomową architekturę sieci (switch, router). W przypadku chęci rozbudowy segmentów (większa ilość urządzeń) lub wprowadzenia dodatkowej segmentacji (więcej VLANów), rekomendowane byłoby wprowadzenie trójpoziomowej architektury (access switch, distribution switch, router). Zmniejszyłoby to obciążenie poszczególnych switchy. Ze względu na dodatkowe koszty realizacyjne, nie zdecydowaliśmy się na wprowadzenie dwóch poziomów switchów.

Sugerowany topologię sieci na schemacie niskopoziomowym przedstawia rysunek 5.



Rysunek 5: Sugerowana topologia sieci: schemat niskopoziomowy

5. Wnioski i podsumowanie

Zadanie uważamy za ciekawe w realizacji. Było to nasze pierwsze spotkanie z projektowaniem sieci i architektury systemów "od zera". Niezwykle pomocne okazało się zobrazowanie kolejnych projektowanych etapów na diagramach i przy pomocy macierzy komunikacji. Dzięki temu proces projektowania przebiegł nam bardzo sprawnie.

W ramach realizacji projektu wykorzystaliśmy następujące narzędzia:

- Overleaf i Latex – do utworzenia niniejszego sprawozdania.
- draw.io – do utworzenia diagramów.
- Microsoft Word – do utworzenia macierzy komunikacji.