

Michalska Aleksandra, Pelczar Jacek
Komputerowe i Sieciowe Systemy Operacyjne
Politechnika Warszawska
Semestr 23Z
Grupa 5CB2

Sprawozdanie z projektu
Projekt systemu informatycznego dla rozgłośni muzycznej

Spis treści

1. Wstęp i cel projektu	2
2. Skład zespołu projektowego	2
3. Zakres projektu	2
4. Produkty w projekcie	2
5. Ograniczenia	2
6. Identyfikacja ryzyka	3
7. Harmonogram projektu	4
8. Architektura systemu	5
9. Implementacja wybranych usług	6
Serwer webowy	6
Serwer WAF + reverse proxy	6
Serwer bazodanowy	7
Serwer plików	7
Firewall i konfiguracja sieci	8
10. Hardening i inne czynności zwiększające bezpieczeństwo systemu	10
10.1. Logowanie zdarzeń i rotacja logów	10
10.2. Skanowanie systemów operacyjnych przeciw malware	10
10.3. Automatyzacja usług bezpieczeństwa	11
10.4. Wyłączenie ipv6	11
10.5. Bezpieczne hasła użytkowników	11
10.6. Uprawnienia sudo	12
10.7. Wyłączenie hibernacji	12
10.8. MAC: AppArmor	12
11. Instrukcje utrzymywania	13
11.1. Ogólne zalecenia	13
11.2. Serwer WAF + reverse proxy	13
11.3. Serwer webowy	13
11.4. Serwer bazodanowy	13
11.5. Serwer plików	14
12. Testy i kosztorys projektu	14
12.1. Testy funkcjonalne	14
12.2. Testy niefunkcjonalne	17
12.3. Kosztorys	18
12.4. Specyfikacja techniczna	18
13. Podsumowanie i wnioski końcowe	18

1. Wstęp i cel projektu

Przedmiotem tego projektu jest zaplanowanie i wdrożenie wybranego systemu informatycznego na podstawie zdobytej dotychczas wiedzy podczas wykładów i laboratoriów. Jako zespół zdecydowaliśmy się na zrealizowanie systemu zarządzania pracą rozgłośni radiowej.

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

2. Skład zespołu projektowego

Projekt jest wykonywany przez zespół w składzie:

- Aleksandra Michalska
- Jacek Pelczar

Role obojga członków zespołu to co-leader, projektant i wykonawca.

3. Zakres projektu

Stworzymy system zarządzania rozgłośnią radiową. Klient rozgłośni będzie mógł sprawdzić plan transmisji na specjalnej stronie WWW. Natomiast pracownik będzie mógł za jej pomocą zarządzać pracą rozgłośni np. dodawać utwory, audycje, reklamy, nowych pracowników. Pracownicy będą podzieleni na role i każda z nich będzie miała odpowiednie uprawnienia np. księgowy może ustalać wynagrodzenie pracowników, a redaktor może ustalać plan transmisji.

4. Produkty w projekcie

System rozgłośni radiowej będzie się składał z następujących elementów:

- **serwer webowy** zapewniający możliwość dostępu do informacji w zależności od uprawnień klienta
- **serwer bazodanowy** przechowujący dane o użytkownikach i związanych z nimi informacjach
- **serwer plików** przechowujący zasoby rozgłośni np. nagrania odtwarzanych utworów
- **serwer administracyjny** służący do zarządzania elementami sieci prywatnej
- **firewall + router** jako punkt styku sieci prywatnej i publicznej i zapewnienia bezpieczeństwa sieci prywatnych
- **serwer reverse proxy** jako punkt dostępowy do serwera WWW i wzmacniania jego bezpieczeństwa.

Dodatkowo zostanie przygotowane urządzenie-klient służące do korzystania z powyższych usług.

5. Ograniczenia

Budżet: Ograniczenia finansowe mogą wpływać na dostępność zasobów potrzebnych do zakupu sprzętu, oprogramowania, licencji, usług hostingowych, zatrudnienia specjalistów, szkoleń i innych czynników niezbędnych do pełnego wdrożenia i funkcjonowania systemu.

Terminy: Zespół developerski może być ograniczony czasem na dostarczenie gotowego systemu. Ograniczenia czasowe wynikające z narzuconego terminu wykonania (koniec semestru), wykonywania wielu projektów jednocześnie (inne przedmioty) i innych czynników mogą wpływać na jakość prac, testów i dokumentacji.

Zasoby ludzkie: Posiadanie odpowiednio wykwalifikowanych członków zespołu developerskiego może być wyzwaniem. Niedobór specjalistów w dziedzinie programowania, bezpieczeństwa informatycznego czy administracji systemami może utrudniać rozwój systemu w odpowiednim tempie i skali. W przypadku przyjętych zleceniodajców ograniczeniem jest niewielka liczba członków zespołu, niepełne wykształcenie oraz małe doświadczenie zawodowe w branży. Ich zaletą jest cena - pracują za zaliczenie przedmiotu.

Wymagania użytkowników: Oczekiwania użytkowników systemu radiowego mogą być różne i wyznaczać konkretne wymagania dla zespołu developerskiego. Może to oznaczać dostosowania interfejsu, funkcjonalności czy sposobu dostępu do treści w celu zadowolenia odbiorców i zwiększenia konkurencyjności systemu.

6. Identyfikacja ryzyka

Wadliwa konfiguracja systemu: Ryzyko wynikające z błędów w konfiguracji lub zarządzaniu systemem, takie jak niewłaściwe ustawienia firewalla czy błędne konfiguracje serwerów, które mogą prowadzić do problemów w dostępności systemu, niewłaściwego funkcjonowania czy zagrożeń bezpieczeństwa.

Ryzyko jest mitygowane poprzez wykonanie testów funkcjonalnych.

Naruszenie poufności danych: Serwer bazodanowy przechowuje informacje o pracownikach, reklamodawcach, utworach, licencjach itp. Istnieje ryzyko naruszenia poufności tych danych przez włamanie się lub nieautoryzowany dostęp, w wyniku którego mogą zostać naruszone dobra osobiste oraz umowy biznesowe z właścicielami prawnymi materiałów artystycznych.

Ryzyko jest mitygowane poprzez minimalizację uprawnień do bazy danych, zastosowanie firewalla oraz WAF, a także implementację strony internetowej mającej na celu zapobieganie m.in. atakom SQL Injection.

Zdarzenia losowe: System informatyczny rozgłośni radiowej może być narażony na zdarzenia losowe, takie jak awarie zasilania, powodzie, pożary, katastrofy naturalne itp. W przypadku takich zdarzeń, system może ulec uszkodzeniu lub niedostępności, co może prowadzić do przerw w nadawaniu i strat finansowych dla rozgłośni radiowej.

Ryzyko jest częściowo mitygowane poprzez wykorzystanie urządzeń typu UPS, co może zminimalizować szkody spowodowane nagłą utratą zasilania. Więcej mitygacji nie wdrożono ze względu na ograniczony budżet i niski stopień krytyczności systemu. W innym przypadku zalecane byłoby wprowadzenie redundancji dostępu do usług czasu rzeczywistego (web), wdrożenie polityki kopii zapasowych i planów ciągłości zadania.

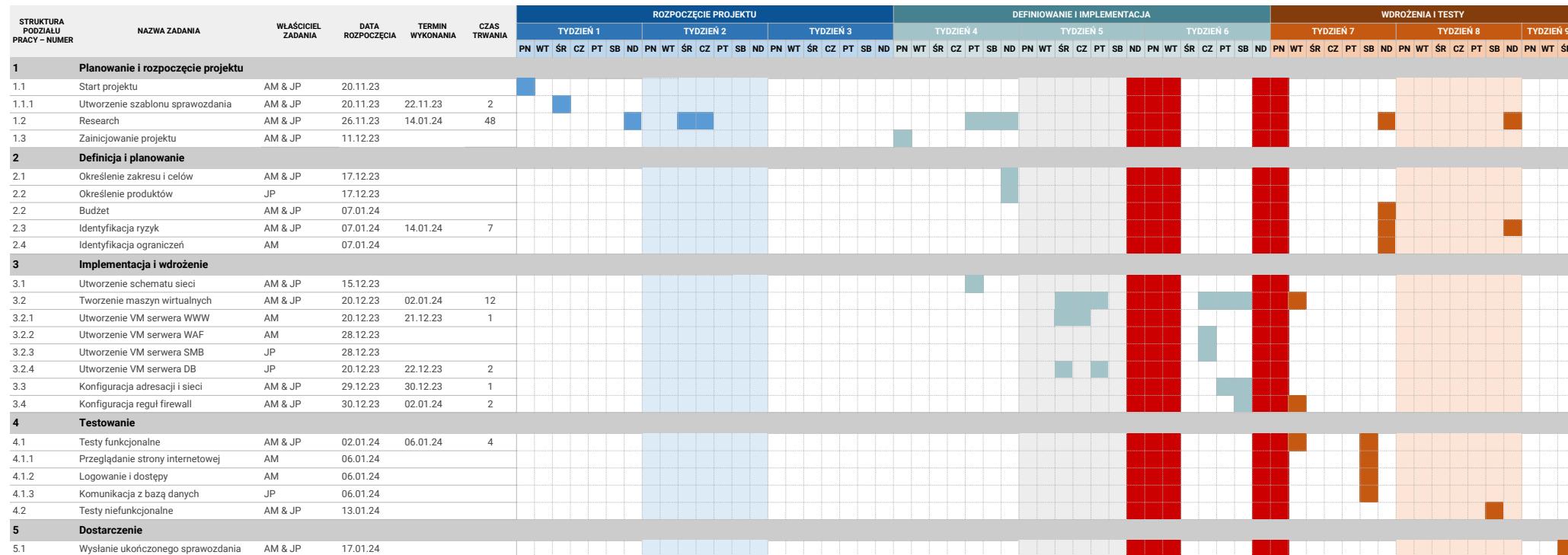
Rozwój systemu ponad dostępne zasoby: Wraz ze zwiększającą się popularnością działalności rozgłośni i rozszerzający się FTE może wzrosnąć zapotrzebowanie na dostęp do świadczonych przez system usług. Przygotowany w tym projekcie system jest gotowy na większe niż średnie planowane zapotrzebowanie, jednak ma swoje ograniczenia wynikające zarówno z software'u jak i ograniczeń sprzętowych.

W razie materializacji ryzyka zalecane jest zwiększenie redundancji usług, modyfikację architektury na mikroservisową, a także zwiększenie parametrów sprzętowych tj. RAM.

Zidentyfikowane ryzyko	Ocena ryzyka	Uzasadnienie	Mitygowane?
Wadliwa konfiguracja systemu	Niskie	Świeży system, przetestowany, aktualne oprogramowanie.	TAK
Naruszenie poufności danych	Umiarkowane	Bezpieczna konfiguracja, firewall'e, wymagane uwierzytelnienia do systemów	TAK
Zdarzenia losowe	Umiarkowane	Klimat umiarkowany, zasilanie zapasowe, ale brak polityk backup i planów ciągłości działania	częściowo
Rozwój systemu ponad dostępne zasoby	Podwyższone	Brak testów wydajnościowych, brak doświadczenia zawodowego developerów	częściowo

7. Harmonogram projektu

TYTUŁ PROJEKTU	System informatyczny rozgłośni radiowej
WYKONAWCY PROJEKTU	AM & JP



8. Architektura systemu

System składa się na 6 hostów: Serwer Reverse Proxy + WAF, Serwer WWW, Serwer Plików, Serwer Bazodanowy, Komputer Administratora, Firewall.

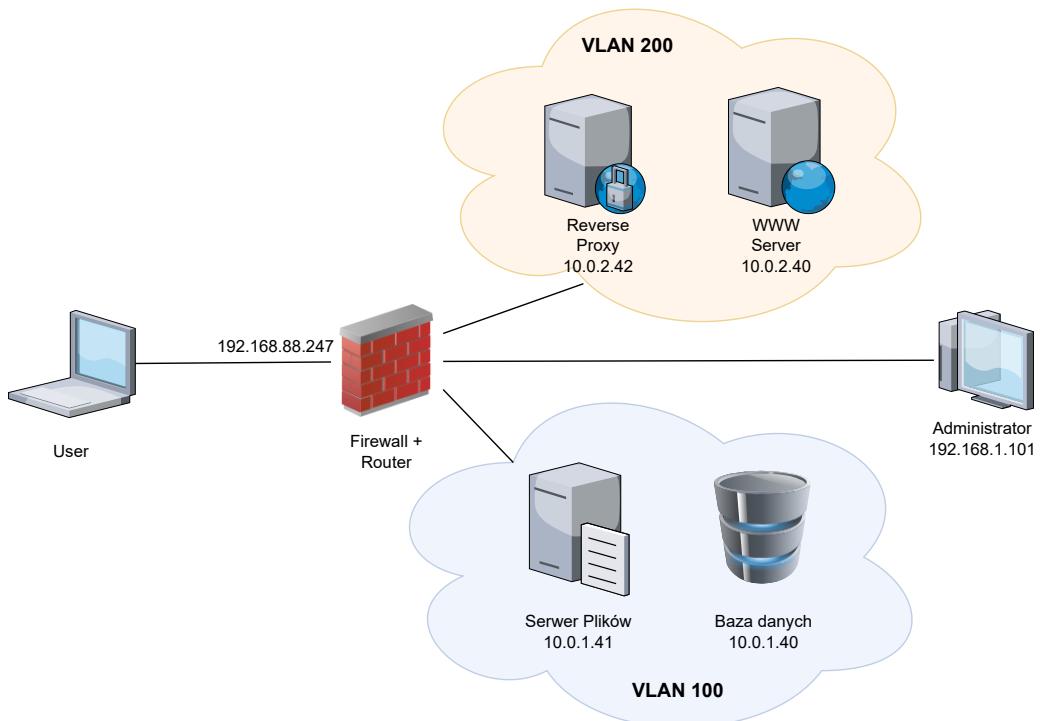
Połączenie z internetem realizuje Firewall, który jednocześnie jest routерem, switchem oraz firewallem. Zabezpiecza on sieć przed nieuprawnionym dostępem oraz zarządza strukturą sieci m.in. podziałem na vlan-y. Reverse Proxy + WAF służy zabezpieczeniu serwera WWW przed potencjalnymi atakami złośliwych aktorów. W bazie danych przechowywane są dane z których korzysta serwer WWW.

Serwer plików służy do udostępniania zasobów (plików) w sieci lokalnej organizacji.

W systemie znajduje się także jeden host administratora, który służy do konfiguracji i zarządzania całą infrastrukturą.

System projektowany jest jako stabilna całość, ale może być rozbudowywany o np. komputery pracowników w sieci lokalnej.

Vlan'ami zarządza switch łączący każdą strefę z routерem.



Rys. 1: Schemat infrastruktury

9. Implementacja wybranych usług

Serwer webowy

Strona internetowa powstała w języku Java (JDK 21.0.1) z wykorzystaniem framework'a Spring Boot w wersji 2.7.7 oraz silnika szablonów Thymeleaf. Ponieważ dostęp do danych jest zależny od uprawnień użytkownika, zaimplementowano mechanizm logowania z różnymi poziomami uprawnień (administrator, zwykły pracownik, oraz gość niezalogowany) z wykorzystaniem narzędzia Spring Security.

Plik konfiguracyjny *application.properties* prezentuje się następująco:

```
server.error.whitelabel.enabled=false

spring.datasource.url=jdbc:oracle:thin:@10.0.1.40:1521/freepdb1
spring.datasource.username=[nazwa użytkownika]
spring.datasource.password=[hasło]
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
logging.level.root=INFO

spring.main.banner-mode=console

server.ssl.certificate=/etc/ssl/certs/ssl-cert-snakeoil.pem
server.ssl.certificate-private-key=/etc/ssl/private/ssl-cert-snakeoil.key
server.port=5000
```

Powyzszy plik zawiera m.in dane niezbędne do połączenia z bazą danych, rodzaj logowanych informacji (INFO jest dosyć niski, jednak adekwatny do spodziewanego ruchu).

Do zapewnienia bezpiecznego połączenia https wykorzystano wygenerowane, samodzielnie podpisane klucze "snakeoil" - zalecane tylko do stosowania w fazie developement lub debug, ale na potrzeby tego zadania wystarczające do przedstawienia Proof of Concept. Przed wdrożeniem systemu silnie sugerowany jest zakup certyfikatu i adekwatna modyfikacja powyższego pliku.

Zadeklarowano również obecność sterownika pozwalającego na integrację aplikacji napisanej w języku JAVA z bazą danych. W tym przypadku skorzystano ze sterownika OJDBC, ponieważ zastosowana baza to baza Oracle.

W celu udostępnienia usługi webowej wykorzystano kontener aplikacji sieciowych Apache Tomcat (9.0.70). Aplikację wystawiono na porcie 5000. W celu minimalizacji konieczności każdorazowej manualnej konfiguracji usług (co mogłoby potencjalnie narazić na błędy) skonfigurowano każdorazowy start usługi po reboocie poprzez dodanie odpowiedniego wpisu do *cron*.

Serwer WAF + reverse proxy

Wdrożono serwer pełniący funkcję Reverse Proxy z funkcjonalnością WAF wykorzystując oprogramowanie *nginx*.

Instalacji nginx dokonano z kodu źródłowego poprzez pobranie narzędzia ze strony: <https://nginx.org/en/download.html> w wersji *nginx-1.22.0*. Konfiguracja przedstawia się następująco:

```
./configure --sbin-path=/usr/bin/nginx --conf-path=/etc/nginx/nginx.conf
→ --error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log
→ --with-pcre --pid-path=/var/run/nginx.pid --add-dynamic-module=../../naxsi/naxsi_src
→ --with-http_ssl_module
```

Wartym odnotowania jest tag *--add-dynamic-module=../../naxsi/naxsi_src*, który wskazuje na potrzebę wprowadzenia do procesu komplikacji modułu naxsi.

Naxsi jest modułem zewnętrznym dla nginx chroniącym przed atakami tj. SQL Injection czy XSS. Pozyskano i zainstalowano z <https://github.com/wargio/naxsi>.

Po komplikacji i instalacji narzędzi przystąpiono do konfiguracji serwera aby pełnił rolę *reverse proxy*.

Proxowana jest witryna pod adresem lokalnym [https://\[adres\]:5000](https://[adres]:5000). Do zapewnienia bezpiecznego połączenia https wykorzystano wygenerowane, samodzielnie podpisane klucze "snakeoil" - analogicznie jak dla serwera www.

W celu minimalizacji konieczności każdorazowej manualnej konfiguracji usług (co mogłoby potencjalnie narazić na błędy) skonfigurowano każdorazowy start usługi po rebootie poprzez dodanie odpowiedniego wpisu do *cron*.

Serwer bazodanowy

Jako bazę danych wybrano Oracle Database Express Edition. Jest to darmowa wersja bazy danych z rodziny Oracle, której specyfikacja wystarcza na potrzeby tego projektu. Oprogramowanie to jest także dostępne do użytku komercyjnego bez opłat.

Zainstalowano system operacyjny Oracle Linux zgodnie z instrukcją producenta <https://docs.oracle.com/en/operating-systems/oracle-linux/8/install/#Oracle-Linux-8> Następnie zainstalowano Oracle Database XE według wskazówek na oficjalnej stronie <https://www.oracle.com/pl/database/technologies/appdev/xe/quickstart.html> Utworzono bazę oraz PDB, gdzie utworzono użytkownika oraz załadowano skrypt SQL tworzący niezbędne tabele i inne obiekty.

Serwer plików

Jako serwer plików wybrano system Debian 12 Bookworm zainstalowanym pakietem Samba, który pozwala na udostępnianie zasobów z wykorzystaniem protokołu SMB. Do fizycznego serwera dołączono dodatkowy dysk HDD, do wyłącznego wykorzystania go przez serwer plików, jako miejsce zamontowania smb share. Utworzono 2 *share*, jeden dostępny bez autoryzacji, a do drugiego ustawiono logowanie poprzez hasło. W tym celu utworzono dodatkowego użytkownika, który służy do logowania przy połączeniu smb. Konfiguracja serwera Samby prezentuje się następująco:

```
[guest]
    path = /srv/samba/guest/
    read only = no
    guest ok = yes
    guest only = yes
    comment = For less important documents TLP:CLEAR

[office]
    path = /srv/samba/office/
    public = no
    writable = yes
    comment = For local office use
    printable = no
    guest ok = no
    security = user
```

Rys. 2: Fragment pliku konfiguracyjnego serwera Samba

Firewall i konfiguracja sieci

Ustalono dopuszczalny i niezbędny ruch w sieci wewnętrznej firmy, jak i dopuszczony ruch na zewnątrz. Podsumowanie tych decyzji zostało przedstawione w formie macierzy komunikacji poniżej:

	Internet (WAN)	WAF	Serwer webowy	Serwer bazodanowy	Serwer plików	Admin	Użytkownicy w sieci LAN
Internet (WAN)		✓	✗	✗	✗	✗	✓
WAF	✓		✓	✗	✗	✓	✓
Serwer webowy	✗	✓		✓	✗	✓	✗
Serwer bazodanowy	✗	✗	✓		✗	✓	✗
Serwer plików	✗	✗	✗	✗		✓	✓
Admin	✗	✓	✓	✓	✓		✓
Użytkownicy w sieci LAN	✓	✓	✗	✗	✓	✓	

- komunikacja zablokowana
- komunikacja dozwolona
- nie dotyczy

Rys. 3: Macierz komunikacji

W sieci zostały wyszczególnione dwie strefy VLAN w celu zapewnienia większego bezpieczeństwa poprzez zmniejszenie domeny broadcast. VLANy są zarządzane automatycznie przez wirtualny switch wbudowany w Proxmox.

Następnie przystąpiono do konfiguracji firewalla (pfSense) zgodnie z przyjętymi wcześniej założeniami:

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	✓ 1/1.51 MiB	*	*	*	LAN Address	44566 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	✓ 0/1.92 MiB	IPv4 *	192.168.1.101	*	VLANYELLOW subnets	*	*	none		admin to vlan yellow	
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	192.168.1.101	*	LAN address	*	*	none		admin to firewall	
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	192.168.1.101	*	VLANBLUE subnets	*	*	none		admin to vlan blue	
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	LAN subnets	*	VLANBLUE subnets	*	*	none		block lan to vlan blue	
<input type="checkbox"/>	✗ 0/900 B	IPv4 *	LAN subnets	*	VLANYELLOW subnets	*	*	none		block lan to vlan yellow	
<input type="checkbox"/>	✗ 0/0 B	IPv4 *	LAN subnets	*	LAN address	*	*	none		block lan to firewall	
<input type="checkbox"/>	✓ 3/10.60 MiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	10.0.2.42	443 (HTTPS)	*	none		NAT wan to www through waf	
<input type="checkbox"/>	✗ 0/3 KiB	IPv6 *	*	*	*	*	*	none		block ipv6	

Rys. 4: Reguły firewall dotyczące komunikacji wewnętrz sieci

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✗ 0/0 B	IPv4 TCP	*	*	This Firewall (self)	80 (HTTP)	*	none		block webconfig	
<input checked="" type="checkbox"/>	✓ 0/3.37 MiB	IPv4 TCP	*	*	10.0.2.42	443 (HTTPS)	*	none		NAT wan to www through waf	
<input type="checkbox"/>	✗ 0/5.95 MiB	IPv4 *	*	*	*	*	*	none		block incoming	
<input type="checkbox"/>	✗ 0/816 B	IPv6 *	*	*	*	*	*	none		block ipv6	

Rys. 5: Reguły firewall dotyczące komunikacji sieci wewnętrznej z siecią zewnętrzną

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 5/360.99 MiB	IPv4 *	*	*	*	*	*	*		none	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	VLANYELLOW subnets	*	192.168.1.101	*	*	none		vlan yellow to admin	
<input type="checkbox"/>	✗ 0/5 KiB	IPv6 *	*	*	*	*	*	none		block ipv6	

Rys. 6: Reguły firewall dotyczące VLAN 200 - oznaczany na żółto

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	10.0.1.40	*	10.0.2.40	1521	*	none		db to www	
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	10.0.1.41	*	LAN subnets	*	*	none		files server to lan	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	VLANBLUE subnets	*	192.168.1.101	*	*	none		vlan blue to admin	
<input type="checkbox"/>	✗ 0/0 B	IPv6 *	*	*	*	*	*	none		block ipv6	

Rys. 7: Reguły firewall dotyczące VLAN 100 - oznaczany na niebiesko

10. Hardening i inne czynności zwiększające bezpieczeństwo systemu

10.1. Logowanie zdarzeń i rotacja logów

Do zarządzania logami zastosowano narzędzie logrotate. Najważniejsze logi będą archiwizowane raz w tygodniu, a te starsze niż 8 tygodni będą usuwane. Dzięki temu zostanie zaoszczędzone miejsce na dysku, jednak w razie potrzeby ostatnie logi będą dostępne i uporządkowane.

```
GNU nano 7.2
# see "man logrotate" for details

# global options do not affect preceding include directives

# rotate log files weekly
/var/log/*.log{
    weekly
    rotate 8
    create
    dateext
    olddir /var/log/oldlogs
    compress
}

/var/log/apt/*.log{
    weekly
    rotate 8
    create
    dateext
    olddir /var/log/apt/oldlogs
    compress
}

/var/log/nginx/*.log{
    weekly
    rotate 8
    create
    dateext
    olddir /var/log/nginx/oldlogs
    compress
}

include /etc/logrotate.d

/var/log/wtmp {
    weekly
    rotate 8
    olddir /var/log/oldlogs
    create 0664 root utmp
    dateext
}
```

Rys. 8: Przykładowa konfiguracja logrotate na serwerze proxy + waf

10.2. Skanowanie systemów operacyjnych przeciw malware

Skanowanie systemów operacyjnych odbywa się za pomocą oprogramowania ClamAV - programu antywirusowego typu open source, przeznaczonego do wykrywania i usuwania wirusów, trojanów, złośliwego oprogramowania oraz innych zagrożeń z systemu Linux.

Skonfigurowano codzienne skanowanie systemu o 4:00, aby móc wykonać pełen skan z usuwaniem zagrożeń w czasie o przewidywanym najmniejszym wykorzystaniu serwerów.

```

root@waf:/var/log# clamscan --version
ClamAV 1.0.3/27156/Tue Jan 16 04:38:07 2024
root@waf:/var/log# sudo systemctl stop clamav-freshclam
root@waf:/var/log# sudo freshclam
Wed Jan 17 04:25:08 2024 -> ClamAV update process started at Wed Jan 17 04:25:08 2024
Wed Jan 17 04:25:08 2024 -> daily.cvd database is up-to-date (version: 27156, sigs: 2050988, f-level: 90, builder: raynman)
Wed Jan 17 04:25:08 2024 -> main.cvd database is up-to-date (version: 62, sigs: 6647427, f-level: 90, builder: sigmgr)
Wed Jan 17 04:25:08 2024 -> bytecode.cvd database is up-to-date (version: 334, sigs: 91, f-level: 90, builder: anvilleg)

```

Rys. 9: Aktualizacja bazy sygnatur ClamAV

10.3. Automatyzacja usług bezpieczeństwa

Jak wspomniano powyżej, usługi tj. logrotate czy ClamAV najlepiej działają, kiedy są uruchamiane regularnie. Aby ułatwić zarządzanie nimi zastosowano usługę cron. Z odpowiednią konfiguracją zaplanowano uruchamianie odpowiednich serwisów w równych odstępach czasowych lub przy każdorazowym uruchomieniu serwera.

```

# 
@reboot /usr/bin/nginx
10 * * * * /usr/bin/logrotate /etc/logrotate.conf
* 4 * * * /usr/bin/clamscan --remove --infected -r /

```

Rys. 10: Przykładowa konfiguracja crontab

10.4. Wyłączenie ipv6

IPv6 mogłyby być pomocne jeśli chodzi o zwiększenie kanałów dostępności. Jednak dla tego systemu informatycznego nie jest przewidywany tak duży ruch sieciowy, a potencjalne słabości wynikające ze złej konfiguracji zabezpieczeń przewyższają potencjalne zalety wynikające z redundancji. Dlatego podjęto decyzję o całkowitym wyłączeniu IPv6.

```

web@www:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:c7:cc:7d brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.0.2.40/24 brd 10.0.2.255 scope global dynamic ens18
        valid_lft 7163sec preferred_lft 7163sec
web@www:~$ 

```

Rys. 11: Ustawienia wyłączające ipv6 w pliku /etc/sysctl.conf

10.5. Bezpieczne hasła użytkowników

Bezpieczne hasło zapobiega nieautoryzowanemu dostępowi do systemu przez osoby nieuprawnione. Stanowi podstawową formę kontroli dostępu. Dlatego ważne jest ustawienie bezpiecznego hasła.

W utworzonym systemie przyjęto politykę haseł przynajmniej 10 znakowych, z co najmniej 1 cyfrą, 1 wielką literą i 1 znakiem specjalnym. Ponadto hasło nie może zawierać w sobie nazwy użytkownika czy nazwy hosta.

```
web@www:~$ passwd
Changing password for web.
Current password:
New password:
Retype new password:
You must choose a longer password.
New password:
Retype new password:
passwd: password updated successfully
web@www:~$
```

Rys. 12: Operacja zmiany hasła

10.6. Uprawnienia sudo

Na serwerach nie wdrażaliśmy uprawnień sudo dla innych użytkowników niż root, ze względu na to, że operacje utrzymywane i konfiguracyjne będą wykonywane tylko przez administratora systemu i nikt inny nie powinien mieć dostępu do wprowadzania zmian.

10.7. Wyłączenie hibernacji

Aby zapobiec hibernacji urządzeń będącymi serwerami, z w związku z tym naglej utracie dostępności usługi, wyłączono hibernację, usypanie oraz wstrzymywanie pracy systemu operacyjnego na hostach-serwerach.

10.8. MAC: AppArmor

Wdrożono Mandatory Access Control w formie modułu AppArmor. Jego konfiguracja jest prostsza i pozwoli na łatwiejsze dodawanie profili w przyszłości przez osoby utrzymujące system. AppArmor pozwala na konfigurowanie bezpieczeństwa za pomocą profili - zarówno gotowych, możliwych do pobrania za pomocą narzędzia *apt-get*, a także samodzielnie utworzonych w języku C.

```
root@waf:/etc/apparmor.d# sudo aa-status
apparmor module is loaded.
12 profiles are loaded.
12 profiles are in enforce mode.
/usr/bin/freshclam
/usr/bin/man
/usr/lib/NetworkManager/nm-dhcp-client.action
/usr/lib/NetworkManager/nm-dhcp-helper
/usr/lib/connman/scripts/dhclient-script
/usr/sbin/clamd
/{,usr/}sbin/dhclient
lsb_release
man_filter
man_groff
nvidia_modprobe
nvidia_modprobe//kmod
0 profiles are in complain mode.
0 profiles are in kill mode.
0 profiles are in unconfined mode.
2 processes have profiles defined.
2 processes are in enforce mode.
/usr/sbin/clamd (532)
/usr/sbin/dhclient (391) /{,usr/}sbin/dhclient
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
0 processes are in mixed mode.
0 processes are in kill mode.
```

Rys. 13: Uruchomiony AppArmor i włączone profile

11. Instrukcje utrzymywania

11.1. Ogólne zalecenia

1. Monitorowanie i zarządzanie konfiguracją:

- Regularnie monitoruj i sprawdzaj konfigurację bazowych usług serwerów, upewniając się, że jest ona zgodna z zaleceniami producenta i spełnia wymagania bezpieczeństwa
- Utrzymuj dokumentację konfiguracji, aby ułatwić zarządzanie i późniejsze zmiany.

2. Regularne aktualizacje i łatki:

- Regularnie aktualizuj i instaluj dostępne łatki bezpieczeństwa, poprawki i aktualizacje dla oprogramowania.
- Śledź powiadomienia o podatnościach bezpieczeństwa związanych z wykorzystywanym oprogramowaniem i natychmiast reaguj na znalezione zagrożenia.

3. Monitoring wydajności:

- Monitoruj i analizuj wydajność serwerów, włączając w to obciążenie procesora, zużycie pamięci, pasmo internetowe, wydajność przekierowań, liczbę i prędkość żądań HTTPS.
- Analizuj logi w celu wykrycia ewentualnych problemów, błędów czy prób ataków.

4. Testy i wdrożenia zmian:

- Przed wprowadzeniem zmian w konfiguracji serwerów, przeprowadź testy w środowisku testowym, aby upewnić się, że zapewniają one oczekiwane rezultaty i nie powodują awarii innego funkcjonalności.
- Utrzymuj odpowiednią dokumentację zmian, tak aby można było śledzić historię i wiedzieć, jakie zmiany wprowadzono oraz z jakiego powodu.

5. Planowanie awaryjne:

- Zaplanuj scenariusze awaryjne i procedury postępowania w przypadku awarii serwerów.
- Przeprowadzaj regularne ćwiczenia i symulacje awarii, aby upewnić się, że personel jest przygotowany do skutecznego i szybkiego reagowania w przypadku problemów.

6. Szkolenia pracowników:

- Zapewnij odpowiednie szkolenia personelu dotyczące zarządzania, konserwacji i bezpieczeństwa systemu.
- Edukuj użytkowników na temat zagrożeń, takich jak phishing i skanowanie malware, i jak ich unikać.

11.2. Serwer WAF + reverse proxy

1. Konfiguracja zasad bezpieczeństwa:

- W razie potrzeby (niewystarczającej lub nadmiernej ochrony przez naxsi) skonfiguruj zasady bezpieczeństwa (w /etc/nginx/nginx.conf) poprzez edycję progów blokowania. Możesz także zastosować LearningMode w celu wychwycenia false positives i dołączenia ich do listy wyjątków.

2. "Quick-fix" na przerwania w działaniu

- Jeśli zaobserwujesz, że nginx może nie działać, zweryfikuj czy jest uruchomiony poprzez próbę odświeżenia za pomocą komendy `nginx -s reload`. Przy poprawnym działaniu nie otrzymasz komunikatu zwrotnego. W przypadku błędu o wyłączonej usłudze, należy uruchomić ją ponownie komendą `nginx`. Wówczas warto zweryfikować, czy nie została uszkodzona konfiguracja cron.

11.3. Serwer webowy

1. Konfiguracja bezpieczeństwa:

- Zapewnij bezpieczną komunikację poprzez stosowanie protokołu HTTPS i skonfigurowanie certyfikatów SSL/TLS. Pamiętaj o weryfikacji ich aktualności.

2. Zarządzanie uwierzytelnianiem i uprawnieniami:

- Regularnie przeglądaj listę kont i uprawnień, aby zarządzać dostępem do systemu.

11.4. Serwer bazodanowy

1. Uprawnienia dostępu:

- Przypisuj uprawnienia nowym użytkownikom zgodnie z zasadą najmniejszych uprawnień, aby zapobiec nieautoryzowanemu dostępowi do danych.
- Ogranicz dostęp do funkcji administracyjnych bazy danych tylko dla niezbędnych użytkowników.

11.5. Serwer plików

1. Monitorowanie zużycia zasobów

- Monitoruj i analizuj wykorzystanie zasobów - w szczególności zużycie przestrzeni dyskowej.
- Wprowadź politykę zarządzania pamięcią masową - limity plików, terminy ich usuwania, reguły dot. inwestycji w zwiększenie zasobu.

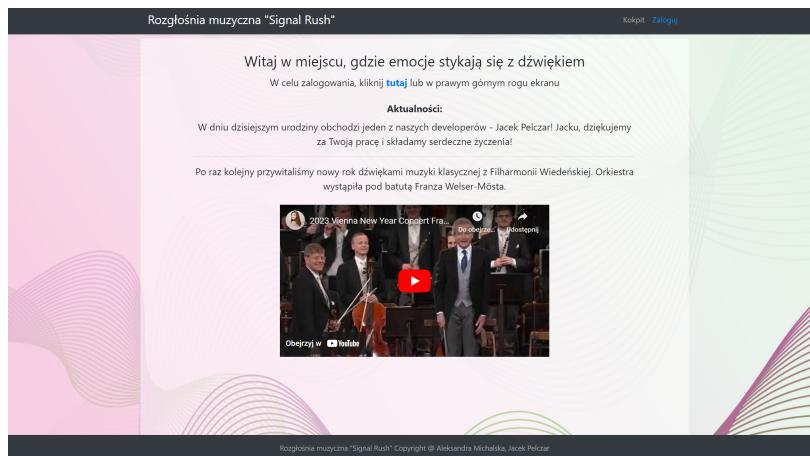
12. Testy i kosztorys projektu

12.1. Testy funkcjonalne

Przeglądanie strony internetowej

Jest to podstawowa funkcjonalność, przeznaczona dla wszystkich użytkowników niezależnie do poziomu uprawnień - dostępna nawet dla gości niezalogowanych.

W ramach testu uruchomiono witrynę rozgłośni za pomocą przeglądarki klienta:



Rys. 14: Wyświetlenie strony głównej

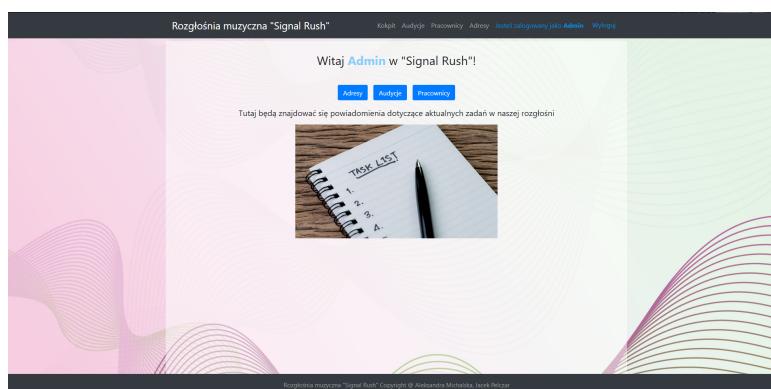
Test przebiegł pozytywnie.

Logowanie i dostępy

Logowanie zapewnia możliwość rozróżnienia uprawnień do zasobów dla różnych grup użytkowników. Przyczynia się to do ochrony wrażliwych zasobów i pozwala na automatyczną kontrolę dostępu.

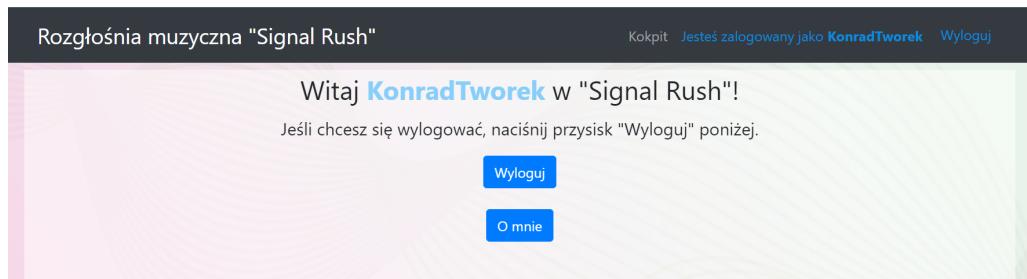
W ramach testu przeprowadzono dwie operacje zalogowania się:

- jako użytkownik z grupy administracyjnej:



Rys. 15: Widok zalogowanego użytkownika administracyjnego

- jako użytkownik z grupy pracowników zwykłych:



Rys. 16: Widok zalogowanego użytkownika - pracownika

Oba testy przebiegły pozytywnie.

Komunikacja z bazą danych

Komunikacja z bazą danych jest niezbędną do prawidłowego działania witryny i funkcjonowania rozgłośni. W bazie przechowywane są informacje o utworach i wykonawcach, o aktualności licencji na emisję, plany i harmonogramy audycji oraz dane pracowników.

W ramach testu zweryfikowano działanie podstawowych operacji na bazie danych:

- SELECT - wyświetlenie swoich danych przez pracownika:

Twoje dane:											
Aby edytować kliknij w rekord.											
Imię	Nazwisko	Data urodzenia	Pesel	Płeć	Data zatrudnienia	Nr konta	Email	Nr telefonu	Adres	Stanowisko	
Konrad	Tworek	12-04-1996	98763647566	K	06-12-2022	97042730434238	konderoTech@Tranzystor	736453625	Warszawa ul.Karolkowa 47	Spiker	

Rys. 17: Wyświetlone dane użytkownika - niejawne zapytanie SELECT

- UPDATE - modyfikacja danych adresowych przez administratora:

Tabela Adresów			
Aby edytować kliknij w rekord.			
ID	Miasto	Ulica	Nr_Lokalu
118	Kraków	Krakowska	4
102	Wąsko	Szeroko	23
101	Sloniów	Orzeszkowa	12
121	Poznań	Pomidorowa	12
122	Gdynia	Jeziorna	44

Tabela Adresów			
Aby edytować kliknij w rekord.			
ID	Miasto	Ulica	Nr_Lokalu
118	Kraków	Krakowska	44
102	Wąsko	Szeroko	23
101	Sloniów	Orzeszkowa	12
121	Poznań	Pomidorowa	12
122	Gdynia	Jeziorna	44
1	Warszawa	Ludwika Warńskiego	19

(a) Przed zmianą

(b) Po zmianie

Rys. 18: Zmodyfikowany nr lokalu w adresie - niejawne zapytanie UPDATE

- DELETE - usunięcie audycji przez administratora:

Tabela Audycji

Aby edytować kliknij w rekord.

ID	Data	Format	Czas_trwania
33	11-01-2023	World Quest with Quebo	00:09:06
42	11-01-2023	World Quest with Quebo	00:09:06
32	12-01-2023	Kołysanki	04:40:00
41	12-01-2023	Kołysanki	04:40:00
34	13-01-2023	Zima z radiem	01:00:00
45	13-01-2023	Zima z radiem	01:00:00
30	23-01-2023	Ranczo	00:25:00
46	23-01-2023	Ranczo	00:25:00
31	29-01-2023	Klasyka	00:30:19
44	29-01-2023	Klasyka	00:30:19
1	13-12-2023	Zima z radiem	00:20:00
43	13-12-2023	Zima z radiem	00:20:00

Tabela Audycji

Aby edytować kliknij w rekord.

ID	Data	Format	Czas_trwania
33	11-01-2023	World Quest with Quebo	00:09:06
42	11-01-2023	World Quest with Quebo	00:09:06
32	12-01-2023	Kołysanki	04:40:00
41	12-01-2023	Kołysanki	04:40:00
34	13-01-2023	Zima z radiem	01:00:00
45	13-01-2023	Zima z radiem	01:00:00
30	23-01-2023	Ranczo	00:25:00
46	23-01-2023	Ranczo	00:25:00
31	29-01-2023	Klasyka	00:30:19
44	29-01-2023	Klasyka	00:30:19
1	13-12-2023	Zima z radiem	00:20:00

(a) Przed usunięciem

(b) Po usunięciu

Rys. 19: Usunięta audycja - niejawne zapytanie DELETE

Powyższe testy przebiegły pomyślnie.

Dostęp do serwera plików

Serwer plików przechowuje i udostępnia pliki dla pracowników korzystających z sieci wewnętrznej systemu. W ramach testów wstawiono pliki na współdzieloną przestrzeń dyskową z poziomu serwera SMB, a następnie zweryfikowano obecność tych plików z poziomu użytkownika w sieci.

```
Terminal - user@fileserver: /srv/samba/office
File Edit View Terminal Tabs Help
user@fileserver:/srv/samba/office$ ls
plan_audycji.txt  wazne_terminy.txt
user@fileserver:/srv/samba/office$ 
```



```
user@admin-notebook: ~ x user@admin-notebook: /media
File Actions Edit View Help
user@admin-notebook: ~ x user@admin-notebook: /media x
user@admin-notebook:~$ ls /media/office/
plan_audycji.txt  wazne_terminy.txt
user@admin-notebook:~$ 
```

(a) Widok z serwera SMB

(b) Widok użytkownika

Rys. 20: Widok na współdzieloną przestrzeń dyskową "office"

Test przebiegł pomyślnie.

12.2. Testy niefunkcjonalne

Wygoda korzystania z witryny

Do testu sporządzono typowe scenariusze użytkowania witryny i interakcji z systemem za jej pośrednictwem, bazując na przeprowadzonych wcześniej testach funkcjonalnych.

Jako użytkownik-gość odwiedzono witrynę i uruchomiono film z platformy youtube.com. Nieauważono nadmiernych niedogodności, minimalna funkcjonalność dla tego profilu powoduje małą powierzchnię błędu.

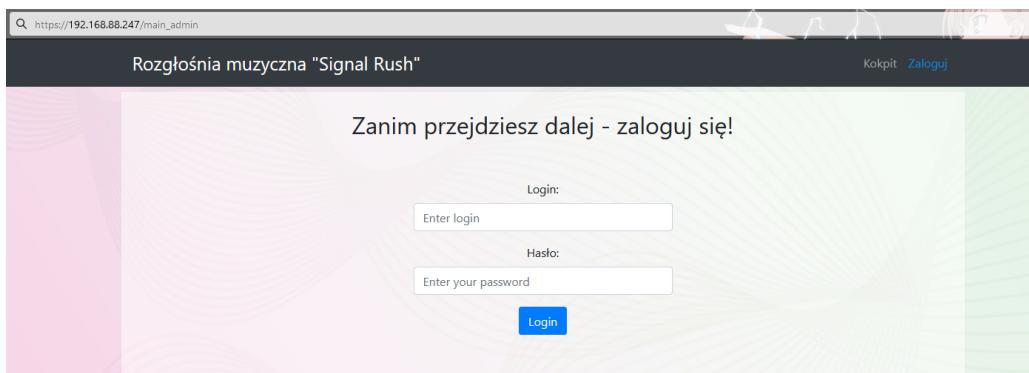
Jako użytkownik-pracownik odczytano własne informacje oraz zmodyfikowano swój nr telefonu. Interfejs okazał się intuicyjny, header umożliwiający przekierowanie do najważniejszych stron ułatwił korzystanie z systemu, a weryfikacja poprawności danych pozwoliła na zapobiegnięcie omyłkowego wpisania złego numeru telefonu.

Jako użytkownik-administrator dokonano modyfikacji utworów w audycji "Zima z radiem". Zadanie było możliwe, użytkownik był nawigowany pomiędzy kolejnymi widokami, co umożliwiło poprawne zrealizowanie zadania. Ilość podstron, przez które trzeba było przejść była odrobinę zbyt duża (5 przekierowań), jednak fakt, że strona kieruje użytkownika do kolejnych widoków w odpowiedniej kolejności mityguje problem.

Testy przebiegły pomyślnie.

Próba zalogowania bez uprawnień

Próba dostania się na stronę /main_admin powoduje przekierowanie na stronę logowania.



Rys. 21: Niedana próba dostania się na stronę /main_admin bez bycia zalogowanym.

Test przebiegł zgodnie z oczekiwaniami.

Próba połączenia z hostem (innym niż WAF) z sieci zewnętrznej

Próba połączenia z hostem administratora z sieci zewnętrznej kończy się niepowodzeniem ze względu na reguły na firewall-u oraz zastosowanie na nim maskarady.

```
C:\Users\Yacek>ping 192.168.0.101

Pinging 192.168.0.101 with 32 bytes of data:
Request timed out.
Reply from 10.1.0.1: Destination net unreachable.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.101:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
C:\Users\Yacek>
```

Rys. 22: Niedana próba wykonania ping do hosta administratora.

Test przebiegł zgodnie z oczekiwaniami.

12.3. Kosztorys

Projektowany system jest przeznaczony do wdrożenia na fizycznym sprzęcie, dlatego też poniżej prezentuje się lista proponowanych urządzeń wraz z kosztorysem i specyfikacją techniczną. Podane ceny są wartościami brutto.

1. Dell PowerEdge R250 - 10 000 PLN
2. Szafa Instalacyjna Rack LANBERG (Flat Pack) V2 - 330 PLN
3. Netgate 4200 pfSense+ Security Gateway - \$ 549 (ok. 2 500 PLN)
4. Dysk HDD WD Purple WD64PURZ - 690 PLN
5. Lenovo ThinkPad X280 (używany) - 800 PLN
6. RackUPS 2000VA/1200W - 1 200 PLN
7. Montaż urządzeń sieciowych oraz instalacja i konfiguracja architektury - 2 000 PLN (20 roboczych godzin)
8. Okablowanie (ethernet, zasilanie) - 200 PLN

Podsumowanie kosztów: 17 720 PLN brutto

12.4. Specyfikacja techniczna

1. Serwer rack Dell PowerEdge R250 - RAM 2x16GB, Intel Xeon E-2314 (4 rdzenie, 4 wątki, 2.80-4.50 GHz, 8 MB cache), dysk HDD SATA 4TB, karta sieciowa Gigabitowa, moc 450W - [Link](#)
2. Szafa Instalacyjna Rack LANBERG (Flat Pack) V2 - ilość jednostek 6U, wymiary 60 cm x 45 cm x 37.4 cm - [Link](#)
3. Netgate 4200 pfSense+ Security Gateway - interfejsy 4 x 2.5 Gbps, moc 13W (idle) - [Link](#)
4. RackUPS 2000VA/1200W - [Link](#)
5. Dysk HDD WD Purple - rozmiar pamięci 6TB, wymiary 3,5", połączenie SATA - [Link](#)
6. Lenovo ThinkPad X280 - i5-8250U, RAM 8GB, 256GB SSD, Windows 10 - [Link](#)

Według ww. listy wdrożenie systemu obejmuje serwer rack, sprzętowy firewall i laptop administratora. Na serwerze rack w wirtualizatorze Proxmox zostaną uruchomione maszyny wirtualne zawierające bazę danych, stronę WWW, WAF oraz serwer plików. Firewall będzie osobnym urządzeniem ze względu na potrzebę zapewnienia odpowiedniej wydajności. Taka propozycja wdrożenia umożliwia rozbudowę systemu w przyszłości, bez potrzeby wymiany całej infrastruktury na nową.

13. Podsumowanie i wnioski końcowe

Celem projektu było stworzenie systemu zarządzania rozgłośnią radiową. W projekcie wykonano infrastrukturę sieci dla małej organizacji, wraz z kosztorysem wdrożenia.

Zaprojektowano stronę WWW połączoną z bazą danych, która umożliwia zarządzanie pracą rozgłośni oraz serwer plików w celu udostępniania zasobów w sieci organizacji. Aby zabezpieczyć się przed potencjalnymi złośliwymi działaniami z sieci zewnętrznej, umieszczono także Reverse Proxy + WAF, który chroni dostęp do strony WWW.

Zastosowano technologie takie jak: Samba, Oracle Database XE, Proxmox (wirtualizacja), pfSense, Nginx, Tomcat, Java, Apparmor, ClamAV, VLAN.

Projekt wdrożono jako Proof Of Concept. Oznacza to, że nie wszystkie elementy zostały wdrożone tak jak przewidywał projekt. Hosty przewidziane w projekcie sieci oraz połączenia sieciowe zostały wdrożone jako maszyny wirtualne w wirtualizatorze Proxmox. Proxmox został zainstalowany na terminalu DELL Optiplex 7050. Ten sposób wdrożenia z założenia miał być dowodem na działanie systemu, natomiast zaleca się wdrożenie na zasobach fizycznych opisanych w rozdziale *12.3. Kosztorys*.

Koszt wdrożenia projektu według przygotowanej propozycji wynosi ok. 17 720 PLN brutto (stan na dzień 17.01.2024).

Przygotowanie projektu zajęło 3 miesiące. Na początku wyznaczono założenia projektowe istępny plan realizacji, natomiast główna realizacja projektu rozpoczęła się na przełomie grudnia i stycznia.

Projekt w przyszłości można rozbudować o: komputery pracowników (osobny vlan, dodatkowy switch), zapewnienie redundancji usług kluczowych, centralne kolekcjonowanie i analizę logów np. w rozwiązaniu SIEM, system analizy ruchu sieciowego i wykrywania zagrożeń typu np. NIDS, system balansowania obciążenia sieci.