Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

przedmiot
BEKOM

Projekt 2

Mateusz Plichta, Karol Żelazowski, Jakub Szweda, Maja Berej
Numer albumu 324939, 324953, 324944, 324903

prowadzący
dr inż. Jędrzej Bieniasz

WARSZAWA 14 lutego 2025

Spis treści

1. Wstęp

Celem projektu jest stworzenie architektury bezpiecznej sieci firmowej.

2. Architektura sieci

Architektura naszej sieci przedstawiona jest na rysunku 2.1. Posiada ona 5 VLANów oraz jest ona połączona z Internetem, chmurą i drugim biurem.

2.1. VLAN 1

W vlanie żółtym znajdują się takie elementy architektury jak: server VPN, jump server, firewall, router z przyłączoną instancją NIDS, a oprócz tego server DNS z log collectorem. W naszym zamyśle Vlan żółty to jedyny bezpośrednio dołączony do internetu Vlan. Z racji tego, że jest on na styku komunikacji z zewnątrz to znajduje się w nim server VPN, jump server oraz publicznie dostępny serwer DNS. Server VPN został stworzony z myślą o pracownikach zdalnych, chmurze oraz drugim biurze. Jump server daje dostęp do serwerów w VLANie niebieskim oraz zielonym. Z racji tego, że firewall stoi już przed Cloudem oraz drugim biurem to nie stawialiśmy kolejnego przed VPN oraz jumpserver. Stoi on natomiast przed ruchem pochodzącym z internetu, a jednocześnie ruchem przechodzącym z VPN server lub jumpserver w głąb naszej infrastruktury. Do routera dołączona została instancja NIDS, która kopiuje ruch sieciowy z routera, nie spowalniając jego pracy. Router ten kieruje ruchem pomiędzy VLANami żółtym, niebieskim oraz zielonym. W tym VLANie znajduje się także publiczny serwer DNS odpowiedzialny za całą strefę zdemilitaryzowaną oraz dołączony log collector, który jest bardzo ważny w przypadku wykrywania komunikacji C2 za pomocą protokołu DNS.

2.2. VLAN 2

VLAN 2 stanowi wydzielony segment w architekturze naszej sieci, który jest dostępny publicznie, jednak dostęp do niego możliwy jest tylko pośrednio przez VLAN 1, który działa jako warstwa zabezpieczająca. Segment ten, umiejscowiony za firewall'em, zawiera kluczowe serwery, takie jak serwer WWW, który jest dodatkowo chroniony przez system WAF (Web Application Firewall), oraz serwer FTP. WAF na serwerze WWW monitoruje i filtruje ruch HTTP/HTTPS, zapobiegając atakom takim jak SQL injection, XSS (Cross-Site Scripting) czy inne formy ataków skierowanych na aplikacje webowe. Dodatkowo, oba serwery są wyposażone w log collectory, które zbierają logi systemowe oraz aplikacyjne, a następnie przesyłają je do centralnego systemu zarządzania logami, umożliwiając monitorowanie, analizę oraz wykrywanie anomalii i potencjalnych zagrożeń. Dzięki temu procesowi możliwe jest bieżące śledzenie stanu bezpieczeństwa serwerów oraz szybka reakcja na pojawiające się incydenty. Firewall między VLAN 1 a VLAN 2 pełni rolę kontrolera

dostępu, blokując nieautoryzowane próby komunikacji i zapewniając, że tylko określone urządzenia i usługi mogą uzyskać dostęp do zasobów w VLAN 2.

2.3. DMZ

Strefa zdemilitaryzowana jest obszarem naszej sieci, który znajduje się na styku naszej sieci wewnętrznej oraz sieci zewnętrznej. Znajdują się w nim dwa VLANy - VLAN żóły i VLAN niebieski. VLANy te są częścią strefy DMZ z uwagi na to, że znajdują się w nim urządzenia sieciowe, do których możliwe jest połączenie się z Internetu, takie jak publiczny serwer DNS, serwer WWW oraz FTP.

2.4. VLAN 3

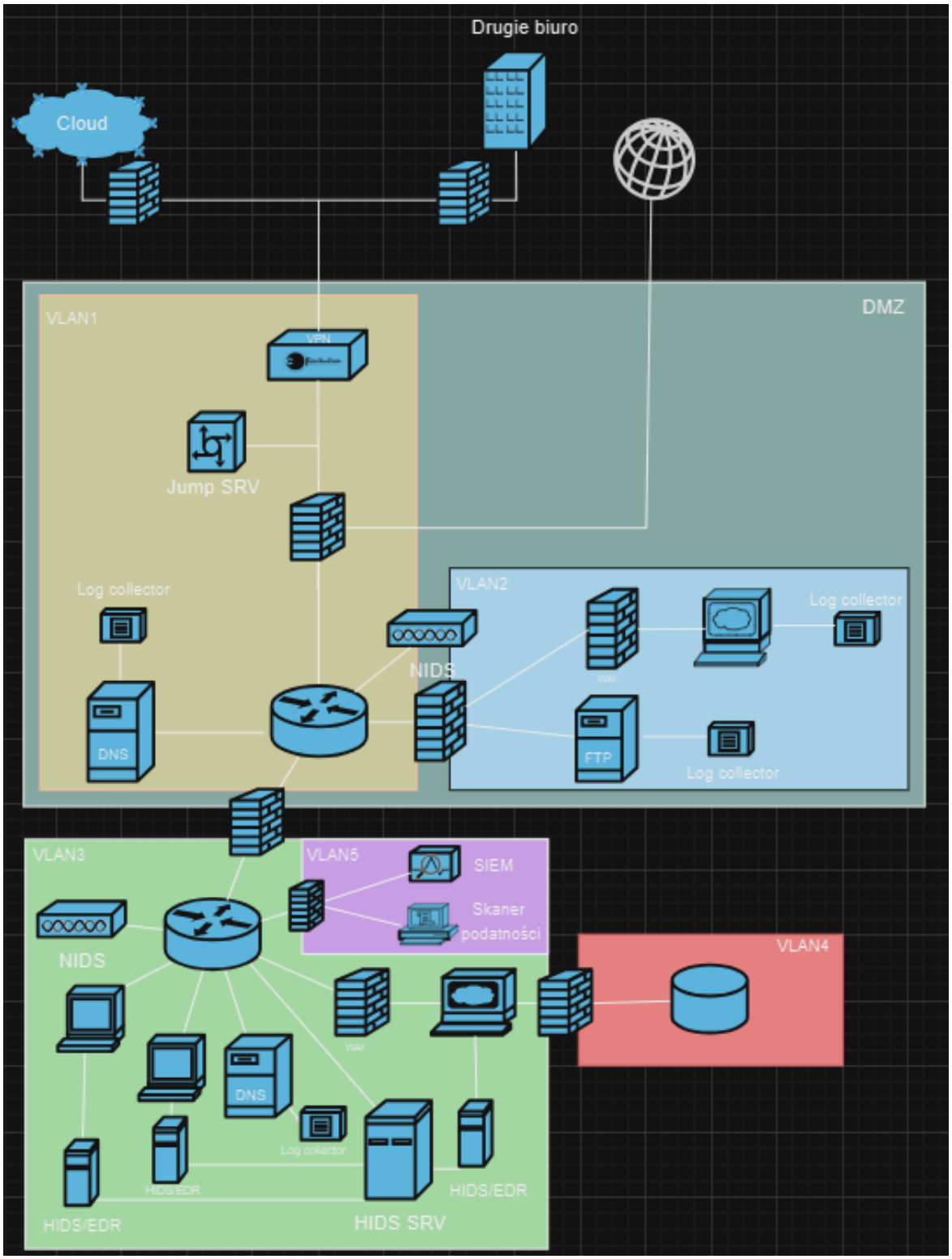
Wydzielony segment infrastruktury sieciowej, który obsługuje główne operacje codzienne klienta, obejmuje terminale pracowników oraz serwer WWW (nginx), na którym zainstalowane są agenty HIDS (Host Intrusion Detection System). Te agenty są połączone z jednym wspólnym serwerem HIDS, również znajdującym się w tym samym VLAN-ie. Na każdym z hostów znajdują się także log collectory (systemy zbierające logi), które zbierają dane dla systemu SIEM (Security Information and Event Management), umieszczonego w VLAN5 (w zależności od systemu użytkownika, mogą to być np. syslog lub sysmon). W tym samym VLAN-ie znajduje się także serwer DNS (Domain Name System), który obsługuje zapytania o serwery WWW w sieci wewnętrznej; na tym serwerze również zainstalowany jest log collector. Wszystkie urządzenia w tej konfiguracji są połączone za pomocą routera, na którym działa agent NIDS (Network Intrusion Detection System). Serwer WWW jest dodatkowo zabezpieczony systemem WAF (Web Application Firewall).

2.5. VLAN 4

Wydzielony segment infrastruktury sieciowej, zaprojektowany specjalnie do przechowywania wrażliwych danych klientów oraz innych krytycznych informacji niezbędnych do działania kluczowych systemów, jest całkowicie odizolowany od reszty sieci. Baza danych w tym segmencie zawiera informacje o charakterze wrażliwym, takie jak dane osobowe, dane finansowe czy dane transakcyjne, które wymagają szczególnej ochrony. Segment jest zabezpieczony przed nieautoryzowanym dostępem poprzez zastosowanie dodatkowego firewall'a, który kontroluje wszelką komunikację między tym segmentem a VLAN3. Firewall pełni funkcję ochrony przed potencjalnymi atakami z innych części sieci oraz zapobiega nieuprawnionemu dostępowi do danych. Wszystkie połączenia z tym segmentem są ściśle kontrolowane i monitorowane, aby zapewnić zgodność z politykami bezpieczeństwa oraz przepisami dotyczącymi ochrony danych, takimi jak RODO czy PCI-DSS.

2.6. VLAN 5

W segmencie sieci, w którym znajdują się system SIEM (Security Information and Event Management) oraz Skaner Podatności, dostęp do tych systemów jest ściśle kontrolowany i możliwy jedynie dla uprawnionych użytkowników, w tym administratorów odpowiedzialnych za zarządzanie bezpieczeństwem. Celem tej konfiguracji jest zapewnienie stałego poziomu ochrony w firmie poprzez monitorowanie, analizowanie i reagowanie na wszelkie potencjalne zagrożenia, a także identyfikowanie słabych punktów w systemach informatycznych. System SIEM gromadzi logi z różnych źródeł w sieci, umożliwiając wykrywanie anomalii oraz podejrzanych działań w czasie rzeczywistym, natomiast Skaner Podatności służy do regularnego sprawdzania infrastruktury pod kątem znanych luk bezpieczeństwa i weryfikowania poziomu zabezpieczeń. Dodatkowo, segment ten jest zabezpieczony firewall'em, który kontroluje komunikację z VLAN3, zapobiegając nieautoryzowanemu dostępowi i minimalizując ryzyko ataków z innych części sieci.



Rysunek 2.1. Architektura sieci

3. Macierz komunikacji

Poniżej znajduje się macierz komunikacji odzwierciedlająca reguły obowiązujące w naszej sieci. Elementy w wierszach inicjują komunikację z elementami w kolumnach. Ustaliliśmy, że VLANy żółty, niebieski, zielony i czerwony nie mogę inicjować komunikacji z VLANem fioletowym, natomiast VLAN fioletowy może inicjować połączenie z pozostałymi VLANami. Oprócz tego, pracownik zdalny może inicjować połączenie z VLANami niebieskim i zielonym, ale z tych VLANów nie jest możliwe nawiązanie połączenia ze zdalnym pracownikiem. Dodatkowo, czerwony VLAN nie może inicjować komunikacji z VLANem zielonym, natomiast zielony może inicjować komunikację z czerwonym.

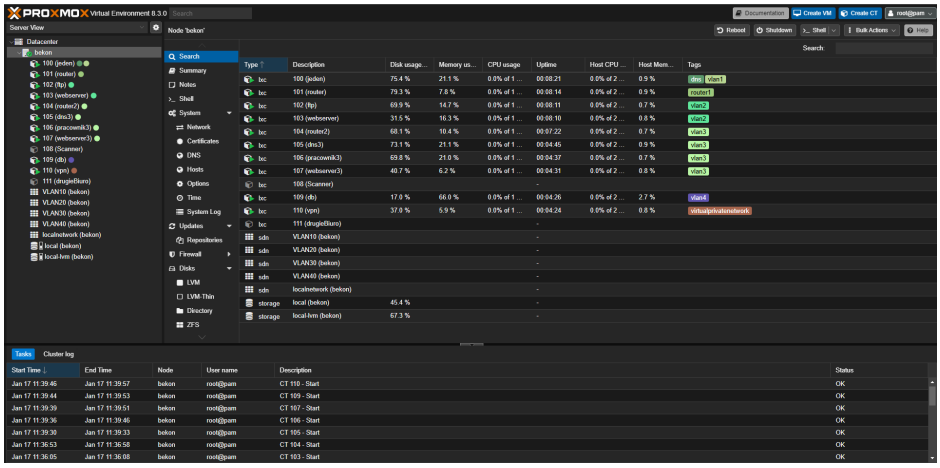
X	Pracownik zdalny	Cloud	Drugie biuro	ŻÓŁTY	NIEBIESKI	ZIELONY	CZERWONY	FIOLETOWY
Pracownik zdalny	X	NIE	NIE	TAK	TAK	TAK	NIE	NIE
Cloud	NIE	X	NIE	TAK	NIE	NIE	NIE	NIE
Drugie biuro	NIE	NIE	X	TAK	NIE	NIE	NIE	NIE
ŻÓŁTY	TAK	TAK	TAK	X	TAK	NIE	NIE	NIE
NIEBIESKI	NIE	NIE	NIE	TAK	X	TAK	NIE	NIE
ZIELONY	NIE	NIE	NIE	NIE	TAK	X	TAK	NIE
CZERWONY	NIE	NIE	NIE	NIE	NIE	NIE	X	NIE
FIOLETOWY	NIE	NIE	NIE	TAK	TAK	TAK	TAK	X

Tabela 3.1. Macierz komunikacji

4. Wdrożenie

4.1. Proxmoxx

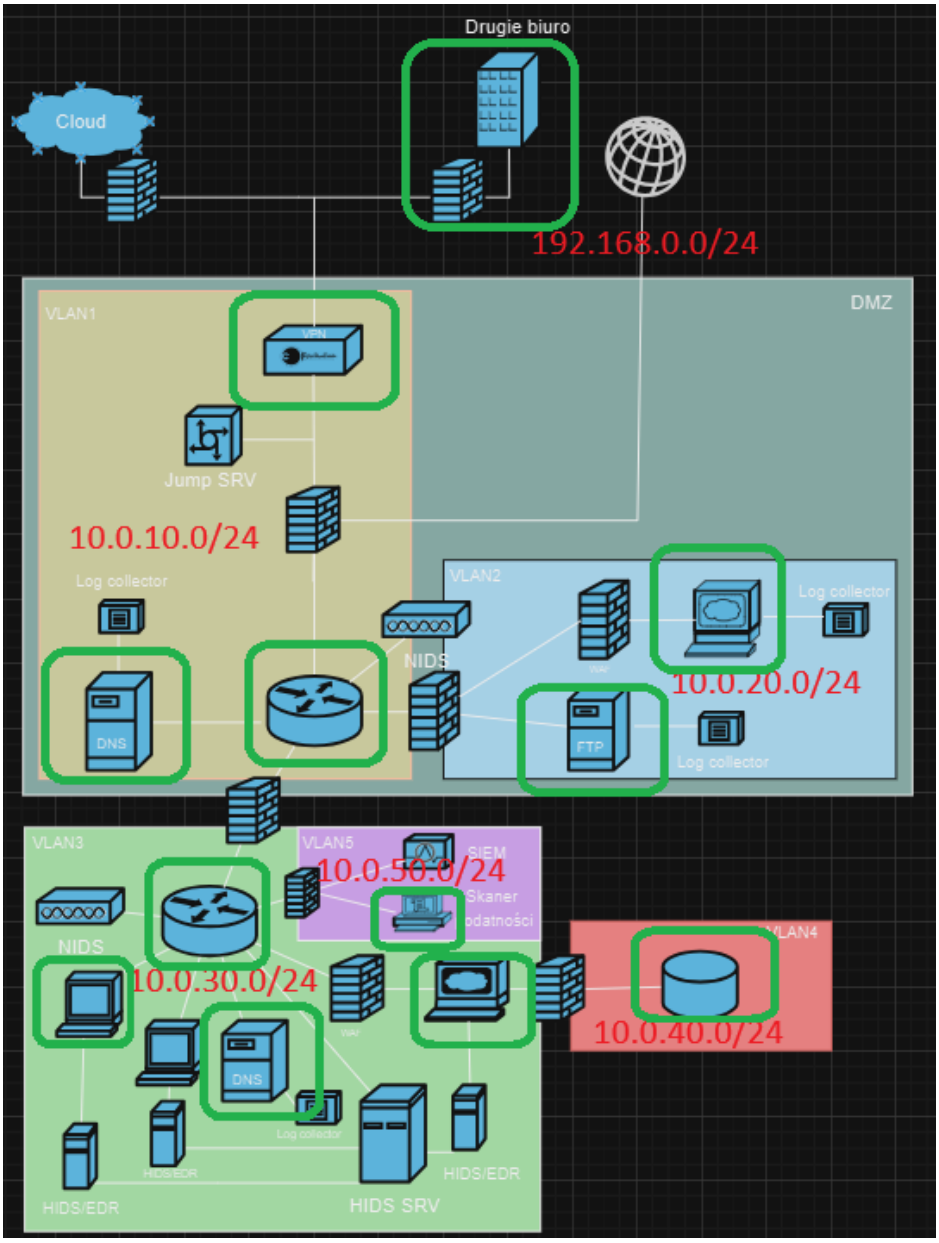
W celu przeprowadzenia symulacji sieci firmowej zdecydowaliśmy się na wykorzystanie platformy Proxmox Virtual Environment (Proxmox VE). Jest to otwartoźródłowe rozwiązanie do zarządzania wirtualizacją, które oferuje szerokie możliwości tworzenia, zarządzania i monitorowania wirtualnych maszyn oraz kontenerów. Dzięki Proxmox możemy w łatwy sposób odtworzyć złożoną infrastrukturę sieciową, co pozwala na realistyczne testowanie konfiguracji, symulowanie ruchu sieciowego oraz sprawdzanie bezpieczeństwa systemów. Ze względu na ograniczone zasoby zdecydowaliśmy się użyc LXC czyli kontenerów, co umożliwiło nam dokładne odwzorowanie środowiska sieciowego przy minimalnym koszcie i wysokim poziomie kontroli nad konfiguracją systemu.



Rysunek 4.1. Panel Proxmox

wprowadzona infrastruktura

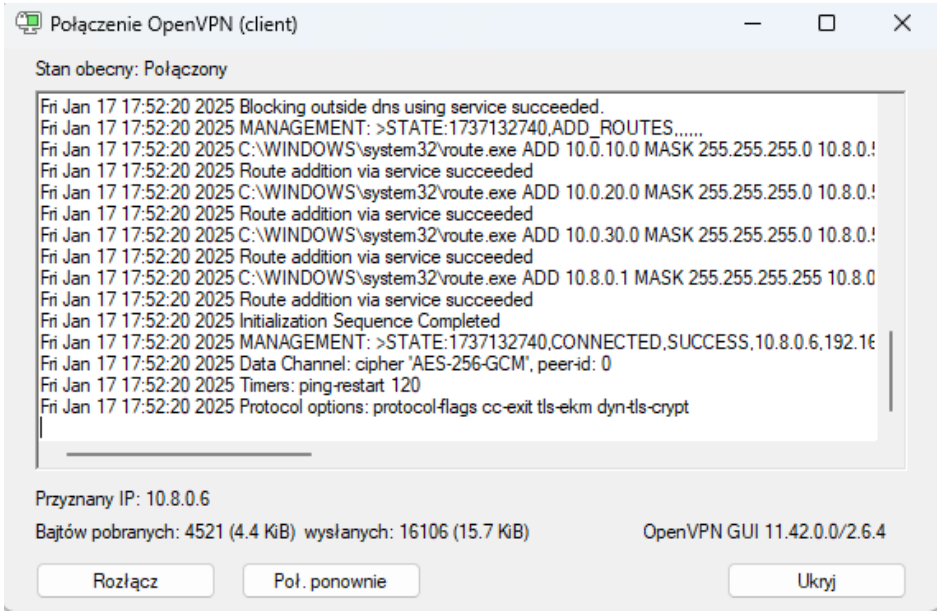
Poniżej przedstawiamy zdjęcie z elementami, które wprowadziliśmy w życie. Elementy, oznaczone zielonymi prostokątami to elementy sieci, które są kontenerami. Oprócz tego dodaliśmy adresacje sieci na rysunku, która w czytelny sposób przestawia segmentację sieci. Firewalle, nie są oznaczone na rysunku ze względu na to, że część z nich wprowadzona jest z poziomu proxmoxx wykorzystując SDN. Oprócz tego dodatkowe zasady wprowadzone na hostach wykorzystując UFW.



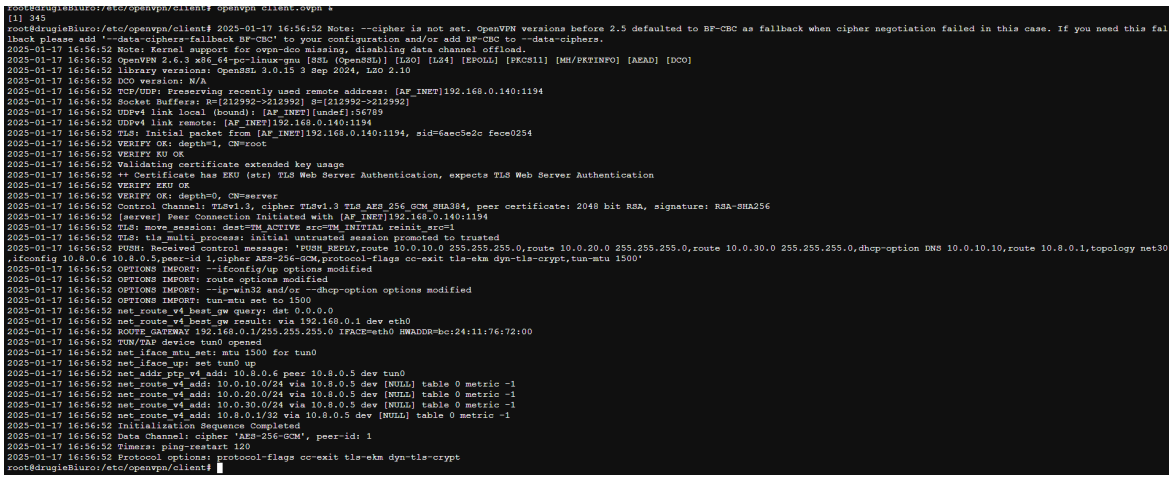
Rysunek 4.2. realizacja w proxmoxx

4.2. Pracownik zdalny

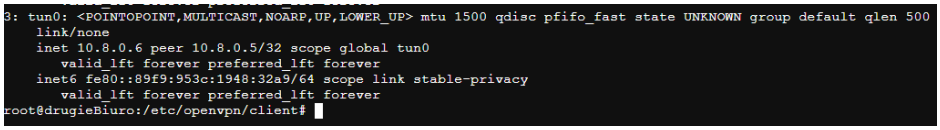
W insfrastrukturze zastosowaliśmy mechanizm VPN oferowany przez usługę openvpn. W tym celu skonfigurowany został serwer openvpn umieszczony w segmencie żółtym (VLAN1). Oraz utworzyliśmy pliki konfiguracyjne dla klientów danego serwera co umożliwiło nam połączenie się do serwera web umieszczonego w infrastrukturze.



Rysunek 4.3. Połączenie się używając openvpvn z komputera lokalnego



Rysunek 4.4. logi serwera vpn

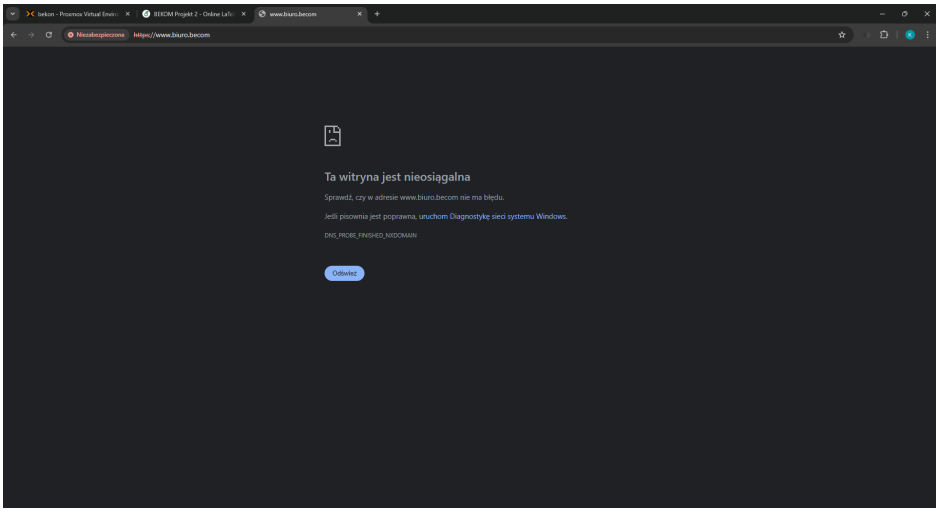


Rysunek 4.5. otrzymana adresacja IP dla podłączonego urządzenia

```
C:\Users\Admin>ping www.biuro.becom

Pinging www.biuro.becom [10.0.20.20] with 32 bytes of data:
Reply from 10.0.20.20: bytes=32 time<1ms TTL=62
Reply from 10.0.20.20: bytes=32 time<1ms TTL=62
Reply from 10.0.20.20: bytes=32 time<1ms TTL=62
Reply from 10.0.20.20: bytes=32 time<1ms TTL=62

Ping statistics for 10.0.20.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Rysunek 4.6. Próba łączności klienta z serwerem infrastruktury wewnętrznej

Rysunek 4.7. Dostęp do strony przy użyciu VPN



Rysunek 4.8. Brak dostępu do strony bez podłączenia z VPNem

4.3. Segment Żółty

W skład tego segmentu wchodzi skonfigurowany serwer ovpn umożliwiający dostęp zdalny jak również serwer DNS umożliwiający połączenie do zasobów współdzielonych między biurami (serwer WWW, FTP). Segment ten ma zatem dostęp do segmentu niebieskiego.

```
root@jeden:~# ping 10.0.20.20
PING 10.0.20.20 (10.0.20.20) 56(84) bytes of data.
64 bytes from 10.0.20.20: icmp_seq=1 ttl=63 time=0.233 ms
64 bytes from 10.0.20.20: icmp_seq=2 ttl=63 time=0.072 ms
64 bytes from 10.0.20.20: icmp_seq=3 ttl=63 time=0.088 ms
64 bytes from 10.0.20.20: icmp_seq=4 ttl=63 time=0.077 ms
64 bytes from 10.0.20.20: icmp_seq=5 ttl=63 time=0.085 ms
^C
--- 10.0.20.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4107ms
rtt min/avg/max/mdev = 0.072/0.111/0.233/0.061 ms
root@jeden:~#
```

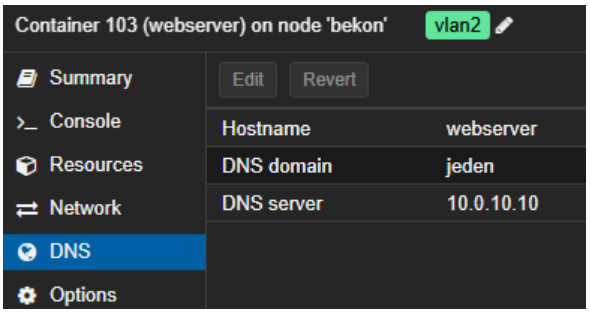
Rysunek 4.9. Test łączności między żółtym i niebieskim serwerem

4.3.1. serwer DNS

W ramach serwera DNS Bind9 dodane zostały dwie domeny infranetowe ‘www.biuro.becom, ftp.biuro.becom‘ umożliwiające wykorzystywanie tych domen do dostępu do zasobów współdzielonych.

```
acl "trusted" {
        192.168.0.0/16;
        localhost;
        localhost;
};

options {
        directory "/var/cache/bind";

        recursion yes;
        allow-query { any; };

        forwarders {
                8.8.8.8;
                8.8.4.4;
        };

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        listen-on { any; };
        listen-on-v6 { any; };
#       tls-cert-file "/etc/bind/strona.crt";
#       tls-key-file "/etc/bind/strona.key";
#       tls-ca-file "/etc/ssl/certs/ca-certificates.crt";

};
```

Rysunek 4.10. Konfiguracja DNS 1



Rysunek 4.13. wprowadzenie domyślnego serwera dns

```
  GNU nano 7.2
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";


zone "biuro.becom" {
        type master;
        file "/etc/bind/db.biuro.becom";
};

zone "20.0.10.in-addr.arpa" {
        type master;
        file "/etc/bind/db.10.0.20";
};
```

Rysunek 4.11. Konfiguracja DNS 2

```
$TTL    604800
@       IN      SOA     ns.biuro.becom. admin.biuro.becom. (
                        1         ; Serial
                        604800    ; Refresh
                        86400     ; Retry
                        2419200   ; Expire
                        604800 )  ; Negative Cache TTL

@       IN      NS      ns.biuro.becom.
ns      IN      A       10.0.20.20
www     IN      A       10.0.20.20
ftp     IN      A       10.0.20.10
```

Rysunek 4.12. Konfiguracja DNS 3

Dla urządzeń w vlan 10 oraz vlan 20 domyślnym serwerem dns jest serwer dns1o nazwie "jeden". Możliwa jest taka konfiguracja z poziomu proxmoxxa.

4.4. Segment Niebieski

Na tym segmencie infrastruktury umieszczone zostały zasoby współdzielone czyli serwer www oraz ftp.

4.4.1. Serwer www

Utworzyliśmy stronę internetową www.biuro.becom przy użyciu usługi nginx a także zabezpieczyliśmy z wykorzystanie WAFa modsecurity przed atakamiem cross site scripting.

Rysunek 4.14. www.biuro.becom

Poniżej znajduje się konfiguracja naszego serwera NGINX. Znajduje się w nim konfiguracja SSL i reverse-proxy.

```
  GNU nano 7.2
user www-data;
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;
include modules-enabled/*.conf;

events {
    worker_connections 1024;
}
http{

server {
    listen 80;
    server_name biuro.becom;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name biuro.becom;



    modsecurity on;
    modsecurity_rules_file /etc/nginx/modsecurity.conf;

    ssl_certificate /etc/ssl/strona/cert.pem;
    ssl_certificate_key /etc/ssl/strona/privkey.pem;
    ssl_dhparam /etc/ssl/strona/dhparam.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:AES128-GCM-SHA256';
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1d;
    ssl_stapling on;
    ssl_stapling_verify on;
#    ssl_trusted_certificate /etc/ssl/strona/myCA.crt;
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
}
```

Rysunek 4.15. Konfiguracja nginx.conf

Poniżej znajduje się plik z konfiguracją ModSecurity.

```
  GNU nano 7.2
# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine DetectionOnly
SecRuleEngine On


SecDebugLog /var/log/modsec_debug.log
SecDebugLogLevel 9

# -- Request body handling ---------------------------------------------------

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On

SecAuditLog /var/log/nginx/modsec.log

#SecRule ARGS "@rx .*?<[^>]+>.*?<[^>]+>.*? " \
#        "id:100000001,phase:2,deny,log,status:403,msg:'Wykryto atak XSS'"

SecRule ARGS "@rx <.*?>" \
    "id:1002,phase:2,deny,log,status:403,msg:'XSS attack detected'"
```

Rysunek 4.16. Konfiguracja modsecurity.conf

4.4.2. Serwer ftp

W tym segmencie postawiliśmy również kontener, w którym zaimplementowaliśmy serwer ftp udostępniający publicznie pliki. Poniżej znajdują się zdjęcia przedstawiające fragmenty pliku konfiguracyjnego serwera ftp.

```
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
```

Rysunek 4.17. Plik konfiguracyjny ftp 1

```
chroot_local_user=YES
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
#chroot_local_user=YES
chroot_list_enable=NO
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# Customization
#
# Some of vsftpd's settings don't fit the filesystem layout by
# default.
#
# This option should be the name of a directory which is empty.  Also, the
# directory should not be writable by the ftp user. This directory is used
# as a secure chroot() jail at times vsftpd does not require filesystem
# access.
secure_chroot_dir=/var/run/vsftpd/empty
#
# This string is the name of the PAM service vsftpd will use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO
```

Rysunek 4.18. Plik konfiguracyjny ftp 2

4.5. Segment Zielony

Jest to główna część naszego infranetu, naszym założeniem jest wykorzystywanie tego segmentu do 'daily operations' naszej firmy. Znajduje się w nim:

- przykład stanowiska użytkownika
- wewnętrzny serwer DNS
- przykład wewnętrznej usługi www

4.5.1. Serwer www

Podobnie jak w przypadku segmentu niebieskiego utworzyliśmy stronę internetową z wykorzystaniem NGINX i ModSecurity z dodatkową zasadą wykrywającą próby ataku SQL Injection. Poniżej znajduje się konfiguracja pliku nginx.conf.

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;
include modules-enabled/*.conf;

events {
    worker_connections 1024;
}
http{

server {
    listen 80;
    server_name biuro.becom;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name biuro.becom;



    modsecurity on;
    modsecurity_rules_file /etc/nginx/modsecurity.conf;

    ssl_certificate /etc/nginx/ssl/cert.pem;
    ssl_certificate_key /etc/nginx/ssl/privkey.pem;
    ssl_dhparam /etc/nginx/ssl/dhparam.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:AES128-GCM-SHA256';
    ssl_prefer_server_ciphers off;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1d;
    ssl_stapling on;
    ssl_stapling_verify on;
#    ssl_trusted_certificate /etc/ssl/strona/myCA.crt;
    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
}
```

Rysunek 4.19. Konfiguracja pliku nginx.conf

Na poniższym zdjęciu znajduje się fragment pliku modsecurity.conf. Reguły w nim zdefiniowane chronią nas przez atakiem SQL Injection.

```
# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On
SecDebugLog /var/log/modsec_debug.log
SecDebugLogLevel 9

# -- Request body handling ---------------------------------------------------

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On
SecAuditLog /var/log/nginx/modsec.log

SecRule REQUEST_URI "@rx %27%20OR%20%271%27%3D%271" \
    "id:100000006,phase:2,log,deny,t:none,msg:'Wykryto SQL Injection'"


SecRule ARGS|ARGS_NAMES|REQUEST_HEADERS|REQUEST_BODY|REQUEST_URI "@rx ' OR '1'='1" \
    "id:100000007,phase:2,log,deny,msg:'Wykryto SQL Injection'"
```

Rysunek 4.20. Konfiguracja pliku modsecurity.conf

4.5.2. Wewnętrzny serwer DNS

Uruchomiliśmy również wewnętrzny serwer DNS, ponieważ chcemy aby nasi pracownicy nie musieli zapamiętywać adresów IP używanych przez nich na co dzień narzędzi. Poniżej znajdują się fragmenty plików konfiguracyjnych serwera DNS Bind9.

```
options {
        directory "/var/cache/bind";
        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        recursion yes;
        allow-recursion { any; };
        allow-query { any; };

        forwarders {
                10.0.10.10;
        };

        dnssec-validation no;

        listen-on { any; };
        listen-on-v6 { any; };
};
```

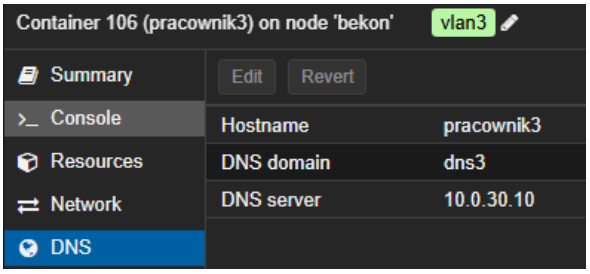
Rysunek 4.21. Konfguracja DNS 1

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "baza.becom" {
        type master;
        file "/etc/bind/db.baza.becom";
};

zone "biuro.becom" {
        type forward;
        forward only;
        forwarders {
                10.0.10.10;
        };
};
```

Rysunek 4.22. Konfguracja DNS 2

```
$TTL    604800
@       IN      SOA     ns.baza.becom. admin.baza.becom. (
                        1         ; Serial
                        604800    ; Refresh
                        86400     ; Retry
                        2419200   ; Expire
                        604800 )  ; Negative Cache TTL

@       IN      NS      ns.biuro.becom.
ns      IN      A       10.0.30.30
www     IN      A       10.0.30.30
```

Rysunek 4.23. Konfguracja DNS 3



Rysunek 4.24. ustawienie domyślnego serwera dns

Dla urządzeń spoza vlan 10 oraz vlan 20 domyślnym serwerem dns jest serwer dns3 o nazwie "dns3". Możliwa jest taka konfiguracja z poziomu proxmoxxa, którą widać powyżej.

4.6. Segment czerwony

W tym segmencie znajduje się wyłącznie baza danych zawierająca wrażliwe informacje o pracownikach. Ze względu na bezpieczeństwo jest ona odizolowana od pozostałych

segmentów. Dopuszczamy jedynie możliwość pobierania z niej danych za pośrednictwem serwera WWW, który je wykorzystuje do swojego działania.

4.6.1. Baza danych

Baza danych została skonfigurowana w taki sposób aby, szyfrować dane w ruchu. Wygenerowaliśmy certyfikat oraz klucz i ścieżki do nich podaliśmy w pliku konfiguracyjnym bazy danych. Poniżej znajduje się dowód poprawnego skonfigurowania SSL w bazie danych w postaci wyświetlenia właściwości szyfrowania komunikacji w bazie.

```
MariaDB [(none)]> SHOW STATUS LIKE 'Ssl_cipher';
+---------------+------------------------+
| Variable_name | Value                  |
+---------------+------------------------+
| Ssl_cipher    | TLS_AES_256_GCM_SHA384 |
+---------------+------------------------+
1 row in set (0.061 sec)

MariaDB [(none)]>
```

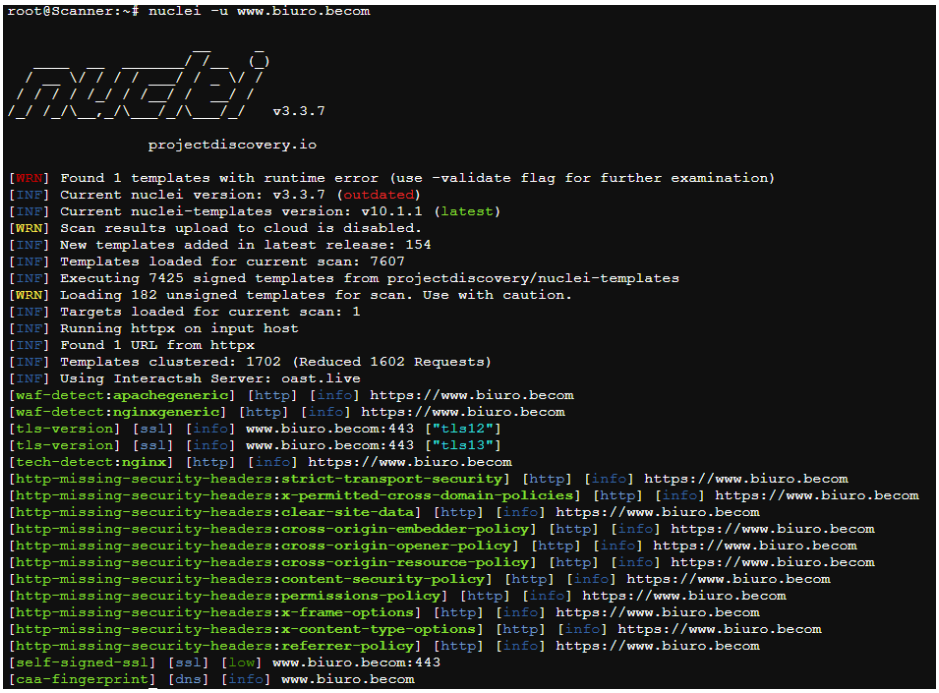
Rysunek 4.25. Dowód skonfigurowania szyfrowania w bazie

4.7. Segment Fioletowy

Segment fioletowy jest odseparowaną, chronioną częścią sieci. Dostęp do VLANa fioletowego nie jest możliwy z żadnego VLANa, ale host w VLANie fioletowym ma dostęp do każdego innego VLANa ze względu na specyfikę hosta znajdującego się w VLANie fioletowym, czyli skanera.

4.7.1. Skaner

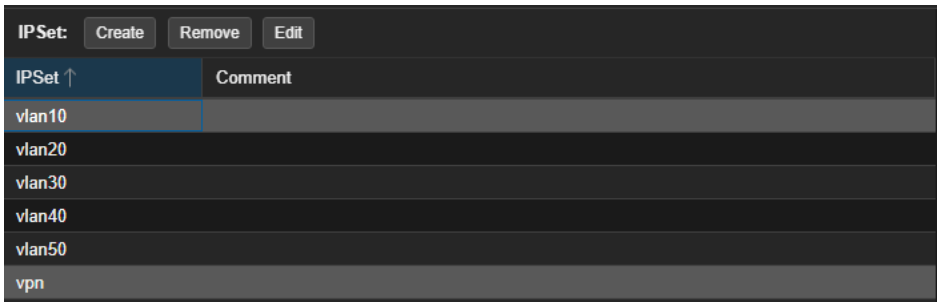
Skanerem, który zaimplementowaliśmy jest skaner Nuclei. Jest to skaner służący do skanowania serwerów webowych i innych usług sieciowych pod kątem podatności bezpieczeństwa. Poniżej zanjduje się wynik skanu jednego z naszych serwerów webowych.



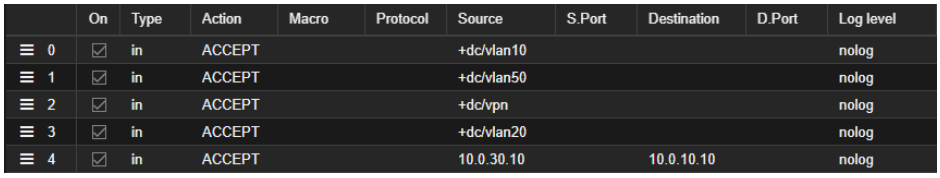
Rysunek 4.26. Wynik skanu skanerem Nuclei

4.8. Firewall

W celu zapewnienia odpowiedniego poziomu bezpieczeństwa i kontroli ruchu sieciowego w symulowanej infrastrukturze, wprowadziliśmy segmentację sieci przy użyciu narzędzi dostępnych w Proxmox Virtual Environment. Kluczowymi elementami tej segmentacji były ipsety oraz security groupy, które umożliwiły precyzyjne zarządzanie ruchem między różnymi segmentami sieci. Dzięki wykorzystaniu IP setów byliśmy w stanie grupować adresy IP na podstawie różnych kryteriów, takich jak lokalizacja, funkcjonalność lub poziom dostępu. To rozwiązanie znacząco uprościło konfigurację reguł zapory (firewall) w Proxmox, ponieważ zamiast definiować osobne reguły dla każdego adresu, mogliśmy stosować reguły do całych grup adresów. W naszym przypadku grupa adresów była bezpośrednio skorelowana z naszymi vlanami. Niestety nie udało się wprowadzić zasad firewallowych stricte na vlan tagach. Natomiast rozwiązanie, które wprowadziliśmy w proxmoxx jest bardzo wygodne. Poniżej widać utworzone ipsety. Adresacje przydzielone są zgodnie ze schematem 10.0.<vlan_number>.0/24

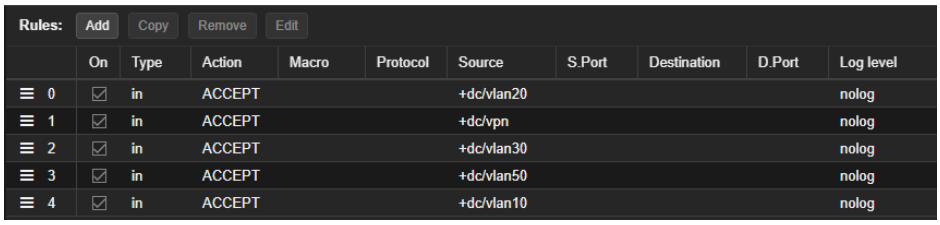


Rysunek 4.27. IP set



	On	Type	Action	Macro	Protocol	Source	S.Port	Destination	D.Port	Log level
0	☑	in	ACCEPT			+dc/vlan10				nolog
1	☑	in	ACCEPT			+dc/vlan50				nolog
2	☑	in	ACCEPT			+dc/vpn				nolog
3	☑	in	ACCEPT			+dc/vlan20				nolog
4	☑	in	ACCEPT			10.0.30.10		10.0.10.10		nolog

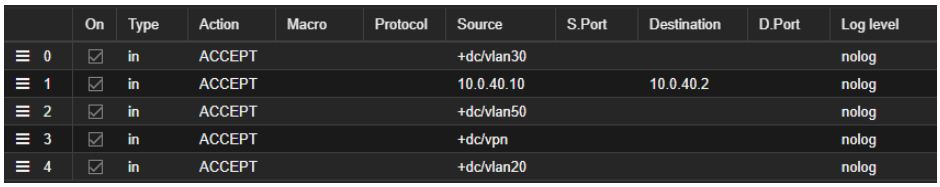
Rysunek 4.28. vlan 10



Rules: Add Copy Remove Edit

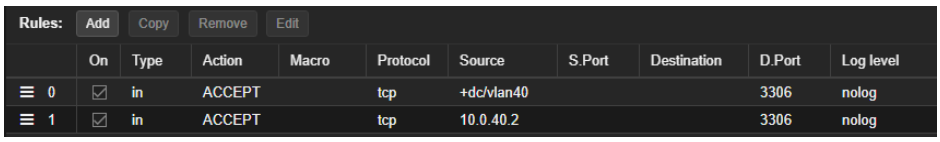
	On	Type	Action	Macro	Protocol	Source	S.Port	Destination	D.Port	Log level
0	☑	in	ACCEPT			+dc/vlan20				nolog
1	☑	in	ACCEPT			+dc/vpn				nolog
2	☑	in	ACCEPT			+dc/vlan30				nolog
3	☑	in	ACCEPT			+dc/vlan50				nolog
4	☑	in	ACCEPT			+dc/vlan10				nolog

Rysunek 4.29. vlan 20



	On	Type	Action	Macro	Protocol	Source	S.Port	Destination	D.Port	Log level
0	☑	in	ACCEPT			+dc/vlan30				nolog
1	☑	in	ACCEPT			10.0.40.10		10.0.40.2		nolog
2	☑	in	ACCEPT			+dc/vlan50				nolog
3	☑	in	ACCEPT			+dc/vpn				nolog
4	☑	in	ACCEPT			+dc/vlan20				nolog

Rysunek 4.30. vlan 30



Rules: Add Copy Remove Edit

	On	Type	Action	Macro	Protocol	Source	S.Port	Destination	D.Port	Log level
0	☑	in	ACCEPT		tcp	+dc/vlan40			3306	nolog
1	☑	in	ACCEPT		tcp	10.0.40.2			3306	nolog

Rysunek 4.31. vlan 40

	On	Type	Action	Macro	Protocol	Source	S.Port	Destination	D.Port	Log level
0	☑	in	ACCEPT			+dc/vlan50				nolog

Rysunek 4.32. vlan 50

	On	Type	Action	Macro	Protocol	Source	S.Port	Destination	D.Port	Log level
≡ 0	☑	in	ACCEPT			+dc/vlan20				nolog
≡ 1	☑	in	ACCEPT			+dc/vpn				nolog
≡ 2	☑	in	ACCEPT			+dc/vlan10				nolog

Rysunek 4.33. vpn

Po wprowadzeniu reguł dostępu sieciowego z poziomu software defined network należy jeszcze wprowadzić zasady ufw dla otwartych portów w zależności od usług osiągalnych na danym kontenerze. Rozwiązanie to jest o tyle wygodne, że grupę zabezpieczeń możemy wprowadzić na każdym urządzeniu w danym vlanie nie przejmując się portami, co bardzo usprawnia segmentacje sieci, a potem wystarczy dodać dla każdego kontenera spersonalizowane zasady oparte tylko na portach. W ten sposób jest znacznie mniej zasad, które należy wprowadzić i znacznie zyskujemy na wydajności.

Na poniższym zdjęciu znajduje się status firewalla UFW na serwerze ftp. Pozwala on na ruch przychodzący jedynie na port 21 i 50000:51000, czyli porty do obsługi serwera ftp.

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
21                         ALLOW IN    Anywhere
50000:51000/tcp            ALLOW IN    Anywhere
21/tcp                     ALLOW IN    Anywhere
20/tcp                     ALLOW IN    Anywhere
21 (v6)                    ALLOW IN    Anywhere (v6)
50000:51000/tcp (v6)       ALLOW IN    Anywhere (v6)
21/tcp (v6)                ALLOW IN    Anywhere (v6)
20/tcp (v6)                ALLOW IN    Anywhere (v6)

root@ftp:~#
```

Rysunek 4.34. Konfiguracja UFW na serwerze ftp

Widoczna poniżej konfiguracja firewalla UFW to konfiguracja na serwerze DNS w VLANie żółtym. Pozwala na ruch przychodzący związany jedynie z protokołem DNS - na porcie 53.

```
root@jeden:/etc/bind# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
53/udp                     ALLOW IN    Anywhere
53/tcp                     ALLOW IN    Anywhere
53/udp (v6)                ALLOW IN    Anywhere (v6)
53/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Rysunek 4.35. Konfiguracja UFW na DNSie w VLANie 10

Poniżej znajduje się konfiguracja UFW na serwerze webowym w VLANie 20. Pozwala ona na ruch przychodzący związany z protokołem HTTPS.

```
root@webserver:/etc/nginx# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
443/tcp                    ALLOW IN    Anywhere
443/tcp (v6)               ALLOW IN    Anywhere (v6)

root@webserver:/etc/nginx#
```

Rysunek 4.36. Konfiguracja UFW na serwerze www w VLANie 20

Poniższa konfiguracja UFW jest konfiguracją dla DNSa w VLANie 30 i pozwala jedynie na ruch związany z protokołem DNS.

```
root@dns3:/etc/bind# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
53                         ALLOW IN    Anywhere
53 (v6)                    ALLOW IN    Anywhere (v6)

root@dns3:/etc/bind#
```

Rysunek 4.37. Konfiguracja UFW na DNSie w VLANie 30

Dla pracowników ustawienia UFW pozwalają jedynie na ruch wychodzący i nie zezwalają na ruch przychodzący.

```
root@pracownik3:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
root@pracownik3:~#
```

Rysunek 4.38. Konfiguracja UFW na komputerze pracownika w VLANie 30

Widoczna poniżej konfiguracja przedstawia konfigurację UFW na serwerze www w VLANie 30.

```
root@webserver3:/home# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
443/tcp                    ALLOW IN    Anywhere
443/tcp (v6)               ALLOW IN    Anywhere (v6)

root@webserver3:/home#
```

Rysunek 4.39. Konfiguracja UFW na serwerze www w VLANie 30

Konfiguracja UFW na skanerze w VLANie 50 nie pozwala na żaden ruch przychodzący.

```
root@Scanner:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
root@Scanner:~#
```

Rysunek 4.40. Konfiguracja UFW na skanerze w VLANie 50

4.9. Testy osiągalności

Poniżej znajduje się wynik poprawnego spingowania pracownego z hosta w VLANie żółtym.

```
root@jeden:~# ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
From 10.0.10.2: icmp_seq=1 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=1 ttl=63 time=1.31 ms
From 10.0.10.2: icmp_seq=2 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=2 ttl=63 time=0.929 ms
From 10.0.10.2: icmp_seq=3 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=3 ttl=63 time=1.01 ms
From 10.0.10.2: icmp_seq=4 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=4 ttl=63 time=0.781 ms
From 10.0.10.2: icmp_seq=5 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=5 ttl=63 time=0.367 ms
From 10.0.10.2: icmp_seq=6 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=6 ttl=63 time=1.90 ms
64 bytes from 10.8.0.6: icmp_seq=7 ttl=63 time=0.406 ms
From 10.0.10.2: icmp_seq=8 Redirect Host(New nexthop: 10.0.10.20)
64 bytes from 10.8.0.6: icmp_seq=8 ttl=63 time=0.449 ms
```

Rysunek 4.41. Ping pracownika zdalnego z VLANa żółtego

Na poniższym zdjęciu widzimy, że zgodnie z oczekiwaniami host w VLANie żółtym nie może spingować hosta w VLANie zielonym.

```
root@jeden:/etc/bind# ping 10.0.30.10
PING 10.0.30.10 (10.0.30.10) 56(84) bytes of data.
^C
--- 10.0.30.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3077ms
```

Rysunek 4.42. Ping hosta w VLANie zielonym z hosta w VLANie żółtym

Ponownie, host w VLANie żółtym nie może skomunikować się z hostem w VLANie czerwonym ani fioletowym

```
root@jeden:/etc/bind# ping 10.0.40.10
PING 10.0.40.10 (10.0.40.10) 56(84) bytes of data.
^C
--- 10.0.40.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3055ms
```

Rysunek 4.43. Ping hosta w VLANie czerwonym z hosta w VLANie żółtym

```
root@jeden:/etc/bind# ping 10.0.50.10
PING 10.0.50.10 (10.0.50.10) 56(84) bytes of data.
^C
--- 10.0.50.10 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1011ms
```

Rysunek 4.44. Ping hosta w VLANie fioletowym z hosta w VLANie żółtym

Poniżej znajduje się dowód, że host z VLANa niebieskiego nie może skomunikować się z pracownikiem zdalnym.

```
root@webserver:/etc/nginx# ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
^C
--- 10.8.0.6 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2067ms
```

Rysunek 4.45. Ping pracownika zdalnego z hosta w VLANie niebieskim

Na poniższym zdjęciu widzimy, że możliwy jest ping hosta w VLANie zielonym z hosta w VLANie niebieskim.

```
root@webserver:/etc/nginx# ping 10.0.30.10
PING 10.0.30.10 (10.0.30.10) 56(84) bytes of data.
64 bytes from 10.0.30.10: icmp_seq=1 ttl=62 time=0.283 ms
64 bytes from 10.0.30.10: icmp_seq=2 ttl=62 time=0.091 ms
64 bytes from 10.0.30.10: icmp_seq=3 ttl=62 time=0.120 ms
64 bytes from 10.0.30.10: icmp_seq=4 ttl=62 time=0.080 ms
^C
--- 10.0.30.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.080/0.143/0.283/0.081 ms
```

Rysunek 4.46. Ping hosta w VLANie zielonym z hosta w VLANie niebieskim

Natomiast, jak widać poniżej, komunikacja z VLANem czerwonym z hosta w VLANie niebieskim jest niemożliwa.

```
root@webserver:/etc/nginx# ping 10.0.40.10
PING 10.0.40.10 (10.0.40.10) 56(84) bytes of data.
^C
--- 10.0.40.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3112ms
```

Rysunek 4.47. Ping hosta w VLANie czerwonym z hosta w VLANie niebieskim

Na poniższym zdjęciu widoczny jest niemożliwy ping hosta w VLANie fioletowym z hosta w VLANie niebieskim.

```
root@webserver:/etc/nginx# ping 10.0.50.10
PING 10.0.50.10 (10.0.50.10) 56(84) bytes of data.

--- 10.0.50.10 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1053ms
```

Rysunek 4.48. Ping hosta w VLANie fioletowym z hosta w VLANie niebieskim

Jak widzimy na zdjęciu poniżej, host w VLANie zielonym nie jest w stanie skomunikować się z pracownikiem zdalnym.

```
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
^C
--- 10.8.0.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1061ms
```

Rysunek 4.49. Ping pracownika zdalnego z hosta w VLANie zielonym

Komunikacja hosta w VLANie zielonym z hostem w VLANie żółtym również nie jest możliwa.

```
root@pracownik3:~# ping 10.0.10.10
PING 10.0.10.10 (10.0.10.10) 56(84) bytes of data.
^C
--- 10.0.10.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2053ms
```

Rysunek 4.50. Ping hosta w VLANie żółtym z hosta w VLANie zielonym

Natomiast, jak widzimy poniżej, z VLANa zielonego można kontaktować się z hostami w VLANie niebieskim.

```
root@pracownik3:~# ping 10.0.20.20
PING 10.0.20.20 (10.0.20.20) 56(84) bytes of data.
64 bytes from 10.0.20.20: icmp_seq=1 ttl=62 time=0.171 ms
64 bytes from 10.0.20.20: icmp_seq=2 ttl=62 time=0.111 ms
64 bytes from 10.0.20.20: icmp_seq=3 ttl=62 time=0.104 ms
^C
--- 10.0.20.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.104/0.128/0.171/0.030 ms
```

Rysunek 4.51. Ping hosta w VLANie niebieskim z hosta w VLANie zielonym

Dodatkowo, z VLANa zielonego jest możliwy kontakt z VLANem czerwonym.

```
root@webserver3:/home# ping 10.0.40.10
PING 10.0.40.10 (10.0.40.10) 56(84) bytes of data.
64 bytes from 10.0.40.10: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 10.0.40.10: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 10.0.40.10: icmp_seq=3 ttl=64 time=0.061 ms
^C
--- 10.0.40.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.043/0.053/0.061/0.007 ms
```

Rysunek 4.52. Ping hosta w VLANie czerwonym z hosta w VLANie zielonym

Natomiast kontakt z VLANu zielonego do fioletowego jest niemożliwy.

```
root@pracownik3:~# ping 10.0.50.10
PING 10.0.50.10 (10.0.50.10) 56(84) bytes of data.
^C
--- 10.0.50.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2026ms
```

Rysunek 4.53. Ping hosta w VLANie fioletowym z hosta w VLANie zielonym

Z hosta w VLANie czerwonym nie da się spingować pracownika zdalnego.

```
root@db:~# ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
^C
--- 10.8.0.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1045ms
```

Rysunek 4.54. Ping z hosta w VLANie czerwonym do pracownika zdalnego

Z VLANa czerwonego nie da się również skontaktować z VLANem żółtym, niebieskim ani fioletowym.

```
root@db:~# ping 10.0.10.10
PING 10.0.10.10 (10.0.10.10) 56(84) bytes of data.

--- 10.0.10.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2050ms
```

Rysunek 4.55. Ping hosta w VLANie żółtym z hosta w VLANie czerwonym

```
root@db:~# ping 10.0.20.20
PING 10.0.20.20 (10.0.20.20) 56(84) bytes of data.
^C
--- 10.0.20.20 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms
```

Rysunek 4.56. Ping hosta w VLANie niebieskim z hosta w VLANie czerwonym

```
root@db:~# ping 10.0.50.10
PING 10.0.50.10 (10.0.50.10) 56(84) bytes of data.
^C
--- 10.0.50.10 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1025ms
```

Rysunek 4.57. Ping hosta w VLANie fioletowym z hosta w VLANie czerwonym

Z hosta w VLANie fioletowym nie da się skontaktować z pracownikiem zdalnym.

```
root@Scanner:~# ping 10.8.0.6
PING 10.8.0.6 (10.8.0.6) 56(84) bytes of data.
^C
--- 10.8.0.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1006ms
```

Rysunek 4.58. Ping pracownika zdalnego z hosta w VLANie czerwonym

Natomiast z hosta w VLANie fioletowym da się skontaktować z VLANem niebieskim, żółtym i zielonym.

```
root@Scanner:~# ping www.biuro.becom
PING www.biuro.becom (10.0.20.20) 56(84) bytes of data.
64 bytes from 10.0.20.20 (10.0.20.20): icmp_seq=1 ttl=62 time=0.072 ms
64 bytes from 10.0.20.20 (10.0.20.20): icmp_seq=2 ttl=62 time=0.114 ms
64 bytes from 10.0.20.20 (10.0.20.20): icmp_seq=3 ttl=62 time=0.123 ms
^C
--- www.biuro.becom ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.072/0.103/0.123/0.022 ms
```

Rysunek 4.59. Ping hosta w VLANie niebieskim i DNSa w VLANie żółtym z hosta w VLANie fioletowym

```
root@Scanner:~# ping 10.0.30.10
PING 10.0.30.10 (10.0.30.10) 56(84) bytes of data.
64 bytes from 10.0.30.10: icmp_seq=1 ttl=63 time=0.180 ms
64 bytes from 10.0.30.10: icmp_seq=2 ttl=63 time=0.072 ms
64 bytes from 10.0.30.10: icmp_seq=3 ttl=63 time=0.092 ms
^C
--- 10.0.30.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.072/0.114/0.180/0.046 ms
```

Rysunek 4.60. Ping hosta w VLANie zielonym z hosta w VLANie fioletowym

5. Audyt

5.1. Wykorzystanie zaawansowanych technik skanowania sieci

W ramach wykonania techniki ofensywnej dokonaliśmy skanowania sieci korporacyjnej w poszukiwaniu znajdujących się w niej hostów. W tym celu wykorzystaliśmy narzędzie nmap. Wyniku skanów nie są do końca zgodne z faktyczną dostępnością w sieci ze względu na to, że nmap wykorzystuje do odnajdowania hostów protokołu ICMP, który na niektórych hostać jest zablokowany przez firewalle.

5.1.1. Segment żółty

Wynik skanu hostów w sieci nmapem z VLANa żółtego wykazał, że z tego VLANa dostępne są hosty w VLANie żółtym, niebieskim, webserver w VLANie zielonym i drugie biuro.

```
root@jeden:~# nmap -sn 10.0.10.0/24 10.0.20.0/24 10.0.30.0/24 10.0.40.0/24 10.0.50.0/24 10.8.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-18 11:12 UTC
Nmap scan report for 10.0.10.2
Host is up (0.000012s latency).
MAC Address: BC:24:11:C2:61:FD (Unknown)
Nmap scan report for 10.0.10.20
Host is up (0.000045s latency).
MAC Address: BC:24:11:4E:ED:7C (Unknown)
Nmap scan report for jeden.jeden (10.0.10.10)
Host is up.
Nmap scan report for 10.0.20.2
Host is up (0.000058s latency).
Nmap scan report for 10.0.20.10
Host is up (0.00011s latency).
Nmap scan report for www.biuro.becom (10.0.20.20)
Host is up (0.00012s latency).
Nmap scan report for 10.0.30.2
Host is up (0.00022s latency).
Nmap scan report for 10.0.30.30
Host is up (0.000082s latency).
Nmap scan report for 10.0.40.1
Host is up (0.000028s latency).
Nmap scan report for 10.0.50.2
Host is up (0.00011s latency).
Nmap scan report for 10.8.0.1
Host is up (0.00015s latency).
Nmap scan report for 10.8.0.6
Host is up (0.00038s latency).
Nmap done: 1536 IP addresses (12 hosts up) scanned in 15.94 seconds
root@jeden:~#
```

Rysunek 5.1. Wynik skanowania z VLANu żółtego

5.1.2. Segment niebieski

Skanowanie sieci z hosta VLANie niebieskim wykazał, że z tego segmentu sieci mamy dostęp do niektórych hostów w VLANie żółty, hostów w VLANie zielonym oraz do drugiego biura.

```
root@webserver:~# nmap -sn 10.0.10.0/24 10.0.20.0/24 10.0.30.0/24 10.0.40.0/24 10.0.50.0/24 10.8.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-18 11:21 UTC
Nmap scan report for 10.0.10.2
Host is up (0.000072s latency).
Nmap scan report for 10.0.10.20
Host is up (0.00013s latency).
Nmap scan report for 10.0.20.2
Host is up (0.000012s latency).
MAC Address: BC:24:11:FC:A1:81 (Unknown)
Nmap scan report for 10.0.20.10
Host is up (0.000032s latency).
MAC Address: BC:24:11:B0:A3:7F (Unknown)
Nmap scan report for webserver.jeden (10.0.20.20)
Host is up.
Nmap scan report for 10.0.30.2
Host is up (0.000086s latency).
Nmap scan report for 10.0.30.10
Host is up (0.00015s latency).
Nmap scan report for 10.0.30.20
Host is up (0.000065s latency).
Nmap scan report for 10.0.30.30
Host is up (0.00011s latency).
Nmap scan report for 10.0.40.1
Host is up (0.000026s latency).
Nmap scan report for 10.0.50.2
Host is up (0.000087s latency).
Nmap scan report for 10.8.0.1
Host is up (0.000090s latency).
Nmap scan report for 10.8.0.6
Host is up (0.00034s latency).
Nmap done: 1536 IP addresses (13 hosts up) scanned in 17.47 seconds
```

Rysunek 5.2. Wynik skanowania z VLANu niebieskiego

5.1.3. Segment zielony

Wynik skanu z hosta w VLANie zielonym wykazał, że z tego VLANu możliwy jest dostęp do VLANu niebieskiego, zielonego oraz do drugiego biura.

```
root@pracownik3:~# nmap -sn 10.0.10.0/24 10.0.20.0/24 10.0.30.0/24 10.0.40.0/24 10.0.50.0/24 10.8.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-18 11:25 UTC
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 512 undergoing Ping Scan
Ping Scan Timing: About 0.59% done
Nmap scan report for 10.0.10.2
Host is up (0.0058s latency).
Nmap scan report for 10.0.10.20
Host is up (0.00048s latency).
Nmap scan report for 10.0.20.2
Host is up (0.00061s latency).
Nmap scan report for 10.0.20.10
Host is up (0.0029s latency).
Nmap scan report for 10.0.20.20
Host is up (0.000056s latency).
Nmap scan report for 10.0.30.2
Host is up (0.000013s latency).
MAC Address: BC:24:11:09:D3:EE (Unknown)
Nmap scan report for 10.0.30.10
Host is up (0.0000080s latency).
MAC Address: BC:24:11:4A:53:39 (Unknown)
Nmap scan report for 10.0.30.30
Host is up (0.000048s latency).
MAC Address: BC:24:11:3A:28:52 (Unknown)
Nmap scan report for pracownik3.dns3 (10.0.30.20)
Host is up.
Nmap scan report for 10.0.40.2
Host is up (0.000019s latency).
Nmap scan report for 10.0.50.2
Host is up (0.0000090s latency).
Nmap scan report for 10.8.0.1
Host is up (0.000046s latency).
Nmap scan report for 10.8.0.6
Host is up (0.00052s latency).
Nmap done: 1536 IP addresses (13 hosts up) scanned in 18.71 seconds
```

Rysunek 5.3. Wynik skanowania z VLANu zielonego

5.1.4. Segment czerwony

Skanowanie sieci z hosta w VLANie czerwonym wykazało dostępność jednego hosta (webservera) w VLANie zielonym.

```
root@db:~# nmap -sn 10.0.10.0/24 10.0.20.0/24 10.0.30.0/24 10.0.40.0/24 10.0.50.0/24 10.8.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-18 11:27 UTC
Nmap scan report for 10.0.30.2
Host is up (0.000020s latency).
Nmap scan report for 10.0.30.30
Host is up (0.031s latency).
Nmap scan report for 10.0.40.2
Host is up (0.000024s latency).
MAC Address: BC:24:11:FB:51:8D (Unknown)
Nmap scan report for db.play.pl (10.0.40.10)
Host is up.
Nmap scan report for 10.0.50.2
Host is up (0.000086s latency).
Nmap done: 1536 IP addresses (5 hosts up) scanned in 57.13 seconds
root@db:~#
```

Rysunek 5.4. Wynik skanowania z VLANu czerwonego

5.1.5. Segment fioletowy

Skanowanie sieci nmapem z perspektywy VLANu fioletowego wykazało dostępność hostów w VLANie niebieskim oraz zielonym.

```
root@Scanner:~# nmap -sn 10.0.10.0/24 10.0.20.0/24 10.0.30.0/24 10.0.40.0/24 10.0.50.0/24 10.8.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-18 11:29 UTC
Nmap scan report for 10.0.10.2
Host is up (0.0099s latency).
Nmap scan report for 10.0.20.2
Host is up (0.000044s latency).
Nmap scan report for 10.0.20.10
Host is up (0.00045s latency).
Nmap scan report for 10.0.20.20
Host is up (0.000058s latency).
Nmap scan report for 10.0.30.2
Host is up (0.000013s latency).
Nmap scan report for 10.0.30.10
Host is up (0.000065s latency).
Nmap scan report for 10.0.30.20
Host is up (0.00038s latency).
Nmap scan report for 10.0.30.30
Host is up (0.00011s latency).
Nmap scan report for 10.0.40.2
Host is up (0.00028s latency).
Nmap scan report for 10.0.50.2
Host is up (0.000022s latency).
MAC Address: BC:24:11:59:91:FD (Unknown)
Nmap scan report for Scanner.dns3 (10.0.50.10)
Host is up.
Nmap done: 1536 IP addresses (11 hosts up) scanned in 34.19 seconds
root@Scanner:~#
```

Rysunek 5.5. Wynik skanowania z VLANu fioletowego

5.1.6. Drugie biuro

Wynik skanowania sieci z hosta w drugim biurze zwrócił dostępność hostów w VLANie niebieskim.

```
root@drugieBiuro:/etc/openvpn/client# nmap -sn 10.0.10.0/24 10.0.20.0/24 10.0.30.0/24 10.0.40.0/24 10.0.50.0/24 10.8.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-18 11:39 UTC
Nmap scan report for 10.0.10.2
Host is up (0.0011s latency).
Nmap scan report for 10.0.10.20
Host is up (0.00097s latency).
Nmap scan report for 10.0.20.2
Host is up (0.0011s latency).
Nmap scan report for 10.0.20.10
Host is up (0.00037s latency).
Nmap scan report for www.biuro.becom (10.0.20.20)
Host is up (0.00070s latency).
Nmap scan report for 10.0.30.2
Host is up (0.0011s latency).
Nmap scan report for 10.0.30.10
Host is up (0.00038s latency).
Nmap scan report for 10.0.30.20
Host is up (0.00021s latency).
Nmap scan report for 10.0.30.30
Host is up (0.00025s latency).
```

Rysunek 5.6. Wynik skanowania z drugiego biura

5.2. Audyt względem standardu

Dokonaliśmy audytu naszej infrastruktury względem dokumentu NIST CSF przeprowadzając ocenę poszczególnych podpunktów według punktacji 0-4 lub N/D. Dokument dołączony jest w pliku Audyt.pdf.

6. Zakończenie

Projektując architekturę sieci wykorzystaliśmy wiedzę uzyskaną nie tylko w trakcie przedmiotu bezpieczeństwo komunikacji, ale także w toku studiów. Praca twórcza w tym obszarze jest jednak wymagająca, gdyż rozwiązań niektórych problemów jest wiele i niełatwo jest wybrać to najbardziej optymalne.