PROJEKT BEZPIECZNEJ ARCHITEKTURY SIECI

Jakub Bliźniuk, Bartłomiej Dmitruk, Kamil Matuszewski, Maciej Matuszewski

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH

BEZPIECZEŃSTWO KOMUNIKACJI

Spis treści

1. Wstęp

Niniejszy dokument przedstawia projekt bezpiecznej architektury sieciowej dla nowego biura, uwzględniający wszystkie wymagania z zakresu segmentacji, bezpieczeństwa i monitorowania.

Na rzecz zwiększenia czytelności w treściach dokumentu znajdować się będą bezpośrednie odwołania do elementów przedstawionych w diagramie architektury, przykładowo: przestrzeń niebieska (odwołanie do określonej przestrzeni), Firewall 10.1.0.1.

2. Część projektowania sieci `Projekt 2.`

2.1. Architektura Firewalli - [`SEG.1`, `SEG.2`]

2.1.1. Umiejscowienie Firewalli

Żółta strefa zdemilitaryzowana chroniona jest przez firewall zewnętrzny - zlokalizowany na granicy z Internetem. Firewall międzystrefowy umieszczony jest za strefą żółtą, między pozostałymi strefami. Dodatkowo każda ze stref posiada dedykowane firewalle

2.1.2. Konfiguracja Firewalli

2.1.2.1. Ogólne zasady konfiguracji firewalli:

Domyślną polityką dla wszystkich firewalli jest DROP, co oznacza automatyczne odrzucanie nieautoryzowanych połączeń. Wszystkie zablokowane połączenia są rejestrowane w logach, podobnie jak połączenia kierowane do stref krytycznych, co umożliwia lepszą kontrolę i analizę potencjalnych zagrożeń. Wszystkie połączenia są obsługiwane z wykorzystaniem stateful inspection, co pozwala na śledzenie ich stanu i identyfikowanie nieautoryzowanych prób dostępu. Na interfejsach zewnętrznych zaimplementowano mechanizmy anti-spoofingu, aby zapobiegać podszywaniu się pod inne adresy IP. Dodatkowo, reguły firewalli są regularnie przeglądane i aktualizowane, co pozwala na dostosowywanie ich do zmieniających się zagrożeń.

Oryginalnie planowaliśmy wykorzystać filtrowanie L7 by ograniczyć przepływy do tylko konkretnych aplikacji, ale niestety okazało się że doświadczenia autorów[1] z rozwiązaniami komercyjnymi jednak nie przenoszą się dobrze na świat otwartoźródłowy i niestety nie ma odpowiednika tej usługi bezpieczństwa w pfSense ani w OPNsense, więc musimy się zadowolić filtrowaniem L4 - z założenia pozwalając tylko na ruch na wykorzystywane przez usługi porty (ale bez bardziej zaawansowanej inspekcji).

2.1.2.2. Kategorie ruchu:

Kategoria ruchu	Usługa	Port/Protokół
Ruch administracyjny	SSH	22/TCP
	Interfejsy zarządzania	443/TCP
	Monitoring i logging	161-162/UDP (SNMP)
Ruch usługowy	HTTP/HTTPS	80,443/TCP
	DNS	53/UDP,TCP
	Aplikacje wewnętrzne	8080,8443/TCP
Ruch bezpieczeństwa	Syslog	514/UDP
	VPN (IPsec)	500,4500/UDP
	VPN (OpenVPN)	1194/UDP
	HIDS/EDR	1514/TCP
Ruch infrastrukturalny	Backup	445/TCP (SMB)
	Aktualizacje systemowe	80,443/TCP
	Synchronizacja czasu (NTP)	123/UDP

Tabela 1: Kategoryzacja ruchu sieciowego z portami

[1] no, jednego autora który to zaproponował

2.1.2.3. Firewall zewnętrzny w strefie żółtej (10.1.0.1)

Głównym zadaniem firewalla zewnętrznego jest filtrowanie ruchu pochodzącego z Internetu. W ramach podstawowej konfiguracji przepuszczany jest ruch VPN oraz HTTP/HTTPS kierowany do serwerów w DMZ, przy jednoczesnym blokowaniu pozostałego ruchu przychodzącego. Firewall realizuje także funkcję NAT dla ruchu wychodzącego. Wszystkie zablokowane połączenia są szczegółowo logowane. Po wstępnej filtracji, dozwolony ruch jest przekazywany do firewalla międzystrefowego.

Źródło	Cel	Port/Protokół	Akcja	Komentarz
Internet	DMZ	443/TCP	PASS	HTTPS dla usług web
Internet	DMZ	500,4500/UDP	PASS	IPsec VPN
Internet	DMZ	1194/UDP	PASS	OpenVPN
Internet	Wszystkie	*	DROP	Domyślna polityka
DMZ	Internet	*	PASS+NAT	Ruch wychodzący
Segmenty wewnętrzne	Internet	*	PASS+NAT	Przez DMZ

Tabela 2: Reguły firewalla zewnętrznego

2.1.2.4. Firewall międzystrefowy

Firewall międzystrefowy pełni kluczową rolę w kontroli ruchu między wszystkimi segmentami sieci. Realizuje kontrolę zgodnie z matrycą dostępu, zarządzając komunikacją z DMZ Gateway do segmentów wewnętrznych oraz między Security Infrastructure a pozostałymi strefami. Szczególną uwagę poświęca kontroli ruchu między DMZ #2 a segmentami wewnętrznymi, zapewniając odpowiednią separację ruchu dla strefy czerwonej oraz kontrolując dostęp do strefy zielonej. Firewall wykonuje zaawansowaną inspekcję pakietów dla całego ruchu międzystrefowego, a wszystkie operacje są centralnie logowane do systemu SIEM.

Interfejsy firewalla międzystrefowego zostały skonfigurowane w sposób umożliwiający efektywną komunikację z każdą strefą.

Interfejs (strefa)	**Adres**	**Rola**
strefa żółta	10.1.0.2/24	komunikacja z zewnętrznym firewallem i usługami DMZ
strefa niebieska (Serwer Usługi X)	10.2.0.2/24	komunikacja z serwerami usług współdzielonych
strefa DMZ #2	10.2.2.2/24	kontrola dostępu do drugiej strefy DMZ
Security Infrastructure	10.2.1.2/24	dostęp do systemów bezpieczeństwa
strefa zielona	10.3.0.2/24	zarządzanie dostępem do środowiska pracy
strefa czerwona	10.4.0.2/24	ścisła kontrola dostępu do strefy krytycznej

Tabela 3: Adresacja interfejsów routera międzystrefowego

2.1.2.5. Reguły dla strefy żółtej (DMZ)

Źródło	Cel	Port/Protokół	Akcja	Komentarz
DMZ	Security Infrastructure	514/UDP	PASS	Syslog
DMZ	Strefa niebieska	80,443/TCP	PASS	HTTP/HTTPS
DMZ	Strefa czerwona	*	DROP	Brak dostępu
DMZ	Strefa zielona	*	DROP	Brak dostępu

Tabela 4: Reguły dla strefy żółtej

2.1.2.6. Reguły dla strefy niebieskiejChildlike-Deck-Alkalize-Palm5-Carded

Źródło	Cel	Port/Protokół	Akcja	Komentarz
Strefa niebieska	DMZ	53/UDP,TCP	PASS	DNS
Strefa niebieska	Security Infrastructure	514/UDP	PASS	Syslog
Strefa niebieska	Strefa zielona	80,443/TCP	PASS	Usługi web
Strefa niebieska	Strefa czerwona	*	DROP	Brak dostępu

Tabela 5: Reguły dla strefy niebieskiej

2.1.2.7. Reguły dla Security Infrastructure

Źródło	Cel	Port/Protokół	Akcja	Komentarz
Security Infrastructure	Wszystkie strefy	22/TCP	PASS	SSH administracyjny
Security Infrastructure	Strefa zielona	8080/TCP	PASS	Wazuh agent
Security Infrastructure	DMZ	1514/TCP	PASS	HIDS komunikacja
Security Infrastructure	Strefa niebieska	1514/TCP	PASS	HIDS komunikacja

Tabela 6: Reguły dla Security Infrastructure

2.1.2.8. Firewall DMZ #2 w strefie niebieskiej (10.2.2.1)

W strefie niebieskiej firewall koncentruje się na ochronie usług w strefie zdemilitaryzowanej tego segmentu. Głównym zadaniem jest kontrola dostępu do serwerów web oraz filtrowanie ruchu między DMZ #2 a pozostałymi segmentami. Podobnie jak pozostałe firewalle, prowadzi szczegółowe logowanie aktywności do systemu SIEM.

Źródło	Cel	Port/Protokół	Akcja	Komentarz
Internet (przez DMZ)	Serwery web	80,443/TCP	PASS	HTTP/HTTPS
Serwery web	Security Infrastructure	514/UDP	PASS	Syslog
Strefa zielona	Serwery web	80,443/TCP	PASS	Dostęp do aplikacji
Wszystkie	Serwery web	*	DROP	Pozostały ruch

Tabela 7: Reguły firewalla DMZ 2

2.1.2.9. Firewall Security Infrastructure (10.2.1.1)

Firewall chroniący Security Infrastructure zapewnia ścisłą kontrolę dostępu do kluczowej infrastruktury bezpieczeństwa, w tym systemów SIEM, EDR i Log Collector. Jego konfiguracja gwarantuje skuteczną separację ruchu monitorującego od ruchu produkcyjnego przy jednoczesnym zapewnieniu kontrolowanego dostępu administracyjnego.

Źródło	Cel	Port/Protokół	Akcja	Komentarz
Wszystkie strefy	SIEM	514/UDP	PASS	Syslog
Strefa zielona	EDR serwer	8080/TCP	PASS	Wazuh agent
Administratorzy	SIEM	5601/HTTPS	PASS	Kibana UI
Administratorzy	SIEM	9200/TCP	PASS	Elasticsearch
Wszystkie	Security Infrastructure	*	DROP	Pozostały ruch

Tabela 8: Reguły firewalla Security Infrastructure

2.1.2.10. Firewall w strefie czerwonej (10.4.0.1)

Ze względu na krytyczny charakter chronionej strefy, firewall w strefie czerwonej implementuje najwyższy poziom restrykcji. Prowadzi ścisłą kontrolę wszystkich połączeń przychodzących i wychodzących przy jednoczesnym szczegółowym logowaniu całej aktywności. Jego głównym zadaniem jest zapewnienie skutecznej izolacji wrażliwych systemów.

Źródło	Cel	Port/Protokół	Akcja	Komentarz
Security Infrastructure	Strefa czerwona	22/TCP	PASS	SSH administracyjny
Strefa czerwona	Security Infrastructure	514/UDP	PASS	Syslog
Strefa czerwona	DMZ	53/UDP,TCP	PASS	DNS
Wszystkie	Strefa czerwona	*	DROP	Pozostały ruch

Tabela 9: Reguły firewalla strefy czerwonej

2.1.2.11. Komunikacja między Firewallami

2.1.2.11.1. Polityka Routingu

Centralnym punktem routingu między strefami jest firewall międzystrefowy, przez którego odpowiednie interfejsy musi przechodzić cały ruch międzystrefowy. Zaimplementowane polityki routingu opierają się na szczegółowych tablicach z uwzględnieniem wymogów bezpieczeństwa.

2.1.2.11.2. Monitoring i Zarządzanie

Kompleksowy monitoring obejmuje wszystkie interfejsy firewalla międzystrefowego przez systemy NIDS. Operacje między strefami są centralnie logowane, a polityki dostępu zarządzane z poziomu Security Infrastructure. System zbiera także metryki wydajności i monitoruje stan wszystkich połączeń.

2.2. Strefy DMZ - [SEG.3]

Nasza architektura zakłada dwie strefy DMZ, które znajdują się w strefie żółtej oraz niebieskiej. Segment żółty oddziela niezaufane sieci lub urządzenia od zasobów wewnętrznych umożliwiając tym samym między innymi: BYOD, zdalny dostęp, czy publiczne serwery. Segment niebieski natomiast pełni rolę segmentu z usługami współdzielonymi, w tym dostępnymi z Internetu (np. serwer web) - tym samym dostęp do Internetu i wystawianie do niego usług realizowane jest właśnie przez DMZ #2. Strefy zdemilitaryzowane korzystać będą z usług bezpieczeństwa takich jak: Firewalle oraz NIDS, które będą kolejno kierować niezbędne informacje do Security Infrastructure, które wyposażone jest w EDR, Log Collector, a także SIEM. Alerty NIDS są wysyłane i agregowane do dedykowanego kolektora logów również znajdującego się w Security Infrastructure. Więcej informacji o usługach bezpieczeństwa można znaleźć w punktach *Architektura Firewalli* oraz *Dokumentacja usług bezpieczeństwa.*

2.3. Plan podziału na VLAN - [SEG.4]

Interfejs (strefa)	VLAN	Sieć
Gateway DMZ (żółta)	VLAN 10	10.1.0.0/24
Serwer Usługi X (niebieski)	VLAN 20	10.2.0.0/24
DMZ 2	VLAN 22	10.2.2.0/24
Security Infrastructure	VLAN 21	10.2.1.0/24
Serwer Host (zielony)	VLAN 30	10.3.0.0/24
Serwer Host (czerwony)	VLAN 40	10.4.0.0/24

Tabela 10: Plan podziału na VLAN

Dla hostów w segmencie zielonym rozwiązania typu EDR wystawione są poprzez usługę serwowaną przez Security Infrastructure wystawioną na Firewallu międzystrefowym. Odpowiednie skonfigurowanie pozwoli na wysyłanie i agregowanie logów z hostów w ramach EDR do dedykowanej części serwerowej EDR, a także na wysyłanie i agregowaine alertów do dedykowanego kolektora logów.

W miarę możliwości chcielibyśmy też użyć, szczgólnie w segmencie zielonym, izolacji portów (private VLAN). Idealnie chcielibyśmy zaimplementować pVLAN wszędzie - uniemożliwiając komunikację L2 z urządzeniami innymi niż sieciowe i te należące do głównego VLANu - ale niestety nie jesteśmy pewni praktycznej możliwości implementacji tego w Open vSwitch przez brak natywnego wsparcia i znaczące problemy ze znalezioną

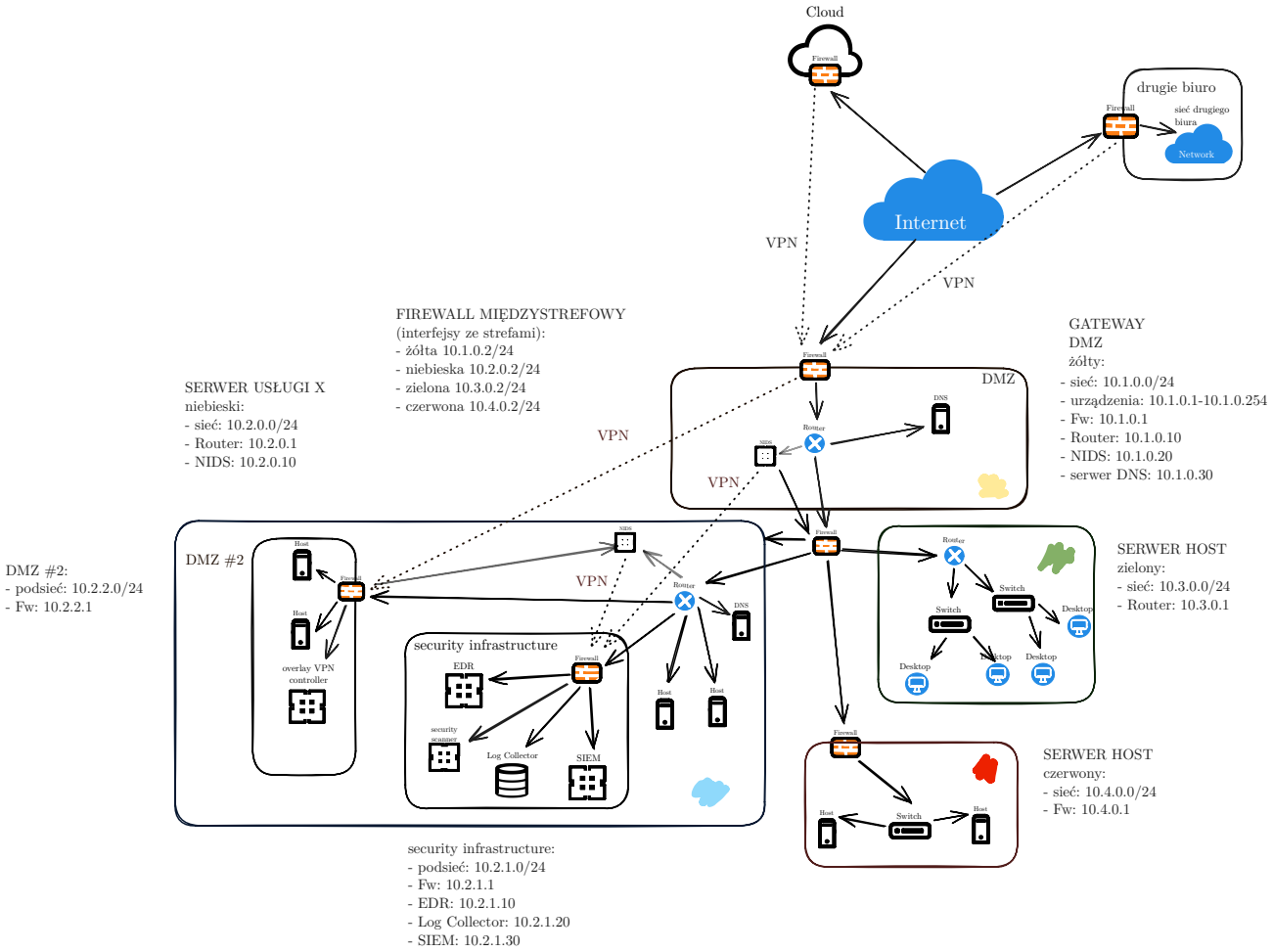
implementacją bazującą na przepływach[2] (brak możliwości użycia ARP przez urządzenia w głównym VLANie, przez co wymagana jest statyczna adresacja MAC), jednak powinno być to możliwe na praktycznie dowolnych fizycznych zarządzalnych switchach klasy enterprise jak i części komeryjnych wirtualnych rozwiązań.

2.4. Konfiguracja portów i usług - [SEG.5]

Host	Cel	Port/Protokół	Nazwa usługi
Serwer web	Internet	80/HTTP	Aplikacja webowa
Serwer web	Internet	443/HTTPS	Aplikacja webowa
SIEM	Administratorzy	5601/HTTPS	Kibana UI
SIEM	Administratorzy	9200/TCP	ElasticSearch
SIEM	Agenci	1514/TCP,UDP	Wazuh Manager
Desktop	SIEM	514/UDP	Syslog
Desktop	SIEM	1514/TCP,UDP	Wazuh Agent
Serwer DNS	Wszystkie strefy	53/UDP,TCP	DNS
Administratorzy	SIEM	22/TCP	SSH
DMZ	Internet	1194/UDP	OpenVPN
DMZ	Internet	4500/UDP	IPsec VPN

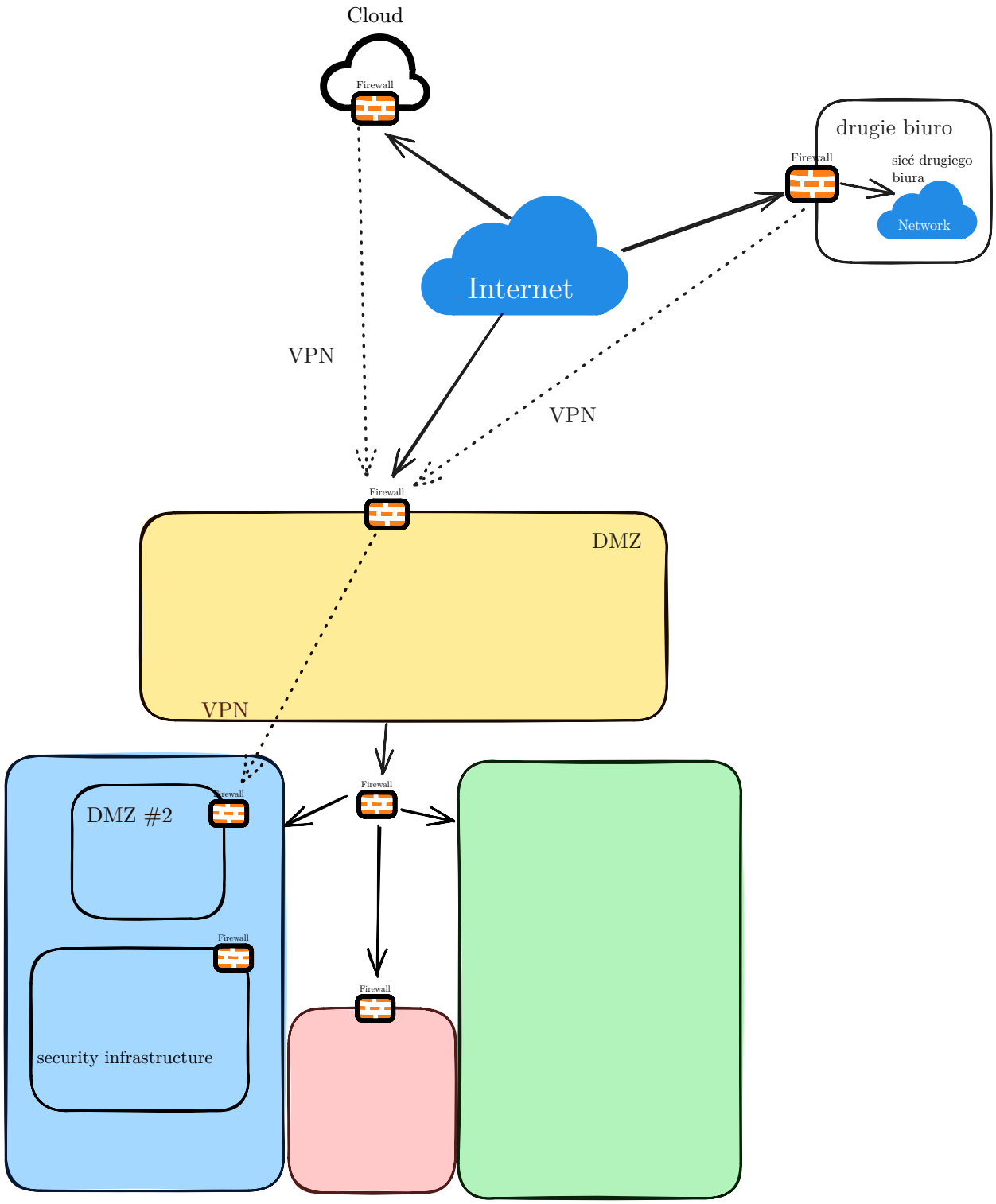
Tabela 11: Konfiguracja portów i usług na hoście

2.5. Architektura sieci - [SEG.6]



Rysunek 1: Pełna architektura sieci z adnotacjami o adresacji

[2]https://wiki.libvirt.org/OVS_and_PVLANS.html



Rysunek 2: Uproszczona architektura sieci - rozmieszczenie firewalli

2.6. Dokumentacja usług bezpieczeństwa - [SEG.7]

2.6.1. Połączenia VPN - [SEC.1, SEC.2]

Połączenia VPN przedstawione są na diagramie architektury sieci przerywaną strzałką (połączenia logiczne). Nasza implementacja uwzględnia zapotrzebowania takie jak:

1. Umożliwienie pracy zdalnej:

- Bezpieczne połączenie przez internet

- Dostęp do zasobów wewnętrznych
- Izolacja ruchu pracownika od reszty sieci

2. Rozwiązanie problemów topologii:

- VPN może „wyprostować" skomplikowane ścieżki routingu
- Tworzy logiczne, bezpośrednie połączenie między punktami
- Enkapsulacja ruchu zwiększa bezpieczeństwo

3. Uwzględnienie DMZ wewnątrz sieci (jak w przypadku DMZ #2):

- VPN izoluje ruch DMZ od reszty infrastruktury
- Pozwala na bardziej precyzyjną kontrolę dostępu
- Redukuje ryzyko związane z „dziurą" w środku sieci

Można powiedzieć, że w naszej architekturze VPN działa jak „plaster na problemy architektury sieciowej", ponieważ pozwala on na odizolowane połączenia między istotnymi elementami naszej infrastruktury: DMZ-y, NIDS-y oraz Security Infrastructure. Ponadto, umożliwia na dostęp/połączenia z Cloud, czy drugim biurem.

2.6.1.1. Site-to-site VPN

Jako nowoczesna (z nawet zewnętrznym wsparciem kryptografii postkwantowej[3]) i bardzo szybka (mimo braku podobnego wsparcia w hardware co IPSec) technologia wyróżnia się tu Wireguard, ale niestety użycie tego protokołu oznaczało by konieczność stawiania własnego firewalla w segmencie chmurowym, co odcina od części integracji sieciowej, natywnego HA i może mieć różny wpływ na koszty (przykładowo, Oracle Cloud nie obciąża bezpośrednio za natywny Site-to-Site VPN wymagając opłaty tylko za przepływ sieciowy) w porównaniu do użycia wbudowanego rozwiązania tego typu - wspierającego zwykle tylko IPSec.

Z tego też powodu projektowi odpowiada tylko IPSec - co powoduje konieczność zdecydowania się na bezpieczną konfigurację tego bardzo rozbudowanego pod tym względem i pełnego przestarzałych opcji protokołu. Konkretniej, wybraliśmy IKEv2 z najwyższymi parametrami wspieranymi przez OCI i Azure:
- ISAKMP: AES-256-CBC, SHA384 (SHA-2) i grupa 20 DH (ECP384)
- IPSec: AES-256-GCM, PFS (grupa 20, ECP384)

2.6.1.2. Remote Access

Ponieważ nie zostało to sprecyzowane w projekcie, zdecydowaliśmy się założyć na potrzeby remote access organizację o naturze hybrydowej w której do pracowników przypisane są jakieś zasoby stacjonarne fizycznie znajdujące się w sieci wewnętrznej (strefie zielonej). By więc umożliwić do nich dostęp - a konsekwentnie z ich pośrednictwem dostęp do sieci na takich samych warunkach jak gdyby byli w biurze - zdecydowaliśmy się zaimplementować VPN overlay/mesh - a konkretniej rozwiązanie VPN P2P z serwerem kontrolnym do zarządzania dostępem (i zwykle też dostarczającego usługi NAT Traversal jak STUN). Najbardziej znanym rozwiązaniem tego typu jest Tailscale - ich serwer kontrolny zamknięte źródło, ale istnieje społeczna alternatywa w postaci Headscale, która jednak oferuje ograniczone możliwości zarządzania wieloma sieciami. Wystarczy to na potrzeby naszego przykładu, ale w pełnej implementacji bardziej odpowiadałby nam bardziej otwarty NetBird lub bardziej skomplikowane (ale i lepiej konfigurowalne) rozwiązania jak Nebula lub NetMaker

2.6.2. Network IDS - [SEC.3]

W architekturze zastosowano dwa systemy NIDS:

- W strefie żółtej (DMZ Gateway) - 10.1.0.20
- W strefie niebieskiej (Serwer usługi X) - 10.2.0.10

Rozmieszczenie NIDS jest strategiczne:
- NIDS w DMZ monitoruje ruch przychodzący z internetu
- NIDS w strefie niebieskiej nadzoruje ruch związany z usługami wewnętrznymi

Dzięki temu prosty port mirroring (SPAN) na dodatkowy port NIDS pozwoli na strategiczne zebranie kluczowego ruchu w sieci.

[3] https://rosenpass.eu/

Ścieżka danych z NIDS:

- Oba systemy NIDS przesyłają dane do Security Infrastructure
- Zapewnia to centralne zbieranie i analizę alertów bezpieczeństwa

2.6.3. EDR - [SEC.4]

Umiejscowienie EDR:

- W strefie Security Infrastructure
- Adres: 10.2.1.10

Integracja z innymi systemami bezpieczeństwa:

- Bezpośrednia komunikacja z SIEM (10.2.1.30)
- Współpraca z Log Collector (10.2.1.20)
- Chroniony przez dedykowany firewall (10.2.1.1)

Zakres monitorowania:

- Potencjalnie wszystkie endpointy w sieci, w tym:
  - Stacje robocze w strefie zielonej
  - Serwery w strefie czerwonej
  - Hosty w DMZ #2
  - Serwery usług w strefie niebieskiej

Architektura zbierania danych:

- Agenty EDR na monitorowanych endpointach
- Centralne zarządzanie z poziomu Security Infrastructure
- Agregacja danych przez SIEM dla kompleksowej analizy

Znaczenie w architekturze bezpieczeństwa:

- Monitoring endpointów w czasie rzeczywistym
- Wykrywanie zagrożeń na poziomie stacji końcowych
- Możliwość szybkiej reakcji na incydenty bezpieczeństwa

2.6.4. Skaner podatności - [SEC.5]

Na rzecz skanowania podatności w segmentach zielonym oraz niebieskim wykorzystywany będzie program Nessus.

2.6.5. SIEM - [SEC.6, SEC.7, SEC.9]

SIEM (10.2.1.30) jest częścią Security Infrastructure:

- Ulokowany w dedykowanym segmencie sieci (10.2.1.0/24)
- Współpracuje z innymi komponentami bezpieczeństwa:
  - Log Collector (10.2.1.20)
  - EDR (10.2.1.10)

Architektura zbierania danych:

- Otrzymuje dane z NIDS z obu stref
- Zbiera logi przez Log Collector
- Integruje się z systemem EDR

Bezpieczeństwo SIEM:

- Chroniony przez dedykowany firewall (10.2.1.1)
- Ulokowany w wydzielonym segmencie bezpieczeństwa
- Ograniczony dostęp z innych stref sieci

Nasza implementacja zapewnia kompleksowe monitorowanie bezpieczeństwa sieci, łącząc dane z różnych źródeł w jednym, centralnym systemie SIEM.

2.6.6. Serwer DNS - [SEC.8, SEC.9]

Umieszczony w segmencie żółtym lokalny serwer DNS pełni funkcję głównego DNS tłumaczącego nazwy na poszczególne adresy IP. Logi DNS z lokalnego serwera DNS są wysyłane i agregowane do dedykowanego kolektora logów SIEM znajdującego się w Security Infrastructure poprzez zrutowanie przez Router, następnie do NIDS strefy żółtej, które poprzez VPN prowadzić będzie do SIEM (jeśli przejdzie filtrowanie firewallem).

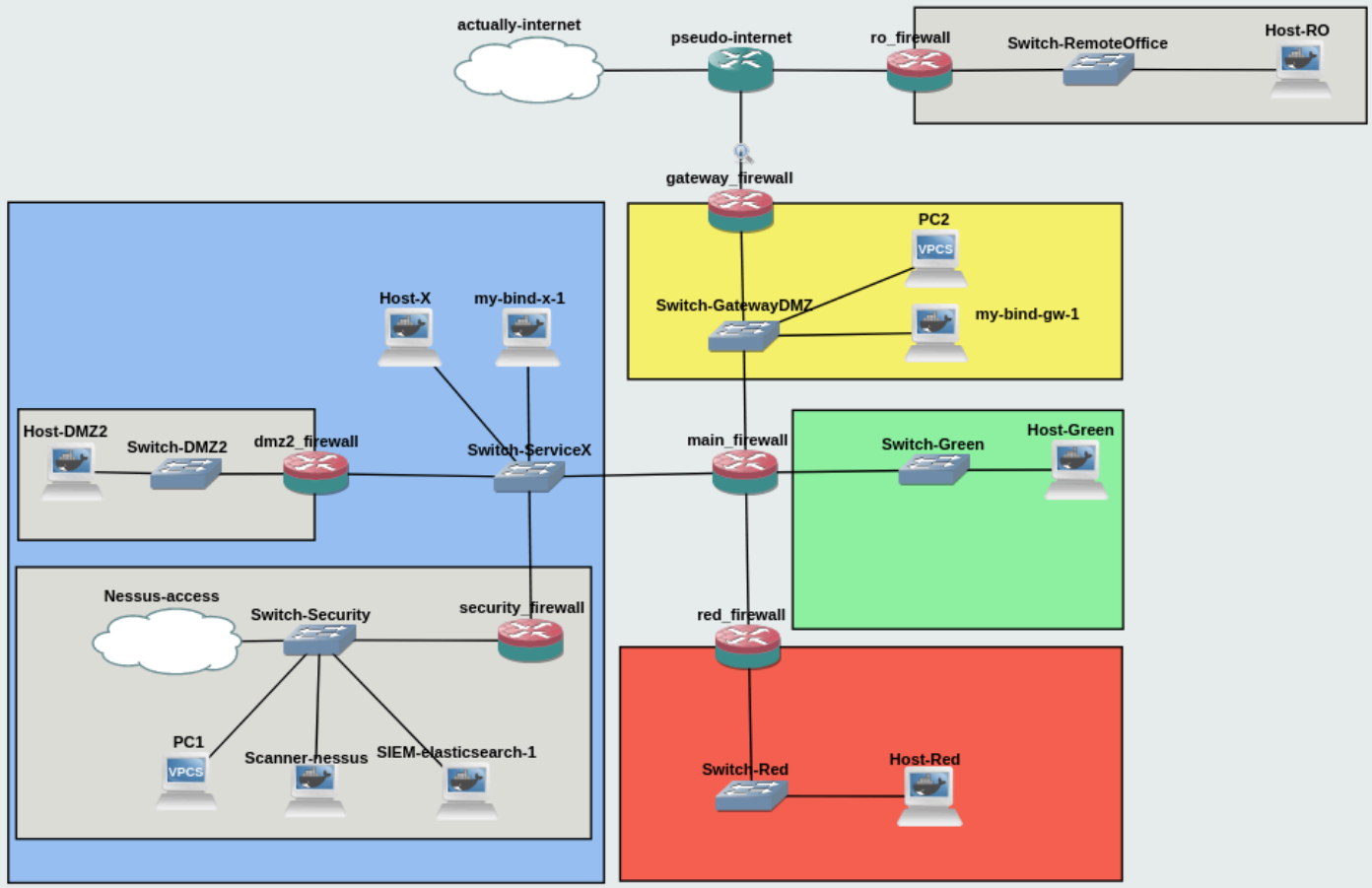
Dodatkowo, na potrzeby obsługi żądań ze strefy zielonej (jak i lokalną obsługę ze strefy niebieskiej, minimalizując bezpośredni footprint usługi ze strefy żółtej) w strefie niebieskiej znajduje się forwarding DNS (z cache na potrzeby wydajności) wykorzystujący serwer w strefie żółtej do rozwiązywania żądań.

2.6.7. Kolektor logów - [SEC.10]

Kolektor logów zgodnie z logiką naszej infrastruktury wystawiony jest w Security Infrastructure.

3. Część wdrożeniowa Lab 2.

3.1. Architektura sieci



Rysunek 3: Architektura sieci

3.2. Dokonane zmiany

Przede wszystkim zdecydowaliśmy się na scalenie Firewalli oraz Routerów. W tym celu wykorzystaliśmy *MikroTik CHR 7.16*. Wybraną przez nas usługą bezpieczeństwa do implementacji został skaner podatności. Zdecydowaliśmy się na rozwiązanie *Nessus*. Dodaliśmy serwer Apache w strefie NIEBIESKIEJ, gdyż w tym miejscu miałyby znajdować się usługi współdzielone z Internetem (Rysunek 3. zawiera zrzut przed dodaniem tej usługi).

3.3. Testy segmentacji, segregacji oraz firewallingu

3.3.1. *ping*

Poniżej przedstawiamy przebieg komendy *ping* przedstawiający realizację wymogu możliwego ruchu międzystrefowego zgodnego z poniższą tabelką:

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

Rysunek 4: Tabela ruchu

3.3.1.1. ŻÓŁTY

```
PC2> ping 10.2.0.100 (do strefy NIEBIESKIEJ)

84 bytes from 10.2.0.100 icmp_seq=1 ttl=63 time=0.590 ms
84 bytes from 10.2.0.100 icmp_seq=2 ttl=63 time=0.449 ms
84 bytes from 10.2.0.100 icmp_seq=3 ttl=63 time=0.533 ms
^C
PC2> ping 10.3.0.100 (do strefy ZIELONEJ)

10.3.0.100 icmp_seq=1 timeout
10.3.0.100 icmp_seq=2 timeout
^C
PC2> ping 10.4.0.100 (do strefy CZERWONEJ)

10.4.0.100 icmp_seq=1 timeout
10.4.0.100 icmp_seq=2 timeout
```

3.3.1.2. NIEBIESKI

```
/ # ping 10.1.0.1 (do strefy ŻÓŁTEJ)
PING 10.1.0.1 (10.1.0.1): 56 data bytes
64 bytes from 10.1.0.1: seq=0 ttl=63 time=0.480 ms
64 bytes from 10.1.0.1: seq=1 ttl=63 time=0.457 ms
64 bytes from 10.1.0.1: seq=2 ttl=63 time=0.450 ms
^C
--- 10.1.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.450/0.462/0.480 ms
/ # ping 10.3.0.100 (do strefy ZIELONEJ)
PING 10.3.0.100 (10.3.0.100): 56 data bytes
64 bytes from 10.3.0.100: seq=0 ttl=63 time=0.248 ms
64 bytes from 10.3.0.100: seq=1 ttl=63 time=0.309 ms
64 bytes from 10.3.0.100: seq=2 ttl=63 time=0.346 ms
^C
--- 10.3.0.100 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.248/0.301/0.346 ms
/ # ping 10.4.0.100 (do strefy CZERWONEJ)
PING 10.4.0.100 (10.4.0.100): 56 data bytes
^C
--- 10.4.0.100 ping statistics ---
```

```
3 packets transmitted, 0 packets received, 100% packet loss
/ #
```

3.3.1.3. ZIELONY

```
/ # ping 10.1.0.1 (do strefy ŻÓŁTEJ)
PING 10.1.0.1 (10.1.0.1): 56 data bytes
^C
--- 10.1.0.1 ping statistics ---
2 packets transmitted, 0 packets received, 100% packet loss
/ # ping 10.2.0.100 (do strefy NIEBIESKIEJ)
PING 10.2.0.100 (10.2.0.100): 56 data bytes
64 bytes from 10.2.0.100: seq=0 ttl=63 time=0.305 ms
64 bytes from 10.2.0.100: seq=1 ttl=63 time=0.252 ms
64 bytes from 10.2.0.100: seq=2 ttl=63 time=0.362 ms
^C
--- 10.2.0.100 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.252/0.306/0.362 ms
/ # ping 10.4.0.100 (do srefy CZERWONEJ - założyliśmy brak przepuszczania ruchu do tej strefuy)
PING 10.4.0.100 (10.4.0.100): 56 data bytes
^C
--- 10.4.0.100 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

3.3.1.4. CZERWONY

```
/ # ping 10.1.0.1 (do strefy ŻÓŁTEJ)
PING 10.1.0.1 (10.1.0.1): 56 data bytes
^C
--- 10.1.0.1 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
/ # ping 10.2.0.100 (do strefy NIEBIESKIEJ)
PING 10.2.0.100 (10.2.0.100): 56 data bytes
^C
--- 10.2.0.100 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
/ # ping 10.3.0.100 (do strefy ZIELONEJ)
PING 10.3.0.100 (10.3.0.100): 56 data bytes
^C
--- 10.3.0.100 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
/ #
```

Polecenia *ping* przeszły zgodnie z naszym zamysłem.

3.3.2. *Nmap*

Postanowiliśmy wykonać skany *Nmap* z poziomu hosta strefy Security - stawiamy się w sytuacji administratora bezpieczeństwa chcącego wykonać audyt bezpieczeństwa i szukającego luk w otwartych portach. W tej perspektywie skanujemy od wewnątrz uzyskując tym samym większe możliwości kolekcjonowania danych.
Flagi nmap:

- sV - wykrywa wersje usług na otwartych portach
- sC - uruchamia podstawowe skrypty NSE (NSE = Nmap Scripting Engine)

sieć ŻÓŁTA:

```
/ # nmap -sV -sC 10.1.0.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 20:03 UTC
Nmap scan report for 10.1.0.1
Host is up (0.00046s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
```

```
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 81:58:16:d5:c5:93:95:93:2a:ba:08:c8:42:f0:54:28 (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
|_http-title: RouterOS
| http-robots.txt: 1 disallowed entry
|_/
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:20 GMT
|     Expires: Sat, 24 Jan 2026 20:03:20 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:20 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
```

```
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.1.0.2
Host is up (0.00038s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 e5:e8:78:72:51:20:13:97:5f:8d:ec:99:32:02:6a:9a (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
| http-robots.txt: 1 disallowed entry
|_/
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:20 GMT
|     Expires: Sat, 24 Jan 2026 20:03:20 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:20 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
```

```
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
|_http-title: RouterOS
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.1.0.30
Host is up (0.00056s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
53/tcp open  domain  NLnet Labs NSD

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (3 hosts up) scanned in 178.77 seconds
/ #
```

sieć NIEBIESKA:

```
/ # nmap -sV -sC 10.2.0.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 20:03 UTC
Nmap scan report for 10.2.0.1
Host is up (0.00051s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 e5:e8:78:72:51:20:13:97:5f:8d:ec:99:32:02:6a:9a (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: RouterOS
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:28 GMT
|     Expires: Sat, 24 Jan 2026 20:03:28 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
```

```
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:28 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.2.0.2
Host is up (0.00038s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ab:01:e4:44:91:28:b7:77:16:b9:fa:c9:37:ec:88:0f (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
|_http-title: RouterOS
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:28 GMT
|     Expires: Sat, 24 Jan 2026 20:03:28 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
```

```
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:28 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
| http-robots.txt: 1 disallowed entry
|_/
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.2.0.3
Host is up (0.00050s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ab:92:69:84:ae:29:6a:71:8c:0b:44:56:98:cb:5e:44 (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
|_http-title: RouterOS
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:28 GMT
|     Expires: Sat, 24 Jan 2026 20:03:28 GMT
|     X-Frame-Options: sameorigin
```

```
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:03:28 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
| http-robots.txt: 1 disallowed entry
|_/
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (3 hosts up) scanned in 177.55 seconds
```

sieć Security:

```
/ # nmap -sV -sC 10.2.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 20:09 UTC
Nmap scan report for 10.2.1.1
Host is up (0.00021s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
```

```
|_  2048 ab:01:e4:44:91:28:b7:77:16:b9:fa:c9:37:ec:88:0f (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
|_http-title: RouterOS
| http-robots.txt: 1 disallowed entry
|_/
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:09:24 GMT
|     Expires: Sat, 24 Jan 2026 20:09:24 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:09:24 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.2.1.10
```

```
Host is up (0.00016s latency).
All 1000 scanned ports on 10.2.1.10 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:87:4B:35:00 (Unknown)

Nmap scan report for 10.2.1.50
Host is up (0.00030s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE      VERSION
22/tcp   open   ssh          OpenSSH 9.9 (protocol 2.0)
1717/tcp closed fj-hdnet
1718/tcp closed h323gatedisc
1719/tcp closed h323gatestat
1720/tcp closed h323q931
1721/tcp closed caicci
1723/tcp closed pptp
1755/tcp closed wms
1761/tcp closed landesk-rc
MAC Address: FA:87:FE:DF:15:10 (Unknown)

Nmap scan report for 10.2.1.100
Host is up (0.0000030s latency).
All 1000 scanned ports on 10.2.1.100 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (4 hosts up) scanned in 179.55 seconds
/ #
```

sieć DMZ#2:

```
/ # nmap -sV -sC 10.2.2.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 20:09 UTC
Nmap scan report for 10.2.2.1
Host is up (0.00052s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ab:92:69:84:ae:29:6a:71:8c:0b:44:56:98:cb:5e:44 (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:09:31 GMT
|     Expires: Sat, 24 Jan 2026 20:09:31 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
```

```
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:09:31 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: RouterOS
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (1 host up) scanned in 171.03 seconds
/ #
```

sieć ZIELONA:

```
/ # nmap -sV -sC 10.2.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 20:09 UTC
Nmap scan report for 10.2.1.1
Host is up (0.00021s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ab:01:e4:44:91:28:b7:77:16:b9:fa:c9:37:ec:88:0f (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
|_http-title: RouterOS
| http-robots.txt: 1 disallowed entry
|_/
```

```
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:09:24 GMT
|     Expires: Sat, 24 Jan 2026 20:09:24 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:09:24 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.2.1.10
Host is up (0.00016s latency).
All 1000 scanned ports on 10.2.1.10 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 02:42:87:4B:35:00 (Unknown)

Nmap scan report for 10.2.1.50
```

```
Host is up (0.00030s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE      VERSION
22/tcp   open   ssh          OpenSSH 9.9 (protocol 2.0)
1717/tcp closed fj-hdnet
1718/tcp closed h323gatedisc
1719/tcp closed h323gatestat
1720/tcp closed h323q931
1721/tcp closed caicci
1723/tcp closed pptp
1755/tcp closed wms
1761/tcp closed landesk-rc
MAC Address: FA:87:FE:DF:15:10 (Unknown)

Nmap scan report for 10.2.1.100
Host is up (0.0000030s latency).
All 1000 scanned ports on 10.2.1.100 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (4 hosts up) scanned in 179.55 seconds
/ # nmap -sV -sC 10.3.0.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-24 20:14 UTC
Nmap scan report for 10.3.0.1
Host is up (0.00056s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
21/tcp   open  ftp            MikroTik router ftpd 7.16
| ftp-syst:
|_  SYST: UNIX MikroTik 7.16
22/tcp   open  ssh            MikroTik RouterOS sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 e5:e8:78:72:51:20:13:97:5f:8d:ec:99:32:02:6a:9a (RSA)
23/tcp   open  telnet         Linux telnetd
80/tcp   open  http
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Cache-Control: max-age=31536000
|     Connection: close
|     Content-Length: 2723
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:14:48 GMT
|     Expires: Sat, 24 Jan 2026 20:14:48 GMT
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang="en">
|     <meta charset="utf-8">
|     <link rel="icon" href="/favicon.png">
|     <link rel="icon" href="/favicon.svg">
|     <title>RouterOS</title>
|     <style>
|     body {
|     font-family: Verdana, Geneva, sans-serif;
|     font-size: 11px;
|     {border: none}
|     img:hover {opacity: 0.8;}
|     font-size: 1.7em;
|     display: inline;
|     margin-bottom: 10px;
|     #container {
|     width: 70%;
```

```
|     margin: 10% auto;
|     #box {
|     background: linear-gradient(#ffffff,#f3f3f3);
|     border: 1px solid #c1c1c1;
|     padding: 30px;
|     .floater {float: left; margin-right: 10px;}
|     .floater label {display: block; text-align: center;}
|     #login {margin: 2em 0 2em 0;}
|     #login td {padding: 0 4px
|   HTTPOptions, RTSPRequest:
|     HTTP/1.0 503 Service Unavailable
|     Cache-Control: no-store
|     Connection: close
|     Content-Length: 109
|     Content-Type: text/html
|     Date: Fri, 24 Jan 2025 20:14:48 GMT
|     Expires: 0
|     Pragma: no-cache
|     X-Frame-Options: sameorigin
|     <!doctype html>
|     <html lang=en>
|     <title>Error 503 : unknown method</title>
|_    <h1>Error 503 : unknown method</h1>
|_http-title: RouterOS
| http-robots.txt: 1 disallowed entry
|_/
2000/tcp open  bandwidth-test MikroTik bandwidth-test server
8291/tcp open  unknown
[...]
Service Info: OS: Linux; Device: router; CPE: cpe:/o:mikrotik:routeros, cpe:/o:linux:linux_kernel

Nmap scan report for 10.3.0.100
Host is up (0.00062s latency).
All 1000 scanned ports on 10.3.0.100 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 177.09 seconds
```

sieć CZERWONA:

W sieci czerwonej nmap nie przynosi żadnych rezultatów, gdyż jest to strefa krytyczna i praktycznie nie ma do niej dostępu z każdego innego punktu (niż wewnątrz sieci czerwonej) topologii - jest to zgodne z założeniem projektu

Analiza skanów nmap ujawnia kilka istotnych aspektów bezpieczeństwa:
1. Kwestie krytyczne:

- Routery MikroTik mają otwarte standardowe porty (21, 22, 23, 80, 2000, 8291)
- Dostępny telnet (port 23) - protokół nieszyfrowany, ale potrzebny do komunikacji z GNS3
- Otwarty FTP (port 21) - również nieszyfrowany
- Serwer HTTP działa bez HTTPS

2. Pozytywne aspekty:

- Strefa czerwona jest dobrze izolowana - brak dostępu z zewnątrz
- DNS (port 53) dostępny tylko w strefie żółtej na dedykowanym serwerze
- Ograniczona liczba hostów w strefach bezpieczeństwa

3. Rekomendacje przy rozwijaniu projektu:

- Wyłączyć telnet - używać tylko SSH
- Zastąpić FTP przez SFTP

- Wdrożyć HTTPS na serwerach web
- Rozważyć zamknięcie portu bandwidth-test (2000)
- Zweryfikować potrzebę dostępu do portu 8291 (zarządzanie MikroTik)
- Wdrożyć dodatkową autentykację dla usług zarządzania

Segmentacja sieci jest prawidłowa, ale zabezpieczenia protokołów wymagają wzmocnienia.W skanach można zauważyć między innymi otwarty port 8291 na Mikrotikach. Można znaleźć informacje w Internecie o istniejącej podatności dla tego portu. Na szczęście tyczy się to wcześniejszych wersji Mikrotika, a nasz *CHR 7.16* nie jest podatny na tego *exploita*.

3.3.3. Wykorzystanie komunikacji z usługą DNS

W ramach projektu zaimplementowaliśmy dwa serwery DNS obsługujące różne segmenty sieci:
1. DNS-GatewayDMZ (10.1.0.30):
2. DNS-X (10.2.0.30):

- Obsługuje zapytania DNS dla sieci:
  - Service X (10.2.0.0/24)
  - DMZ2 (10.2.2.0/24)
  - Security (10.2.1.0/24)
  - Green (10.3.0.0/24)
- Skonfigurowany jako serwer rekurencyjny z forwardingiem na 8.8.8.8 oraz 8.8.4.4

Serwery zostały zaimplementowane jako kontenery Docker oparte na Alpine Linux z zainstalowanym BIND9. Konfiguracja BIND9:
- Ograniczenie zapytań do określonych podsieci
- Wyłączenie IPv6
- Włączenie rekurencji i forwardingu
- Konfiguracja logowania przez syslog

Plik konfiguracyjny dla DNS-GatewayDMZ to:

```
options {
        directory "/var/cache/bind";
        pid-file "/var/run/named/named.pid";

        listen-on { any; };
        listen-on-v6 { none; };

        allow-query {
                10.2.0.0/24;  # niebieski
                10.2.1.0/24;  # security
                10.2.2.0/24;  # dmz2
                10.3.0.0/24;  # green
        };

        forwarders {
                8.8.8.8;
                8.8.4.4;
        };

        auth-nxdomain no;
        recursion yes;
        dnssec-validation auto;
};

logging {
        channel default_syslog {
                syslog daemon;
                severity info;
        };
```

```
        category default { default_syslog; };
        category general { default_syslog; };
};
```

Wchodząc na serwer DNS można podejrzeć kolekcjonowanie logów.
Rezultatami działania serweru DNS jest poniższy przykład zastosowania *ping* do google.com (przy wcześniejszym skonfigurowaniu *nameserver* na hoście):

```
/ # echo "nameserver 10.2.0.30" > /etc/resolv.conf
/ # nslookup google.com
Server:         10.2.0.30
Address:        10.2.0.30:53

Non-authoritative answer:
Name:   google.com
Address: 142.251.37.110

Non-authoritative answer:
Name:   google.com
Address: 2a00:1450:4014:80f::200e

/ # ping google.com
PING google.com (142.251.37.110): 56 data bytes
64 bytes from 142.251.37.110: seq=0 ttl=116 time=17.349 ms
64 bytes from 142.251.37.110: seq=1 ttl=116 time=17.352 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 17.349/17.350/17.352 ms
/ #
```

3.3.4. *Wireshark*

W wybranym przez nas programie *GNS3* integracja *Wiresharka* z infrastrukturą jest bardzo wygodna, ponieważ deweloperzy zadbali o „podejrzenie" ruchu poprzez kliknięcie wybranego linku. W celu zachowania przejrzystości sprawozdania poniżej przedstawiamy przykładowy ruch przechwycony przez *Wireshark* (ruch w poszczególnych miejscach jest definiowany między innymi tablicami routingu oraz regułami firewallowymi):

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x627e0963
2	2.051554	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x60bad916
3	5.274881	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xdf05d4dc
4	6.275916	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa26582ba
5	7.276867	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa26582ba
6	7.412068	10.2.0.100	10.1.0.100	ICMP	98	Echo (ping) request id=0x01c2, seq=0/0, ttl=63 (reply in 7)
7	7.412154	10.1.0.100	10.2.0.100	ICMP	98	Echo (ping) reply id=0x01c2, seq=0/0, ttl=64 (request in 6)
8	7.412274	10.1.0.1	10.1.0.100	ICMP	126	Redirect (Redirect for host)
9	7.477218	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x60bad916
10	8.412050	10.2.0.100	10.1.0.100	ICMP	98	Echo (ping) request id=0x01c2, seq=1/256, ttl=63 (reply in 11)
11	8.412140	10.1.0.100	10.2.0.100	ICMP	98	Echo (ping) reply id=0x01c2, seq=1/256, ttl=64 (request in 10)
12	8.412259	10.1.0.1	10.1.0.100	ICMP	126	Redirect (Redirect for host)
13	9.412151	10.2.0.100	10.1.0.100	ICMP	98	Echo (ping) request id=0x01c2, seq=2/512, ttl=63 (reply in 14)
14	9.412271	10.1.0.100	10.2.0.100	ICMP	98	Echo (ping) reply id=0x01c2, seq=2/512, ttl=64 (request in 13)
15	9.412437	10.1.0.1	10.1.0.100	ICMP	126	Redirect (Redirect for host)
16	9.799164	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x60bad916
17	9.919482	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa26582ba
18	10.412119	10.2.0.100	10.1.0.100	ICMP	98	Echo (ping) request id=0x01c2, seq=3/768, ttl=63 (reply in 19)
19	10.412219	10.1.0.100	10.2.0.100	ICMP	98	Echo (ping) reply id=0x01c2, seq=3/768, ttl=64 (request in 18)
20	10.412334	10.1.0.1	10.1.0.100	ICMP	126	Redirect (Redirect for host)
21	11.412148	10.2.0.100	10.1.0.100	ICMP	98	Echo (ping) request id=0x01c2, seq=4/1024, ttl=63 (reply in 22)
22	11.412244	10.1.0.100	10.2.0.100	ICMP	98	Echo (ping) reply id=0x01c2, seq=4/1024, ttl=64 (request in 21)
23	11.412401	10.1.0.1	10.1.0.100	ICMP	126	Redirect (Redirect for host)
24	12.412157	10.2.0.100	10.1.0.100	ICMP	98	Echo (ping) request id=0x01c2, seq=5/1280, ttl=63 (reply in 25)
25	12.412247	10.1.0.100	10.2.0.100	ICMP	98	Echo (ping) reply id=0x01c2, seq=5/1280, ttl=64 (request in 24)
26	12.412368	10.1.0.1	10.1.0.100	ICMP	126	Redirect (Redirect for host)
27	12.482786	0c:bb:ae:57:00:00	Private_66:68:01	ARP	42	Who has 10.1.0.100? Tell 10.1.0.1
28	12.482870	Private_66:68:01	0c:bb:ae:57:00:00	ARP	42	10.1.0.100 is at 00:50:79:66:68:01
29	12.821511	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x60bad916
30	13.041937	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa26582ba

Rysunek 5: Przykładowe przechwycenie ruchu *ping* z hosta niebeiskiego do hosta żółtego dla linku Switch--GatewayDMZ i PC2

3.3.5. Propozycja modelu operacyjnego wykorzystania usługi

Nasz model realizuje model operacyjny wykorzystania usługi realizującej:

1. Segmentacja sieci:

- Strefa niebieska - zawiera serwery dostępowe i usługi publiczne ubogacona o strefy Security oraz DMZ#2
- Strefa żółta - sieć DMZ gateway
- Strefa zielona - realizacja podstawowego środowiska biura
- Strefa czerwona - wyizolowana sieć dla szczególnie wrażliwych systemów

2. Kontrola dostępu:

- Wielopoziomowa ochrona przez firewalle
- Separacja ruchu między strefami
- Dedykowana strefa Security

3. Model bezpieczeństwa:

- Strefa DMZ jako bufor między internetem a siecią wewnętrzną
- Skanowanie ruchu przez dedykowany system *Nessusa*
- Kontrola dostępu do Internetu w strefie DMZ Gateway

4. Zarządzanie:

- Centralne zarządzanie przez gateway_firewall
- Dedykowane switche dla każdej strefy funkcjonalnej
- Monitoring bezpieczeństwa

Dostęp zdalny:
- Realizowany przez ro_firewall i Switch-RemoteOffice
- Separacja ruchu zdalnego od sieci wewnętrznej

5. Skalowalność:

- Możliwość rozbudowy każdej ze stref
- Elastyczna architektura pozwalająca na dodawanie nowych usług i systemów

Nasz model zapewnia wielowarstwowe bezpieczeństwo przy jednoczesnym zachowaniu funkcjonalności i możliwości rozwoju infrastruktury.

3.4. Site-to-site VPN

3.4.1. Drugie biuro

3.4.1.1. Konfiguracja IPSEC

gateway_firewall:

```
[admin@MikroTik] /ip/ipsec> peer print
Flags: X - disabled; D - dynamic; R - responder
 0     name="peer1" address=123.123.123.6/32 profile=default exchange-mode=main
       send-initial-contact=yes
[admin@MikroTik] /ip/ipsec> policy print
Flags: T - TEMPLATE; A - ACTIVE; * - DEFAULT
Columns: PEER, TUNNEL, SRC-ADDRESS, DST-ADDRESS, PROTOCOL, ACTION, LEVEL, PH2-COUNT
#     PEER   TUNNEL  SRC-ADDRESS  DST-ADDRESS     PROTOCOL  ACTION   LEVEL    P
0 T *                ::/0         ::/0            all
1  A  peer1  yes     10.0.0.0/14  192.168.0.0/24  all       encrypt  require  2
[admin@MikroTik] /ip/ipsec> active-peers print
Flags: R - RESPONDER
Columns: STATE, UPTIME, PH2-TOTAL, REMOTE-ADDRESS
#    STATE        UPTIME  PH2-TOTAL  REMOTE-ADDRESS
0    established  21m25s          1  123.123.123.6
1 R  established  21m25s          1  123.123.123.6
```

ro_firewall:

```
[admin@MikroTik] /ip/ipsec> peer print
Flags: X - disabled; D - dynamic; R - responder
 0     name="peer1" address=123.123.123.2/32 profile=default exchange-mode=main
       send-initial-contact=yes
[admin@MikroTik] /ip/ipsec> policy print
Flags: T - TEMPLATE; A - ACTIVE; * - DEFAULT
Columns: PEER, TUNNEL, SRC-ADDRESS, DST-ADDRESS, PROTOCOL, ACTION, LEVEL, PH2-COUNT
#     PEER   TUNNEL  SRC-ADDRESS     DST-ADDRESS  PROTOCOL  ACTION   LEVEL    P
0 T *                ::/0            ::/0         all
1  A  peer1  yes     192.168.0.0/24  10.0.0.0/14  all       encrypt  require  4
[admin@MikroTik] /ip/ipsec> active-peers print
Flags: R - RESPONDER
Columns: STATE, UPTIME, PH2-TOTAL, REMOTE-ADDRESS
#    STATE        UPTIME  PH2-TOTAL  REMOTE-ADDRESS
0    established  24m13s          2  123.123.123.2
1 R  established  24m13s          2  123.123.123.2
```

3.4.1.2. Potwierdzenie łączności

```
PC2> ping 192.168.0.101

84 bytes from 192.168.0.101 icmp_seq=1 ttl=62 time=0.811 ms
84 bytes from 192.168.0.101 icmp_seq=2 ttl=62 time=0.954 ms
84 bytes from 192.168.0.101 icmp_seq=3 ttl=62 time=0.791 ms
84 bytes from 192.168.0.101 icmp_seq=4 ttl=62 time=0.794 ms
84 bytes from 192.168.0.101 icmp_seq=5 ttl=62 time=0.707 ms
```

3.4.1.3. *Wireshark*

W ramach testu konfiguracji naszego rozwiązania Site-To-Site VPN wykonaliśmy ping z hosta w żółtej strefie o adresie 10.1.0.100 do hosta w drugim biurze o adresie 192.168.0.101. Jak widać, na odcinku do gateway'a hosta z żółtej strefy pakiety są wysyłane w tradycyjnej formie z użyciem protokołu ICMP i nie są jeszcze zaszyfrowane:

```
31.094626   192.168.0.101   10.1.0.100      ICMP   98 Echo (ping) request  id=0x4ab0, seq=1/256, ttl=62 (reply in 33)
31.094704   10.1.0.100      192.168.0.101   ICMP   98 Echo (ping) reply    id=0x4ab0, seq=1/256, ttl=64 (request in 32)
32.095713   192.168.0.101   10.1.0.100      ICMP   98 Echo (ping) request  id=0x4bb0, seq=2/512, ttl=62 (reply in 35)
32.095777   10.1.0.100      192.168.0.101   ICMP   98 Echo (ping) reply    id=0x4bb0, seq=2/512, ttl=64 (request in 34)
```

Rysunek 6: Pakiety między PC2, a Switch-GatewayDMZ

Szyfrowanie następuje na routerze brzegowym, który enkapsuluje pakiety przed opuszczeniem prywatnej sieci i przekazaniem ich do internetu. Pakiety są przesyłane za pomocą protokołu ESP (Encapsulating Security Payload), co zapewnia poufność, integralność oraz uwierzytelnienie danych. Po zaszyfrowaniu zmieniają się adresy IP w nagłówku zewnętrznym – teraz wskazują one na adresy routerów znajdujących się na końcach tunelu VPN. Jest to zgodne z działaniem trybu tunelowego IPsec, w którym szyfrowany jest cały pakiet, łącznie z oryginalnymi nagłówkami IP.

```
123.123.123.6   123.123.123.2   ESP   166 ESP (SPI=0x02f8653a)
123.123.123.2   123.123.123.6   ESP   166 ESP (SPI=0x0a0df1fa)
123.123.123.6   123.123.123.2   ESP   166 ESP (SPI=0x02f8653a)
123.123.123.2   123.123.123.6   ESP   166 ESP (SPI=0x0a0df1fa)
```

Rysunek 7: Pakiety między pseudo-internet, a ro_firewall

Po przejściu przez router drugiego biura wiadomość zostaje odszyfrowana i ponownie jest przesyłana po protokole ICMP z oryginalnymi adresami source (10.1.0.100) i destination (192.168.0.101).

```
192.168.0.101   10.1.0.100      ICMP   98 Echo (ping) request  id=0x4ab0, seq=1/256, ttl=64 (reply in 6)
10.1.0.100      192.168.0.101   ICMP   98 Echo (ping) reply    id=0x4ab0, seq=1/256, ttl=62 (request in 5)
192.168.0.101   10.1.0.100      ICMP   98 Echo (ping) request  id=0x4bb0, seq=2/512, ttl=64 (reply in 8)
10.1.0.100      192.168.0.101   ICMP   98 Echo (ping) reply    id=0x4bb0, seq=2/512, ttl=62 (request in 7)
```

Rysunek 8: Pakiety między Switch-RemoteOffice, a PC3

3.5. Skaner podatności - *Nessus*

3.5.1. Skanowanie

Wykorzystanie skanera zaprezentowaliśmy w części 3. projektu.

3.5.2. Propozycja integracji

Nasz skaner Nessus jest w pełni zintegrowany z siecią wewnętrzną biura i tym samym jest w stanie realizować skany podatności wybranych podsieci.

3.5.3. Propozycja modelu operacyjnego wykorzystania usługi

Nasz model operacyjny Nessusa uwzględnia:

1. Architektura wdrożenia

- Lokalizacja skanera: strefa Security
- Dedykowany interfejs skanujący
- Separacja ruchu skanowania od ruchu produkcyjnego
- Dostęp administracyjny tylko z określonych hostów zarządzających

Idealistycznie założylibyśmy (przy dalszym rozwoju infrastruktury oraz większym zasobie czasowym) uwzględnienie:

2. Polityka skanowania:

- Harmonogram skanów:
  - Pełne skany: raz w miesiącu
  - Skany krytycznych systemów: co tydzień
  - Skany przyrostowe: codziennie
  - Skany ad-hoc: na żądanie przy zmianach
- Zakres skanowania (zrealizowane w naszym „Demo”):
  - Podział na grupy systemów:
    - Systemy użytkowe (strefa czerwona)
    - Systemy DMZ Gateway (strefa żółta)
    - Systemy usług współdzielonych (strefa niebieska)
    - Systemy izolowane (strefa czerwona)

3. Zarządzanie wynikami:

- Automatyzacja raportowania:
  - Raporty dzienne dla nowych podatności
  - Raporty tygodniowe dla trendu podatności
  - Raporty miesięczne dla managementu
- Integracja z systemem ticketowym
- Priorytetyzacja podatności (CVSS)

4. Procedury operacyjne:

- Weryfikacja wyników (eliminacja false-positive)
- Eskalacja krytycznych podatności
- Śledzenie procesu naprawy
- Weryfikacja skuteczności napraw
- Dokumentowanie wyjątków

4. Audyt bezpieczeństwa sieci `Projekt 3., Laboratorium 3.`

4.1. Eksploitacja + reverse shell na wybranym hoście `AUD.ACT.1.1`

W naszym scenariuszu zakładamy podejście `grey box`, to znaczy:

- znamy sieć i mamy jej projekt bezpieczeńswtwa
- mamy dostęp do sieci w podejściu zdalnym - w naszym scenariuszu łączymy się przez „Internet”

- mamy dostęp do wszystkich hostów w ramach architektury - zdalnego (reprezentacja sieci zdalnej) oraz wszystkich hostów lokalnych

Z tą wiedzą koncentrujemy swój atak na strefie niebieskiej, ponieważ jest to segment z usługami współdzielonymi, w tym dostępnymi z Internetu w dodatku posiadający serwer Apache - podmiot często zawierający podatności i łatwy do sterowania, szczególnie jeśli jest zainstalowany na Linuxie. Mając dostęp do projektu bezpieczeństwa zauważamy, że architekci zaplanowali użycie wersji *Apache HTTP Server 2.4.49*, co jest krytyczne pod względem bezpieczeństwa, gdyż istnieją podatności opisane w Internecie (przez co nawet średnio zaawansowany cyberprzestępca jest w stanie dokonać nadużycia). Tym samym znajdujemy *exploita* `CVE-2021-41773 / CVE-2021-42013` w *exploit-db*, który posłuży nam jako „narzędzie zbrodni".
Krótko o *exploicie*:
1. Podatność:

- Pozwala na zdalne wykonanie kodu (RCE) poprzez nieprawidłową normalizację ścieżek URL w konfiguracji CGI.

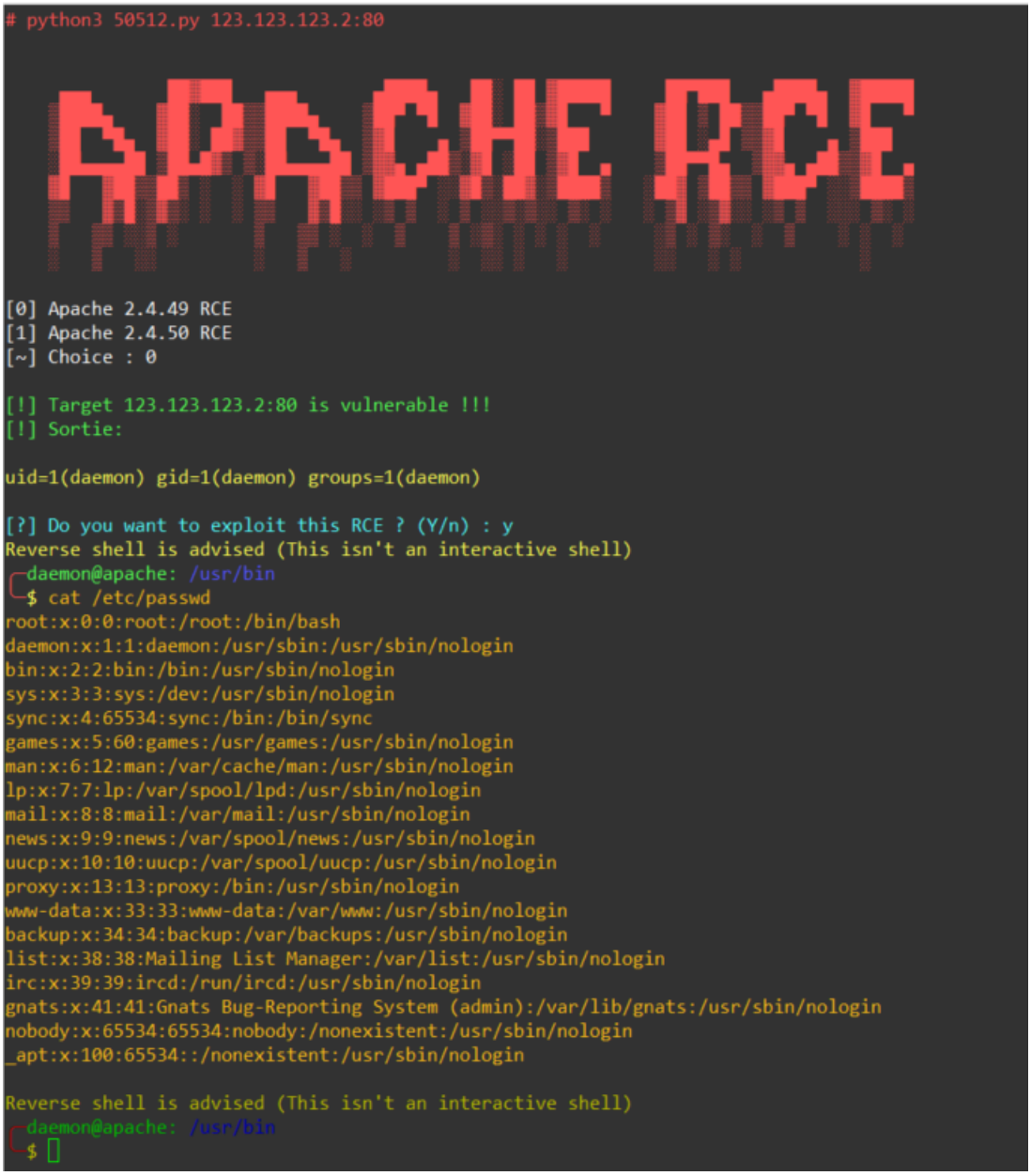
2. Mechanizm działania:

- Celem jest 123.123.123.2:80 (translacja NAT jest włączona)
- Utrzymanie komunikacji dzieje się poprzez serwer HTTP. Jest to idealne rozwiązanie, gdyż na Firewallach będzie przepuszczony ten ruch z uwagi na funkcjonalność serwera.
- Wykorzystuje specjalnie spreparowane ścieżki URL z sekwencjami ucieczkowymi (.%2e)
- Daje dostęp do /bin/bash poprzez traversal katalogów
- Payload dla 2.4.49: /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/bin/bash
- Payload dla 2.4.50: wykorzystuje dodatkowo podwójne kodowanie (%32%65)
  - Kodowanie te wpisuje się w Path Traversal w którym to chcemy uzyskać dostęp do plików i katalogów przechowywanych poza głównym katalogiem aplikacji webowej Apache. Wykorzystuje sekwencje takie jak „../" (zakodowane jako %2e%2e/) do nawigacji w górę drzewa katalogów.

3. Funkcjonalność:

- Sprawdza podatność poprzez próbę wykonania komendy „id"
- Po potwierdzeniu podatności udostępnia pseudo-shell
- Umożliwia wykonywanie poleceń na serwerze
- Wspiera podstawowe komendy jak pwd, whoami, clear

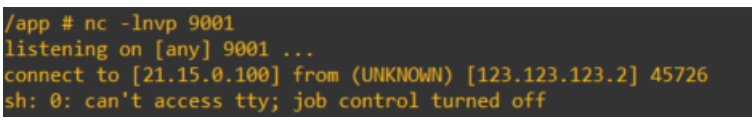
W GNS3 maszyna atakująca podłączona jest do *pseudo-internet*, natomiast serwer Apache, tak jak wspominaliśmy, do strefy NIEBIESKIEJ.
1. Na maszynie atakującej uruchamiamy *exploit*:



Rysunek 9: Uruchomienie *exploita*

2. Uzyskujemy RCE po porcie HTTP - próba uruchomienia *Reverse Shella* jest naturalnym następcą naszego działania („próba", ponieważ musimy przetestować czy ruch po innym porcie niż 80 zadziała).

3. Testujemy różne sposoby uzyskania zdalnego dostępu z pomocą witryny https://www.revshells.com/. *Reverse Shell* działa (w środowisku nietestowym można by się spodziewać adresu z puli publicznej). Ruch Firewalli



Rysunek 10: Uruchomienie *Reverse Shell*

4. Z tego poziomu atakujący może już dużo, my reprezentacyjnie przedstawimy wyświetlenie i wykradnięcie informacji.

5. W dowolnej chwili możemy zakończyć działanie. Oczywiście, wachlarz możliwości jest rozległy i przy odpowiednich uprawnieniach moglibyśmy chociażby ustawić C2 po HTTP w celu utrzymania komunikacji (albo do momentu załatania podatności przez architektów do ponownego łączenia się, jeśli wystąpiłaby taka chęć).

6. Przykładowo wykorzystaliśmy *linpeas.sh* do działań eksfiltracji danych, takich jak:

- Wyszukiwanie plików konfiguracyjnych, kopii zapasowych i logów.
- Sprawdzanie plików z potencjalnie wrażliwymi uprawnieniami.
- Analizowanie plików bash_history, ssh oraz innych. Wykrywanie hasłowych fraz w plikach (np. password, pwd).
- Zbieranie informacji o systemie, np. działających usługach, procesach, czy uprawnieniach.

Output z narzędzia znajduje się w Sekcja 5 (Załączniki).

4.2. Audyt względem standardu NIST Cybersecurity Framework `AUD.NIST`

Wyniki audytu zgodności ze standardem NIST Cybersecurity Framework znajduje się w Sekcja 5 (Załączniki).

W procesie audytu zastosowaliśmy następującą klasyfikację wyników:
- *N/A* - nie dotyczy
- *N/D* - nie określono
- *C* - zgodny
- *PC* - częściowo zgodny
- *NC* - niezgodny

Ze względu na specyfikę pracy w środowisku wirtualnym oraz brak wdrożonego projektu, większość reguł kontrolnych nie mogła zostać w pełni oceniona. Większość reguł kontrolnych mających odzwierciedlenie w naszym projekcie było zgodne lub częściowo zgodne ze standardem NIST CF.

Ciekawym doświadczeniem dla nas okazała się praca przy audycie zgodności ze standardami bezpieczeństwa, może być to przydatna pod kątem przyszłości wiedza.

4.3. Skanowanie podatności z wykorzystaniem Nessus

W ramach audytu bezpieczeństwa przeprowadzono szczegółowe skanowanie sieci przy użyciu narzędzia Nessus. Skanowanie zostało wykonane w dniu 25 stycznia 2025 roku i objęło wszystkie urządzenia w badanej sieci. Wygenerowane raporty dołączyliśmy do sprawozdania w Sekcja 5 (Załączniki).

Podczas skanowania szczególną uwagę zwrócono na serwer Apache działający na hoście 10.2.0.40, gdzie wykryto najwięcej krytycznych podatności. Na tym hoście zidentyfikowano dziewięć podatności o poziomie krytycznym, siedem o wysokim poziomie ryzyka oraz dwie o średnim poziomie ryzyka. Wszystkie te podatności związane są z zainstalowaną wersją Apache 2.4.49. Wśród najpoważniejszych znalezisk znalazły się podatności typu Path Traversal, Buffer Overflow w module mod_lua oraz problemy związane z HTTP Request Smuggling.

Skanowanie ujawniło również szereg otwartych portów i aktywnych usług na różnych hostach w sieci. Większość hostów odpowiadała na zapytania ICMP, co świadczy o ich dostępności w sieci. Na wielu urządzeniach wykryto niezabezpieczone porty TCP, które mogą stanowić potencjalne punkty wejścia dla atakujących.

Wśród wykrytych problemów bezpieczeństwa znalazły się także podatności związane z protokołami sieciowymi oraz kwestie dotyczące uwierzytelniania. Na niektórych hostach zidentyfikowano problemy z konfiguracją usług sieciowych, co może prowadzić do nieautoryzowanego dostępu lub wycieków informacji.

Skan wykazał również obecność podatności typu ICMP Timestamp Request, która może umożliwić atakującym uzyskanie informacji o czasie systemowym urządzeń. Choć sama w sobie nie jest krytyczna, może być wykorzystana jako element bardziej złożonych ataków.

Przeprowadzone skanowanie pokazało, że infrastruktura wymaga szczególnej uwagi w zakresie aktualizacji oprogramowania, zwłaszcza serwera Apache, oraz właściwej konfiguracji zabezpieczeń sieciowych. Regularne przeprowadzanie takich skanów może pomóc w utrzymaniu odpowiedniego poziomu bezpieczeństwa sieci.

4.4. Prezentacja

W ramach realizacji projektu opracowaliśmy również prezentację podsumowującą wyniki prac testowych i audytowych. Prezentacja ta może zostać wykorzystana podczas spotkania podsumowującego projekt bezpiecznej sieci. Dokument został przygotowany w celu przedstawienia wyników przed różnymi odbiorcami, w tym technicznymi i nietechnicznymi. Link do prezentacji znajduje się w Sekcja 5 (Załączniki).

4.5. Rekomendacje

Na podstawie przeprowadzonego audytu oraz analizy podatności, przedstawiamy kluczowe rekomendacje mające na celu zwiększenie bezpieczeństwa sieci

4.5.1. Aktualizacja i zarządzanie oprogramowaniem

- Przeprowadzić natychmiastową aktualizację Apache do wersji 2.4.60 lub nowszej w celu usunięcia wykrytych podatności
- Wdrożyć system regularnych aktualizacji i zarządzania poprawkami bezpieczeństwa
- Utrzymywać aktualną dokumentację używanego oprogramowania i jego wersji

4.5.2. Wzmocnienie zabezpieczeń sieci

- Zrewidować i zaostrzyć reguły firewalla, stosując zasadę najmniejszych uprawnień
- Wprowadzić wieloskładnikowe uwierzytelnianie dla dostępu do kluczowych systemów
- Ograniczyć dostęp do portów i usług tylko do niezbędnego minimum

4.5.3. Monitoring i wykrywanie zagrożeń

- Rozszerzyć system monitorowania ruchu sieciowego o dodatkowe punkty kontrolne
- Wdrożyć automatyczną analizę logów w systemie SIEM
- Regularnie przeprowadzać skany podatności i testy bezpieczeństwa
- Skonfigurować alerty dla podejrzanych aktywności w sieci

4.5.4. Procedury bezpieczeństwa

- Opracować i regularnie testować plan reagowania na incydenty
- Wdrożyć system regularnych kopii zapasowych krytycznych systemów
- Przeprowadzać regularne szkolenia zespołu z zakresu bezpieczeństwa
- Dokumentować i aktualizować procedury bezpieczeństwa

4.5.5. Konfiguracja usług

- Wyłączyć wszystkie nieużywane usługi i porty
- Zmienić domyślne hasła i ustawienia konfiguracyjne
- Regularnie przeglądać i aktualizować polityki dostępu
- Zabezpieczyć punkty końcowe i interfejsy administracyjne

4.5.6. Zarządzanie dostępem

- Wprowadzić ścisłą kontrolę dostępu do zasobów sieciowych
- Regularnie przeglądać i aktualizować uprawnienia użytkowników
- Wdrożyć system zarządzania tożsamością i dostępem
- Monitorować i rejestrować wszystkie próby dostępu do krytycznych systemów

Wdrożenie powyższych rekomendacji powinno znacząco zwiększyć poziom bezpieczeństwa sieci i zminimalizować ryzyko potencjalnych ataków. Zalecamy regularne przeglądy i aktualizacje tych zabezpieczeń w miarę pojawiania się nowych zagrożeń.

5. Załączniki

- Raport [NIST Cybersecurity Framework]
- Raport [Linpeas]
- Raport Nessus
- Raport Nessus - podsumowanie
- Prezentacja