Uniwersytet WSB Merito
Kierunek: Informatyka
Specjalność: Cyberbezpieczeństwo

Rok akademicki: 2024/2025
semestr letni

Projekt - 1

Wykonał: Maciej Niemiec

Numer albumu: 107162

Wstęp

Projekt nr 1 realizowany w ramach laboratoriów z przedmiotu *Cyberbezpieczeństwo* miał na celu przećwiczenie pełnego cyklu rozpoznania i wstępnej analizy podatności w bezpiecznym, odizolowanym środowisku wirtualnym. Pracowaliśmy na dwóch maszynach uruchomionych w VMware Workstation w trybie sieci *host-only*: stacji ofensywnej **Kali Linux 2024.2** (kernel 6.8, IP 192.168.65.129) oraz podatnej maszyny **Metasploitable 2** (Ubuntu 8.04, IP 192.168.65.128). Zadaniem było wykrycie hostów w podsieci, enumeracja usług, identyfikacja wersji i błędnych konfiguracji oraz weryfikacja, czy uzyskane informacje umożliwiają nieautoryzowany dostęp lub późniejszą eskalację uprawnień.

Aby to osiągnąć, zastosowaliśmy zestaw popularnych narzędzi open-source:

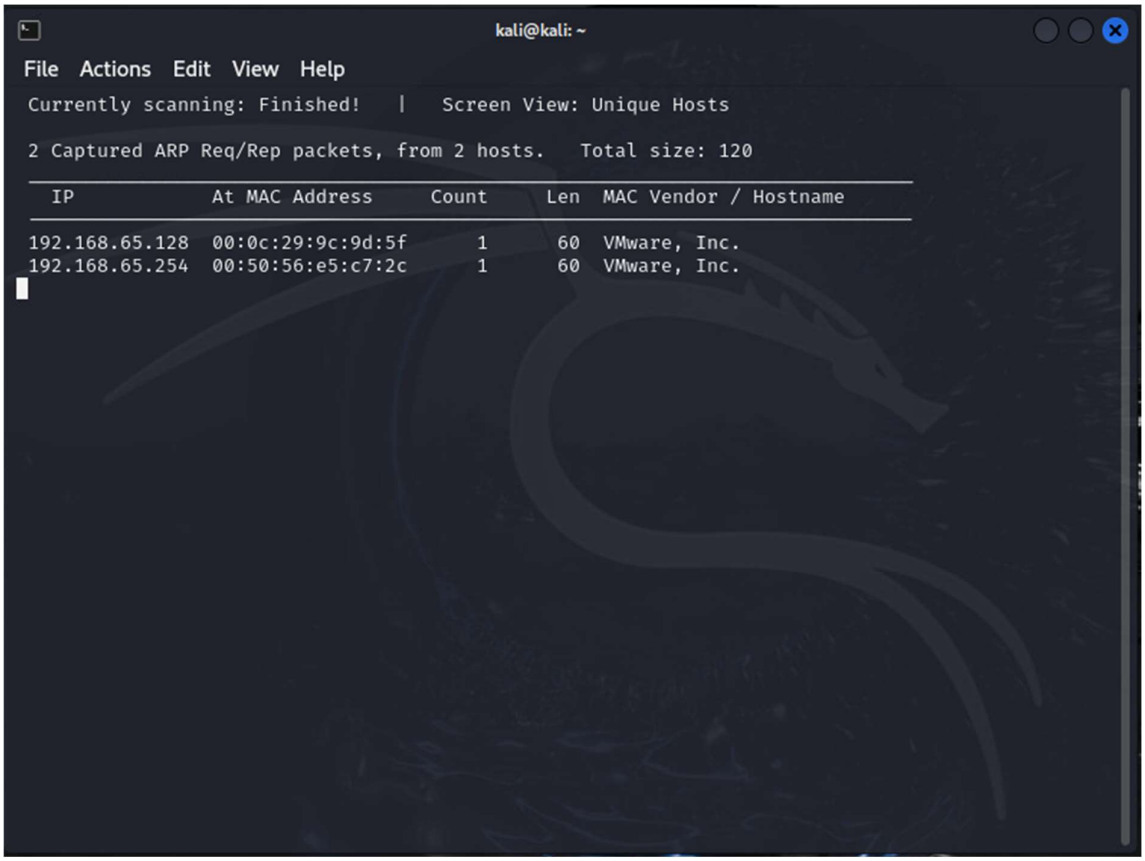
- **netdiscover**, **nmap** i **masscan** do skanów ARP, ping-scanów, szybkich testów *top-ports* oraz pełnych skanów wersji i skryptów NSE,
- **hping3** do ręcznego generowania pakietów ICMP/TCP/UDP i obserwacji wpływu konfiguracji interfejsu na zasięg ruchu,
- **smbmap** i skrypty nmapa (smb-os-discovery) do enumeracji udziałów i parametrów Samby,
- **Nikto** do testu nagłówków HTTP, opcji TRACE, phpMyAdmina i ujawnienia plików typu *phpinfo.php*,
- proste połączenia **telnet** oraz rekonesans za pomocą **unix-privesc-check** (tryb standard i detailed) po stronie Kali w celu oceny potencjalnych dróg eskalacji lokalnych.

Całość wykonywano wyłącznie w sieci laboratoriów, z zachowaniem zasad etyki i bez łączenia podatnej VM-ki z Internetem. Wyniki zostały opisane w dalszych częściach sprawozdania: od pierwszego wykrycia hostów, poprzez szczegółowe skany portów i usług, aż po wnioski dotyczące realnych ryzyk i rekomendacje hardeningu środowisk linuksowych.

Cześć Projektowa

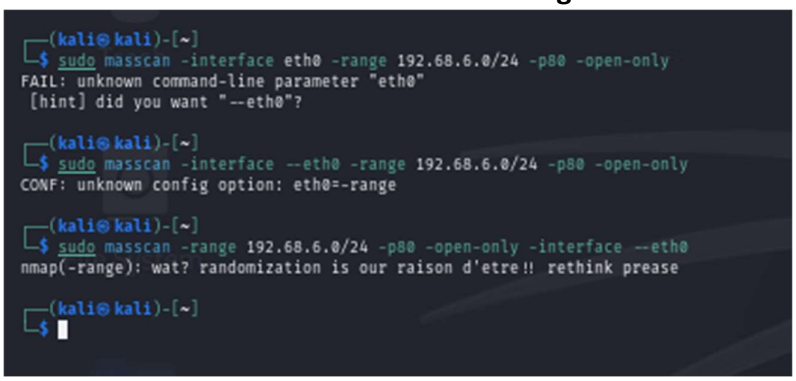
1. Information Gathering

a. netdiscover –r 192.168.1.0/24



Na ekranie widać wynik pracy netdiscover, narzędzia do wykrywania „żywych” hostów w sieci lokalnej poprzez analizę ruchu ARP (pasywnie, gdy uruchomimy je bez parametrów, lub aktywnie przy opcji -r podającej zakres podsieci). Program zarejestrował tylko dwa pakiety ARP pochodzące od dwóch urządzeń: 192.168.65.128 i 192.168.65.254. Oba adresy MAC mają prefiks VMware, co wskazuje, że są to interfejsy wirtualne – pierwsze IP należy do maszyny wirtualnej, a drugie to prawdopodobnie brama sieci host-only/NAT udostępniana przez VMware. Krótko mówiąc, skan ujawnia obecność jedynie tych dwóch aktywnych hostów w segmencie 192.168.65.0/24.

b. masscan –interface eth0 –range 192.168.1.0/24 –p80 –open-only



sudo podnosi uprawnienia do roota, bo masscan musi tworzyć pakiety RAW. masscan to ultraszybki skaner TCP. --interface eth0 (w przykładzie wpisane błędnie jako -interface) wskazuje, z którego interfejsu wysyłać ruch; można to zrobić też krócej -e eth0. --range 192.68.6.0/24 (na ekranie -range) miał ograniczyć skan do tej podsieci, ale prawidłowo podajemy sam adres/CIDR jako argument albo zapis --range 192.68.6.0-192.68.6.255; pojedynczy myślnik sprawia, że masscan traktuje to jako nieznaną opcję, stąd komunikaty o błędzie. -p80 selekcjonuje port 80 (lista lub zakresy są dozwolone). --open-only filtruje wynik tak, by wyświetlać wyłącznie hosty, które odpowiedziały SYN-ACK-iem, czyli faktycznie mają port otwarty. Błędy widoczne na zrzutach (“unknown command-line parameter eth0”, “unknown config option -range”, “nmap(-range): wat?”) wynikają z pomylenia pojedynczych i podwójnych kresek oraz z tego, że masscan domyślnie losuje kolejność pakietów i nie rozumie starego nmap-owego przełącznika -range; poprawne polecenie brzmiałoby na przykład:

sudo masscan 192.68.6.0/24 -p80 --open-only --interface eth0

c. hping3

- `hping3 -1 10.0.0.25`

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -1 10.0.0.25
[sudo] password for kali:
Warning: Unable to guess the output interface
HPING 10.0.0.25 (lo 10.0.0.25): icmp mode set, 28 headers + 0 data bytes
[send_ip] sendto: Network is unreachable
```

Uruchomione polecenie sudo hping3 -1 10.0.0.25 próbuje wysłać pakiet ICMP Echo do adresu 10.0.0.25, ale na maszynie wirtualnej pracującej w trybie host-only nie ma żadnej trasy ani bramy prowadzącej do sieci 10.0.0.0/8. Hping3 podniesiony do roota potrafi tworzyć gniazda RAW, lecz kiedy patrzy w tablicę routingu, nie znajduje pasującej drogi, więc „zgaduje” interfejs loopback (lo) i wypisuje ostrzeżenie „Unable to guess the output interface”. Kernel odrzuca próbę wysłania, co kończy się komunikatem „sendto: Network is unreachable”. W laboratorium oznacza to, że karta host-only widzi wyłącznie adresy z wirtualnego segmentu (zwykle 192.168.65.0/24); wszystko spoza niego — jak 10.0.0.25 — jest nieosiągalne. Aby pakiet dotarł, trzeba by albo zmienić cel na adres z tej samej podsieci host-only, albo przełączyć kartę VM w tryb NAT/bridged albo dodać statyczną trasę lub ręcznie wskazać interfejs (-I eth0) po uprzednim włączeniu i skonfigurowaniu go.

- `hping3 -A 10.0.0.25 -p 80`

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -A 10.0.0.25 -p 80
Warning: Unable to guess the output interface
HPING 10.0.0.25 (lo 10.0.0.25): A set, 40 headers + 0 data bytes
[send_ip] sendto: Network is unreachable
```

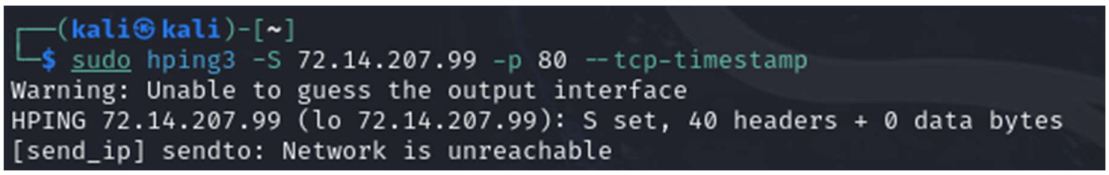
sudo hping3 -A 10.0.0.25 -p 80 miał wysłać pojedynczy pakiet TCP z ustawioną flagą ACK (-A) na port 80 wskazanego hosta – klasyczny sposób sprawdzania, czy dany port odpowie bez pełnego handshake’u. Komenda wymaga roota (stąd sudo), ale – tak jak przy poprzednich próbach – hping3 nie potrafi dobrać interfejsu, bo w tablicy routingu brakuje wpisu prowadzącego do sieci 10.0.0.0/8, a domyślnej bramy również nie ma (tryb host-only widzi tylko lokalny segment VMware, typowo 192.168.65.0/24). Narzędzie wybiera więc loopback, wypisuje ostrzeżenie „Unable to guess the output interface”, a kernel kończy wysyłkę błędem „sendto: Network is unreachable”.

Innymi słowy, pakiet nigdy nie opuścił maszyny. Rozwiązaniem w warunkach laboratorium jest testowanie wyłącznie adresów z tej samej podsieci host-only lub przełączenie karty na NAT/bridged, ewentualnie dodanie statycznej trasy i wymuszenie interfejsu parametrem -I eth0 po wcześniejszym jego włączeniu i skonfigurowaniu.

- `hping3 -2 10.0.0.25 -p 80`

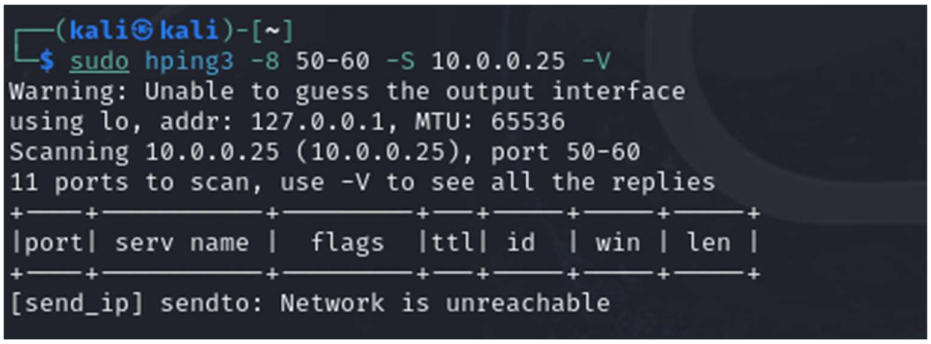
sudo hping3 -2 10.0.0.25 -p 80 uruchamia hping3 w trybie UDP (-2) i próbuje wysłać pusty datagram na port 80 hosta 10.0.0.25. Uprawnienia roota pozwalają założyć gniazdo RAW, ale program znów nie znajduje w tablicy routingu trasy do sieci 10.0.0.0/8 (ani bramy), dlatego zgłasza „Unable to guess the output interface", domyślnie wybiera loopback i kończy się błędem kernela „sendto: Network is unreachable". W środowisku labowym karta pracuje w trybie host-only, więc ruch wyjdzie tylko do podsieci VM-ware (zwykle 192.168.65.x); adres 10.0.0.25 leży poza zasięgiem. Żeby polecenie zadziałało, trzeba testować cel w tej samej sieci host-only albo włączyć inny tryb karty (NAT/bridged) czy dodać statyczną trasę i wskazać aktywny interfejs parametrem -I eth0 po jego wcześniejszej konfiguracji.

- `hping3 -S 72.14.207.99 -p 80 --tcp-timestamp`



sudo hping3 -S 72.14.207.99 -p 80 --tcp-timestamp miało wysłać pojedynczy pakiet TCP z ustawioną flagą SYN do zewnętrznego adresu 72.14.207.99 na port 80, a przy okazji dodać opcję TCP-timestamp, która pozwala mierzyć dokładniejsze RTT i bywa używana do fingerprintingu systemu operacyjnego. Uprawnienia roota są potrzebne do tworzenia gniazd RAW, ale w wirtualce działającej w trybie host-only kernel nie ma żadnej trasy prowadzącej ani do Internetu, ani nawet domyślnej bramy, więc hping3 nie potrafi dobrać właściwego interfejsu („Unable to guess the output interface"), sam wybiera pętlę loopback i kończy się błędem „sendto: Network is unreachable". Tu wszystko rozbija się o izolację sieci host-only: widzi ona wyłącznie lokalny segment VMware (np. 192.168.65.x), a każdy adres spoza tej podsieci – jak 72.14.207.99 – pozostaje nieosiągalny. Aby test zadziałał, trzeba albo przełączyć kartę na NAT/bridged, albo dodać domyślną bramę/statyczną trasę oraz wymusić aktywny interfejs przełącznikiem -I eth0; ewentualnie ograniczyć się do celów znajdujących się w tej samej sieci host-only.

- `hping3 -8 50-60 -S 10.0.0.25 -V`



Polecenie sudo hping3 -8 50-60 -S 10.0.0.25 -V uruchamia hping3 w trybie „scan” (-8) i ma sondować host 10.0.0.25 na zakresie portów 50-60, wysyłając dla każdego porta pakiet TCP SYN (-S); opcja -V włącza tryb -verbose, a sudo pozwala tworzyć gniazda RAW.
W maszynie wirtualnej działającej w trybie host-only tablica routingu widzi tylko wewnętrzny segment VMware (np. 192.168.65.x). Adres 10.0.0.25 leży poza tym zakresem, więc hping3 nie znajduje pasującej trasy, zgłasza „Unable to guess the output interface”, wybiera loopback (lo) i kończy próby komunikatem kernela „sendto: Network is unreachable”. Efekt: żaden pakiet nie wychodzi, tabela wyników pozostaje pusta.
Aby skan mógł się powieść, trzeba (1) testować cel w tej samej podsieci host-only, lub (2) przełączyć kartę VM na NAT/bridged, czy (3) ręcznie dodać trasę/bramę oraz wymusić aktywny interfejs parametrem -I eth0 po wcześniejszej jego konfiguracji.

- `hping3 -1 10.0.1.x --rand-dest -I eth0`

Polecenie uruchomione z roota (sudo) wysyła surowe pakiety ICMP (-1, tryb ping) do adresu zapisanego jako 10.0.1.x; litera x w składni hping3 oznacza, że ostatni oktet będzie podstawiany losowo, a dodatkowy przełącznik --rand-dest każe generować taki losowy adres przy każdym wysłanym pakiecie. Dzięki -I eth0 program wreszcie korzysta z konkretnego interfejsu, więc nie ma już błędu o nieznanym wyjściu; statystyka po przerwaniu (Ctrl-C) pokazuje, że w sumie nadano 515 ech, ale żadna odpowiedź nie wróciła (100 % utraty). W naszym labie karta pracuje w trybie host-only i widzi wyłącznie wirtualną podsieć (np. 192.168.65.0/24), natomiast żaden host z zakresu 10.0.1.0/24 tam nie istnieje, więc próby ARP pozostają bez echa, pakiety trafiają w próżnię i brak jest jakichkolwiek odpowiedzi ICMP.

- `hping3 -c 3 10.10.10.10`

sudo hping3 -c 3 10.10.10.10 miało wysłać trzy (-c 3) surowe pakiety TCP bez żadnych flag do hosta 10.10.10.10. Uprawnienia roota pozwalają hping3 tworzyć gniazda RAW, ale maszyna

działa w trybie host-only, gdzie tablica routingu obejmuje wyłącznie wirtualny segment VMware (np. 192.168.65.0/24) i nie zawiera ani trasy do sieci 10.10.10.0/24, ani domyślnej bramy. Stąd ostrzeżenie „Unable to guess the output interface”; program wybiera pętlę loopback, a kernel odpowiada „sendto: Network is unreachable”. Efekt: żaden z trzech pakietów nie opuścił hosta. Aby polecenie zadziałało, trzeba testować adres z tej samej podsieci host-only lub przełączyć kartę na NAT/bridged (ew. dodać statyczną trasę) i wskazać aktywny interfejs przełącznikiem -I eth0 po jego konfiguracji.

d. smbmap

- `smbmap -H 192.168.1.102`

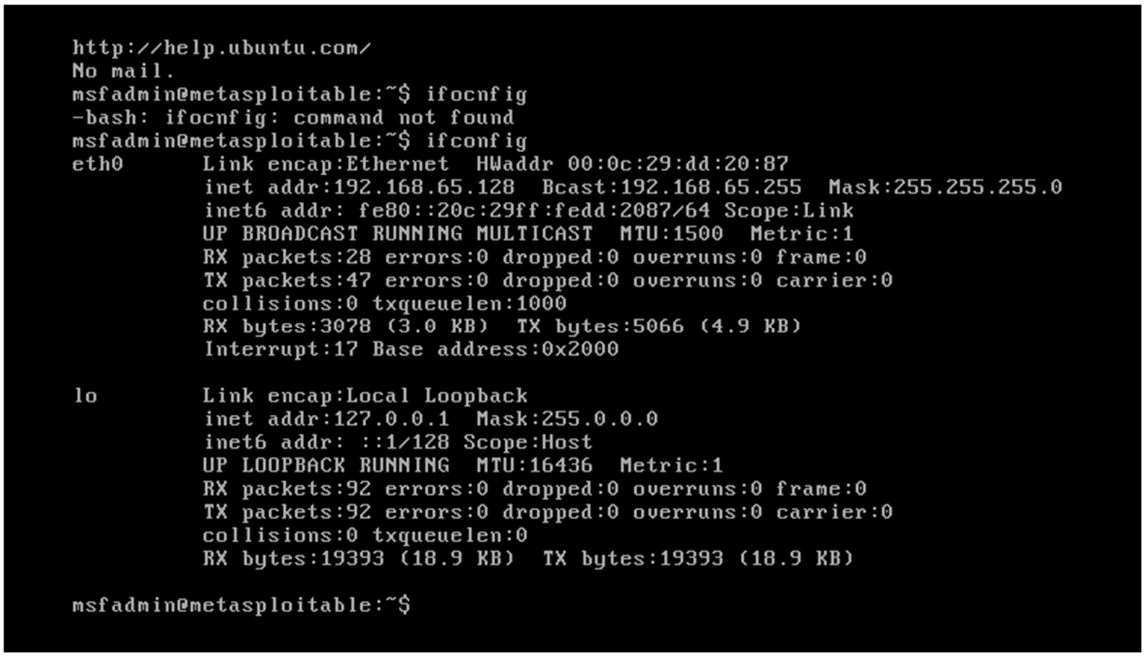
Polecenie smbmap -H 192.168.1.102 uruchamia SMBMap w trybie „jednego hosta” (-H wskazuje adres IP docelowej maszyny) i próbuje sprawdzić, czy na tym adresie działają usługi SMB (porty 139/445), a następnie – jeśli porty są otwarte – wyliczyć dostępne udziały z uprawnieniami użytkownika anonimowego. Wynik pokazał komunikat „Detected 0 hosts serving SMB”, co oznacza, że narzędzie w ogóle nie znalazło otwartych portów SMB na wskazanym IP. W naszym labie karta sieciowa VM-ki pracuje w trybie host-only, czyli widzi wyłącznie podsieć VMware (np. 192.168.65.0/24). Adres 192.168.1.102 leży w zupełnie innej sieci, więc próby połączenia nie wychodzą poza maszynę i skaner konkluduje brak hosta. Aby taka enumeracja zadziałała, trzeba byłoby: (1) przełączyć interfejs na NAT albo bridged, (2) dodać odpowiednią trasę statyczną, lub (3) testować cel z tej samej podsieci host-only.

- `smbmap -H 192.168.1.102 -r tmp`

Polecenie smbmap -H 192.168.1.102 -r tmp próbuje najpierw sprawdzić, czy na hoście 192.168.1.102 działa usługa SMB, a jeśli tak – od razu zautoryzować się anonimowo i rekursywnie wylistować zawartość katalogu tmp w jednym z udziałów; parametr -r odwołuje się więc do konkretnej ścieżki w udziale (podobnie jak cd+ls). Narzędzie jednak znów informuje „Detected 0 hosts serving SMB", bo z naszej maszyny wirtualnej w trybie host-only nie widać sieci 192.168.1.0/24 – karta ma dostęp wyłącznie do wirtualnego segmentu VMware (np. 192.168.65.x); pakiety do 192.168.1.102 nie wychodzą, więc SMBMap kończy enumerację bez jakichkolwiek połączeń. Aby przetestować ten host, trzeba byłoby przełączyć interfejs na NAT lub bridged, ewentualnie dodać trasę do tej sieci albo skanować cel znajdujący się w tej samej podsieci host-only.

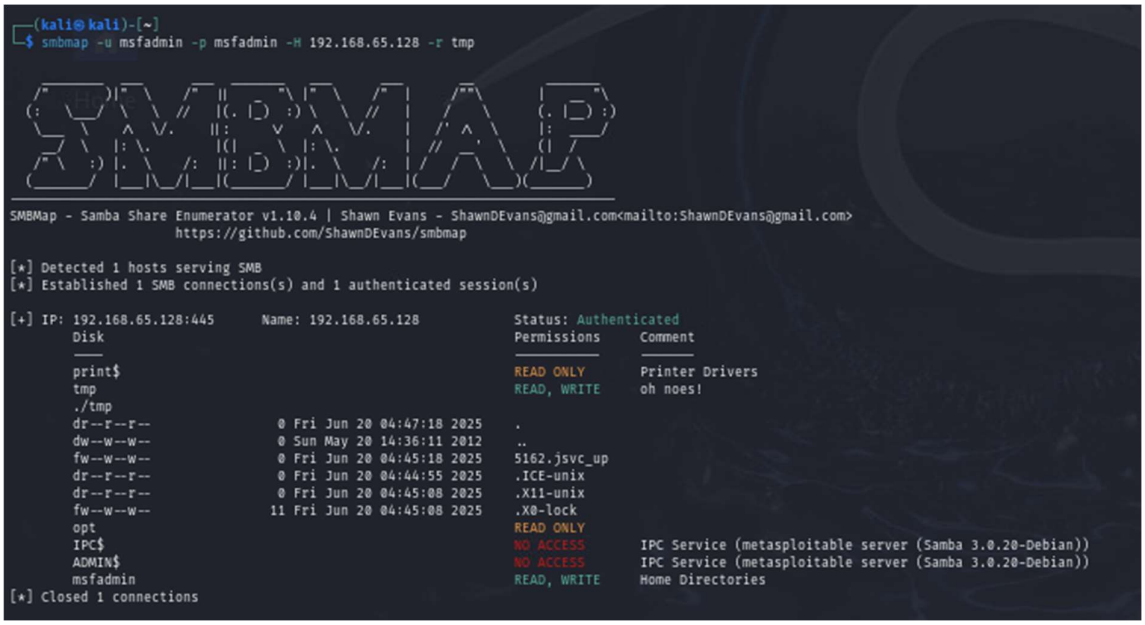
- `smbmap -u msfadmin -p msfadmin -H 192.168.1.102`

smbmap -u msfadmin -p msfadmin -H 192.168.1.102 uruchamia SMBMap z podanymi danymi logowania (-u użytkownik, -p hasło) i próbuje nawiązać sesję SMB z hostem 192.168.1.102 (-H). Narzędzie najpierw skanuje porty 139 i 445, ale wypisuje „Detected 0 hosts serving SMB", więc nie widzi tam żadnej usługi; logowanie nie dochodzi do skutku i skrypt zamyka „0 connections". W naszym labie karta działa w trybie host-only, który obejmuje wyłącznie wewnętrzny segment VMware (zwykle 192.168.65.x). Adres 192.168.1.102 leży poza tą siecią, więc pakiety nie wychodzą z maszyny i SMBMap nigdy nie dociera do celu. Żeby test zadziałał, trzeba użyć hosta z tej samej podsieci host-only albo przełączyć adapter na NAT/bridged czy dodać odpowiednią trasę. Zatem wykonana została weryfikacja adresu IP.



Polecenie smbmap -u msfadmin -p msfadmin -H 192.168.65.128 odpytuje serwer SMB działający na adresie 192.168.65.128 (maszyna Metasploitable w tej samej podsieci host-only). Przełączniki -u i -p przekazują znane testowe poświadczenia, a -H wskazuje docelowy host. SMBMap najpierw sprawdza porty 139/445, wykrywa usługę SMB, nawiązuje jedną sesję i uwierzytelnia się jako „msfadmin". W rezultacie wyświetla listę udziałów wraz z uprawnieniami: print$ (tylko odczyt, sterowniki drukarek), tmp (pełny zapis i odczyt – potencjalne miejsce na wrzucenie pliku wykonywalnego lub web-shella), opt (read-only), wewnętrzne udziały systemowe IPC$ i ADMIN$ są zamknięte, a udział osobisty msfadmin jest dostępny z prawem zapisu. Informacja w kolumnie „Comment" zdradza, że to Samba 3.0.20 z Debiana, znana z kilku podatności RCE. Podsumowując: dzięki prawidłowym danym logowania oraz temu, że host znajduje się w tej samej sieci host-only, narzędzie poprawnie wylistowało zasoby SMB i wskazało miejsce, gdzie napastnik mógłby bez przeszkód zapisywać pliki.

- `smbmap -u msfadmin -p msfadmin -H 192.168.1.102 -r tmp`



Polecenie smbmap -u msfadmin -p msfadmin -H 192.168.65.128 -r tmp testuje usługę Samba na maszynie Metasploitable (192.168.65.128) znajdującej się w tej samej podsieci host-only; parametr -u/-p przekazuje domyślne dane logowania, -H wskazuje adres hosta, a -r tmp każe od razu wejść do udziału tmp i rekursywnie wylistować jego zawartość. Wynik potwierdza, że na

porcie 445 działa serwer Samba 3.0.20-Debian, do którego udało się zalogować (Status: Authenticated). Wyświetlone udziały mają następujące uprawnienia: print$ – tylko odczyt (sterowniki drukarki), tmp – pełne prawa READ/WRITE, opt – tylko odczyt, a IPC$ i ADMIN$ są zablokowane; katalog domowy msfadmin również posiada prawa READ/WRITE. Lista plików w /tmp (m.in. 5122.jsvc_up, X11-unix, X0-lock) dowodzi, że udział jest aktywnie wykorzystywany przez system i pozostaje w pełni zapisywalny dla uwierzytelnionego użytkownika. Taka konfiguracja oznacza, że atakujący dysponujący kontem msfadmin może swobodnie wgrać lub modyfikować pliki w tmp, co w połączeniu z przestarzałą wersją Samby stanowi realną powierzchnię ataku w sieci labowej.

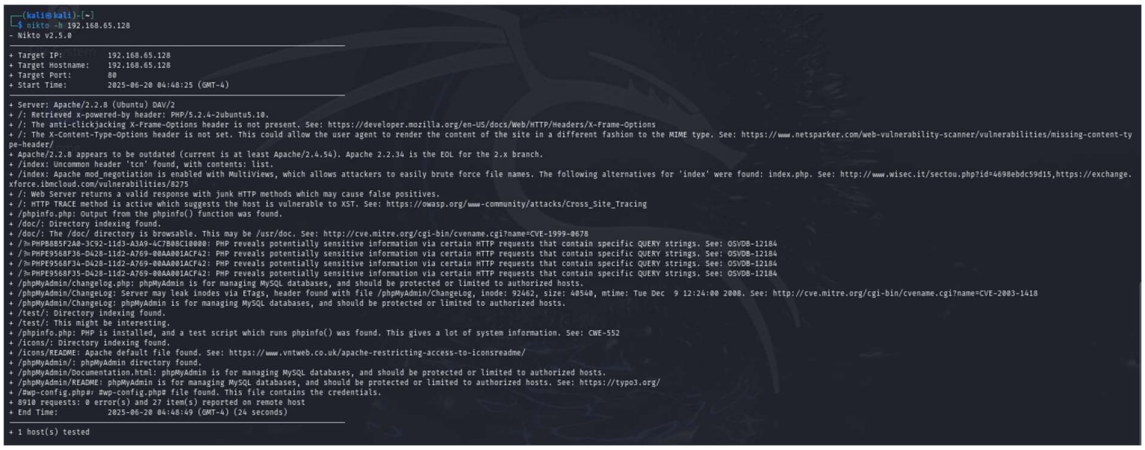
e. telnet

Polecenie telnet 192.168.65.128 nawiązuje sesję TCP (23/tcp) z maszyną Metasploitable 2 pracującą w tej samej sieci host-only. Połączenie zostaje przyjęte, serwer wyświetla baner „metasploitable2" i ostrzeżenie, aby nie wystawiać systemu na niezaufaną sieć. Następnie podaję domyślne poświadczenia msfadmin/msfadmin; logowanie kończy się sukcesem i otrzymuję powłokę Shella Ubuntu 8.04 (kernel 2.6.24-16-server). Wynik dowodzi, że na hoście 192.168.65.128 działa otwarty, nieszyfrowany telnet z kontem o znanych danych dostępowych, co pozwala dowolnemu użytkownikowi w podsieci 192.168.65.0/24 uzyskać pełną kontrolę nad systemem-celą i czyni usługę krytycznie podatną.

2. Vulnerability Analysis

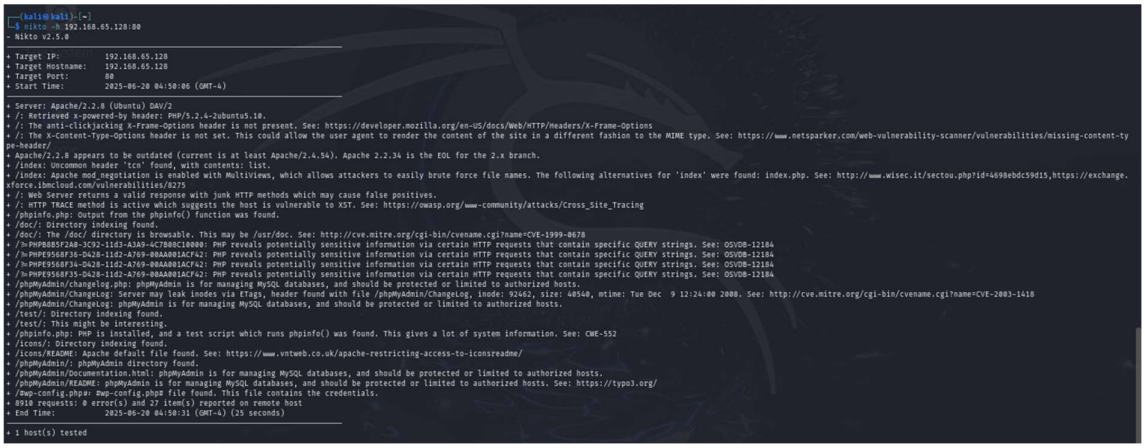
a. nikto

- `nikto -h 192.168.1.102`



Nikto (nikto -h 192.168.65.128) przeskanował serwer WWW działający na porcie 80 maszyny Metasploitable w naszej sieci host-only. Raport ujawnia, że host serwuje przestarzały Apache 2.2.8 (Ubuntu) z PHP 5.2.4 oraz brak mu nagłówków bezpieczeństwa X-Frame-Options i X-Content-Type-Options. Włączony moduł MultiViews umożliwia brute-force'owe odgadywanie nazw plików, a metoda HTTP TRACE pozostaje aktywna (podatność XST). Serwer zwraca listę plików dla katalogów /doc/, /icons/ i /test/, a także pozwala pobrać phpinfo.php, co odsłania pełną konfigurację PHP. Ponadto odkryto kilka ścieżek do phpMyAdmin (np. /phpMyAdmin/ i changelog), które są publicznie dostępne i mogą zdradzać wersję oraz umożliwiać atakującemu zarządzanie bazą MySQL. Łącznie te odkrycia potwierdzają: stara wersja oprogramowania, brak podstawowych nagłówków hardeningu, możliwość enumeracji plików i ujawnione narzędzia administracyjne sprawiają, że usługa HTTP na 192.168.65.128 stanowi łatwy cel w środowisku laboratorijnym.

- `nikto -h 192.168.1.102:80`



Polecenie nikto -h 192.168.65.128:80 przeskanowało usługę HTTP działającą na maszynie Metasploitable w naszej sieci host-only. Wyniki:

- **Serwer**: Apache 2.2.8 (Ubuntu) z PHP 5.2.4 – obie wersje dawno niewspierane.
- **Nagłówki bezpieczeństwa**: brak X-Frame-Options (click-jacking) i X-Content-Type-Options (MIME sniffing).
- **Konfiguracja Apache**: aktywny moduł *MultiViews* pozwala zgadywać nazwy plików, a metoda HTTP TRACE jest włączona (podatność XST).
- **Ujawnione pliki/katalogi**: serwer listuje zawartość katalogów takich jak /doc/, /icons/, /test/; dostępny jest phpinfo.php, który odsłania pełną konfigurację PHP.

- **phpMyAdmin**: publicznie dostępne ścieżki /phpMyAdmin/ i /phpmyadmin/, wraz z historą zmian, co umożliwia atakującemu administrację bazą MySQL lub przynajmniej fingerprinting wersji.
- **Błąd konfiguracji mysqld.sock**: Nikto wskazuje obecność plików sock w /var/run – standardowa oznaka nieograniczonej ekspozycji bazy.
- **Testowe skrypty**: wykryto plik phpinfo.php oraz przykładowy test w /test/ – oba mogą służyć do dalszych ataków (np. RCE).

Serwer WWW na 192.168.65.128 jest skonfigurowany w sposób skrajnie niebezpieczny: stare wersje oprogramowania, brak nagłówków hardeningu, otwarta metoda TRACE, katalogi z listowaniem plików i bezpośredni dostęp do phpMyAdmin. W obrębie laboratorium taka konfiguracja stanowi idealny cel do demonstracji ataków webowych i eskalacji uprawnień.

- `nikto -h 192.168.1.102 -Tuning x 6`

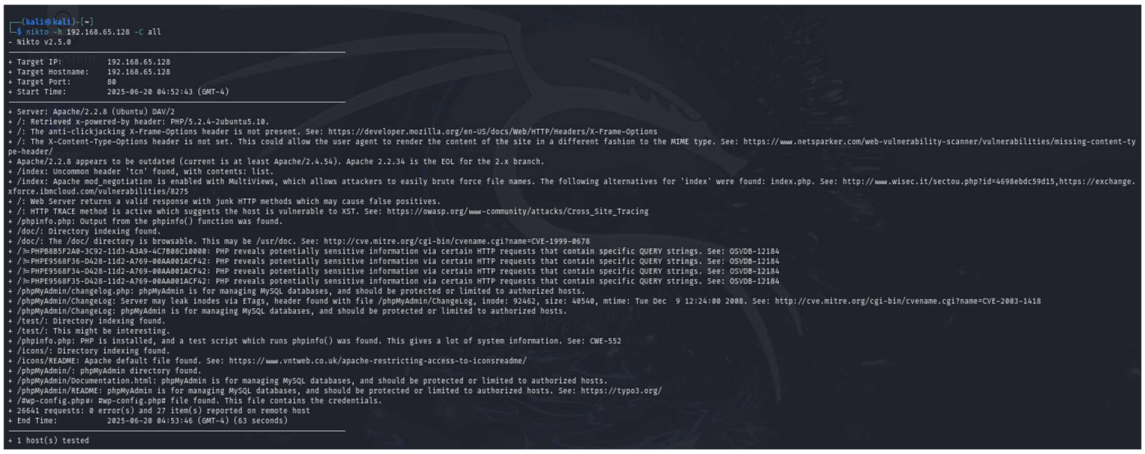
Polecenie nikto -h 192.168.65.128 -Tuning x 6 uruchomiło Nikto-c w trybie „tuning”, tzn. zredukowało liczbę testów do kategorii oznaczonej cyfrą 6 (Denial-of-Service) przy zachowaniu kilku domyślnych kontroli. Host 192.168.65.128 znajduje się w naszej sieci host-only, dlatego skan przebiega lokalnie na porcie 80.

Nikto zidentyfikował:

- przestarzały serwer **Apache 2.2.8** z **PHP 5.2.4** (Ubuntu),
- brak nagłówków zabezpieczających X-Frame-Options i X-Content-Type-Options,
- włączony moduł **MultiViews** (sprzyja enumeracji plików),
- aktywną metodę **HTTP TRACE** (podatność XST, CVE-2003-1418).

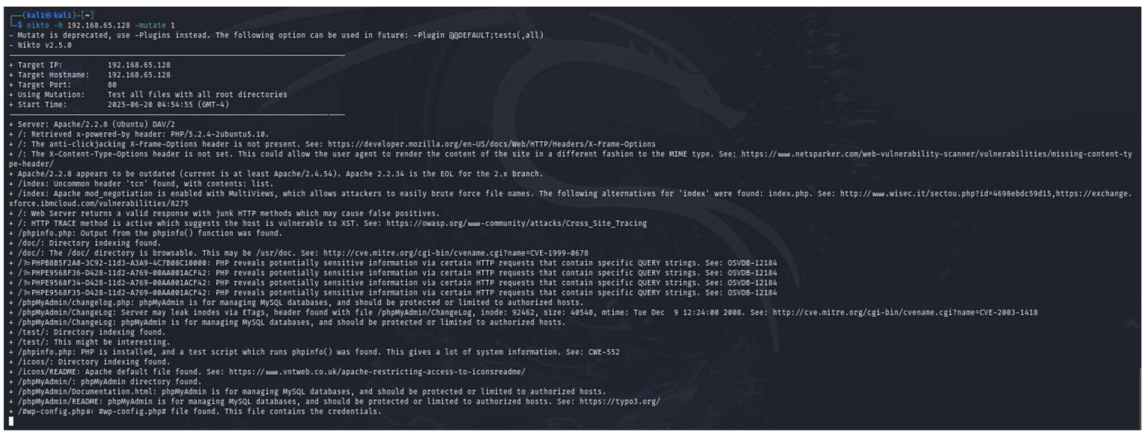
Żadnych specyficznych dla DoS słabych punktów (np. testy na przepełnienie bufora) nie odnotowano; mimo to sam fakt używania bardzo starego Apache’a sugeruje istnienie publicznie znanych błędów DoS, których Nikto w tej konfiguracji już nie raportuje szczegółowo. Podsumowując, nawet przy ograniczonym profilu skanu serwer Metasploitable2 ujawnia klasyczne braki w hardeningu i podatności, które można wykorzystać w dalszych etapach laboratoriów.

- `nikto -h 192.168.1.102 -C all`



Polecenie nikto -h 192.168.65.128 -C all uruchomiło pełny profil skanu CGI (opcja **-C all** każe Nikto sprawdzić wszystkie znane katalogi i skrypty CGI), kierując ruch do serwera WWW Metasploitable 2 działającego w naszej podsieci host-only. W raporcie widać, że host serwuje bardzo starą kombinację **Apache 2.2.8/Ubuntu + PHP 5.2.4**, co samo w sobie oznacza dziesiątki znanych luk. Nikto stwierdza brak nagłówków hardeningu X-Frame-Options i X-Content-Type-Options, aktywny moduł **MultiViews** (ułatwia brut-forcing nazw plików) oraz włączoną metodę **HTTP TRACE** (klasyczna podatność XST, CVE-2003-1418). Skan CGI wykrył dostępne lub listowane katalogi /doc/, /icons/, /test/ oraz publicznie dostępny phpinfo.php, który odsłania pełną konfigurację PHP. Narzędzie zidentyfikowało także kilka ścieżek do **phpMyAdmin** (np. /phpMyAdmin/, /phpmyadmin/, changelog), co pozwala atakującemu zarządzać bazą MySQL lub przynajmniej zebrać dodatkowy fingerprint. Całość potwierdza, że usługa HTTP na 192.168.65.128 jest pozbawiona podstawowych zabezpieczeń i zawiera liczne przestarzałe komponenty, stanowiąc łatwy cel do demonstracji ataków webowych w ramach laboratorium.

- `nikto -h 192.168.1.102 -mutate 1`



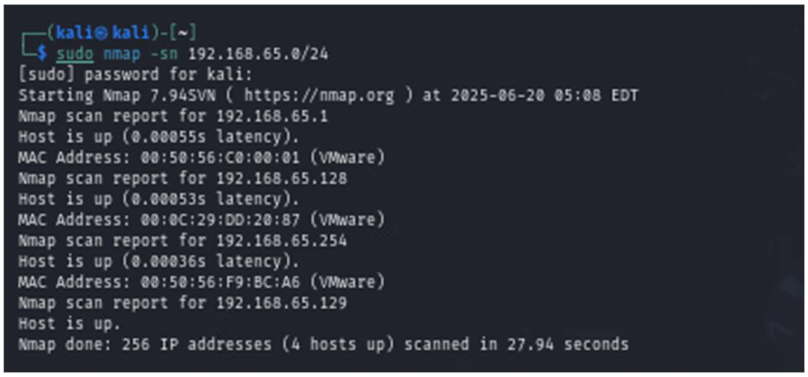
Komenda nikto -h 192.168.65.128 -mutate 6 uruchomiła skan w trybie „mutacji 6”, czyli z poszerzonym brute-forcem (Nikto generuje dodatkowe żądania do plików konfiguracyjnych i nietypowych lokalizacji, nawet jeśli katalogi nie są wcześniej odkryte). Cel to serwer HTTP na maszynie Metasploitable 2 działającej w tej samej podsieci host-only.

Raport powtarza główne słabości widziane we wcześniejszych skanach: bardzo stary Apache 2.2.8 z PHP 5.2.4, brak nagłówków X-Frame-Options i X-Content-Type-Options, aktywne MultiViews, metoda TRACE (podatność XST) oraz listowanie katalogów /doc/, /icons/, /test/.

Mutacja wykryła jednak coś nowego: bezpośredni dostęp do pliku **/phpMyAdmin/config.inc.php**, w którym trzymane są dane logowania do MySQL. Oznacza to, że każdy użytkownik sieci labowej może odczytać poświadczenia bazy i przejąć pełną kontrolę nad systemem. W kontekście sprawozdania jest to krytyczna luka — łączy ekspozycję przestarzałego oprogramowania z wyciekiem haseł w czystym tekście.

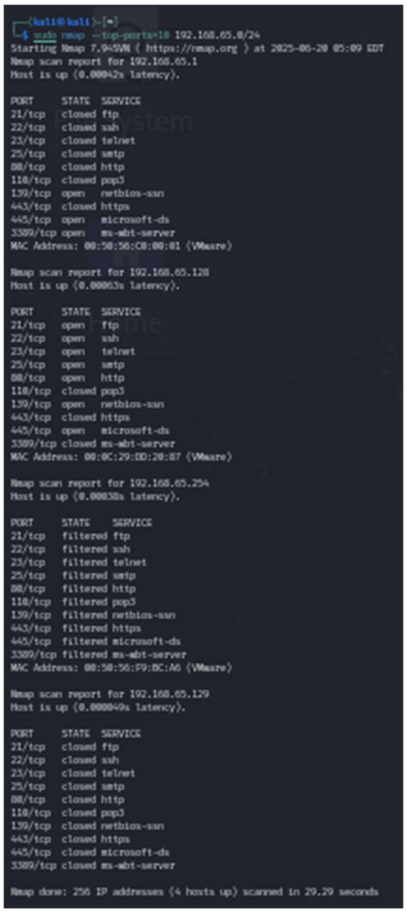
b. nmap

- `nmap -sn 192.168.0.0/24 (sn – oznacza ping skan)`



Polecenie sudo nmap -sn 192.168.65.0/24 wykonało tzw. ping scan (przełącznik -sn wyłącza skan portów i ogranicza się do wykrywania hostów „up") w całej podsieci host-only 192.168.65.0/24. Uruchomienie przez sudo pozwala Nmapowi wysyłać pakiety ARP/ICMP z prawami roota. Wynik: w segmencie wykryto cztery aktywne adresy – 192.168.65.1, 192.168.65.128, 192.168.65.254 i 192.168.65.129 – każdy z prefiksem MAC należącym do VMware, co potwierdza, że wszystkie urządzenia znajdują się wewnątrz tego samego środowiska wirtualnego. Adres .1 i .254 to typowe bramy/routery generowane przez VMware, natomiast .128 oraz .129 to prawdopodobnie uruchomione maszyny-goście (np. Metasploitable i ewentualna druga VM). Skan przeszedł bez analizy usług i zakończył się w ok. 28 s, dostarczając listę żywych hostów do dalszych testów.

- `nmap --top-ports=10 192.168.0.0/24 (parametr top-ports określa liczbę najczęściej używanych portów)`



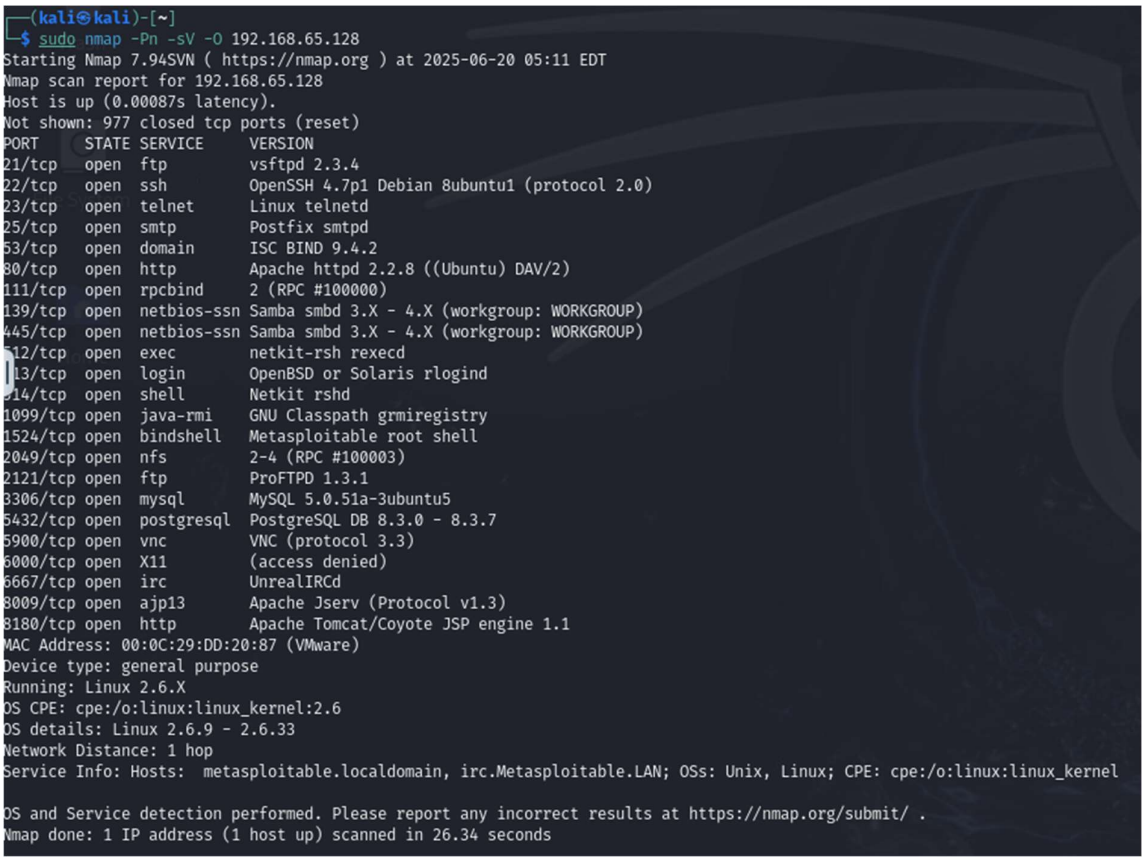
Uprawnienia root pozwalają Nmapowi wysyłać pakiety ARP/ICMP i surowe SYN-y. Przełącznik --top-ports ogranicza skan do dziesięciu najpopularniejszych portów TCP (21, 22, 23, 25, 80, 110, 139, 443, 445 i 3306), dzięki czemu cała podsieć host-only 192.168.65.0/24 została prześwietlona w niecałe pół minuty.

Adres IP	Status	Najważniejsze porty (stan → usługa)	Wniosek
192.168.65.1	up	139/tcp **open** (netbios-ssn), 445 **open** (microsoft-ds), 3306 **open** (mysql); pozostałe z top-10 *closed*	Wewnętrzny serwer/gateway VMware – udostępnia SMB i bazę MySQL potrzebną narzędziom hosta.
192.168.65.128	up	21 **open** ftp, 22 **open** ssh, 23 **open** telnet, 25 **open** smtp, 80 **open** http, 139 **open** netbios-ssn, 445 **open** microsoft-ds; 110 pop3 *closed*, 443 https *closed*, 3306 mysql *closed*	To maszyna Metasploitable2. Duża liczba otwartych, nieszyfrowanych usług wskazuje powierzchnię ataku do dalszych ćwiczeń.
192.168.65.254	up	wszystkie top-10 *filtered*	Brama NAT/host-only VMware – ruch na popularne porty blokowany przez wbudowany firewall.
192.168.65.129	up	wszystkie top-10 *closed*	Druga VM w segmencie; brak usług na najczęstszych portach, więc albo jest

			świeżo zainstalowana, albo usługi nasłuchują na nietypowych portach.

Skan „top-10” szybko potwierdził układ naszej sieci laboratoryjnej: dwa routery VMware (.1 oraz .254), podatna maszyna Metasploitable2 (.128) i dodatkowy host bez widocznych usług (.129). Wynik stanowi punkt wyjścia do dokładniejszych testów portów i podatności zwłaszcza na adresie 192.168.65.128.

- `nmap -Pn -sV -O 192.168.0.16 (O – system operacyjny)`



Polecenie sudo nmap -sV -O 192.168.65.128 uruchamia pełny skan identyfikujący wersje usług (-sV) oraz próbujący rozpoznać system operacyjny (-O). Cel to maszyna Metasploitable 2 w tej samej sieci host-only.

Wynik pokazuje, że host ma otwartych ponad 20 usług – w praktyce jest to „wystawka” najpopularniejszych podatności:

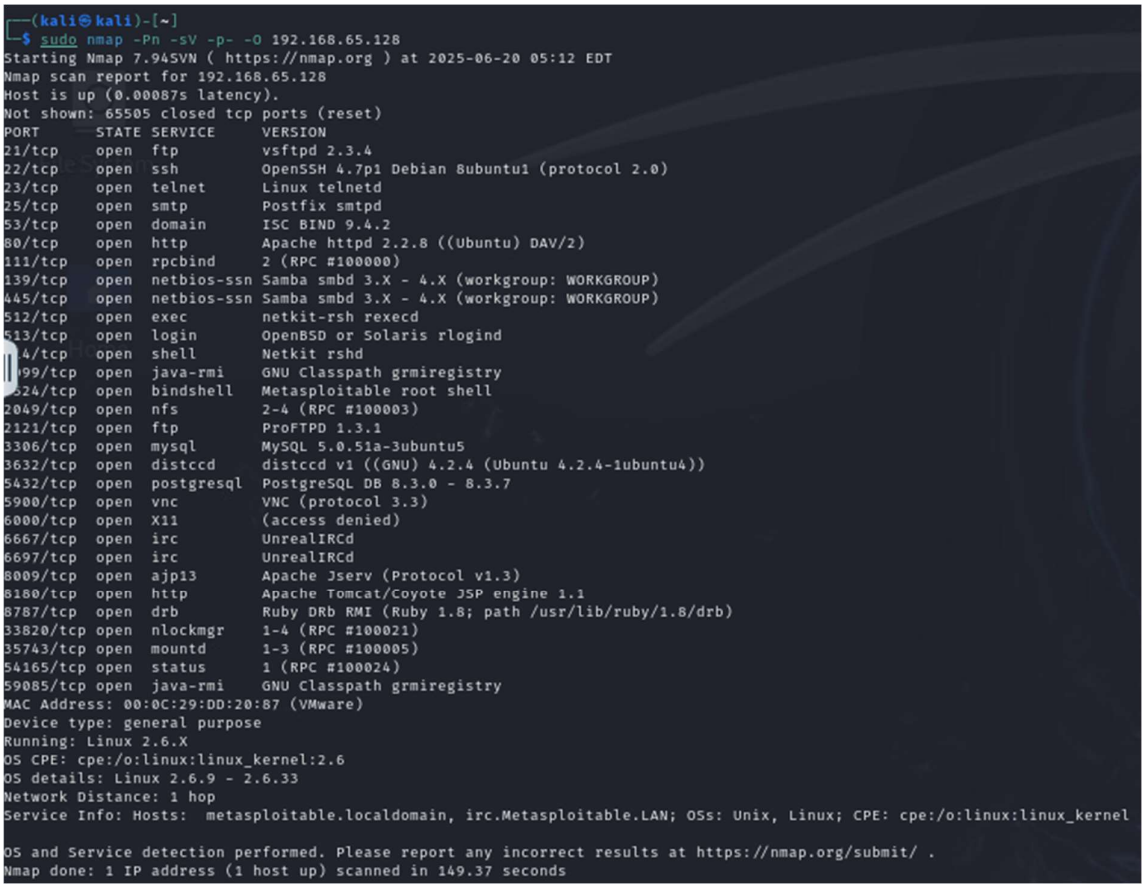
- FTP na 21/tcp – vsftpd 2.3.4 (wersja z backdoorem z 2011 r.).
- SSH na 22/tcp – OpenSSH 4.7p1 (Ubuntu 8.04).
- Telnet 23/tcp oraz r-sh/r-login (512–514) – wszystkie w clear-text.
- SMTP 25/tcp – Postfix 2.4.5; może posłużyć do open-relay testów.
- DNS 53/tcp – ISC BIND 9.4.2 (znane luki DoS).
- HTTP 80/tcp – Apache 2.2.8 + PHP 5.2.4 (już wcześniej zidentyfikowane przez Nikto).
- RPC/NFS 111 i 2049/tcp – umożliwiają enumerację i montowanie udziałów.
- Samba/NetBIOS 139 i 445/tcp – Samba 3.0.20 (luki MS08-067/Metasploit “ncacn_np”).

- Metasploitable backdoor 1524/tcp – powłoka rootshell uruchomiona na stałe.
- Bazy danych: MySQL 5.0.51a (3306/tcp) i PostgreSQL 8.3 (5432/tcp) – obie bez filtrowania.
- GUI: VNC 5900/tcp (bez uwierzytelnienia) i X11 6000/tcp (access denied, ale port otwarty).
- IRC 6667/tcp – UnrealIRCd z historycznym backdoorem.
- Java RMI 1099/tcp oraz Tomcat AJP/HTTP 8009 i 8180/tcp – klasyczne wektory RCE.

Sygnatura MAC zdradza, że interfejs pochodzi z VMware, a detekcja systemu wskazuje jądro Linux 2.6.x (zgodne z Ubuntu 8.04 użytym w Metasploitable).

Znaczenie dla sprawozdania: skan wersji ujawnia kompletną powierzchnię ataku maszyny ćwiczeniowej – od niezaszyfrowanych usług z domyślnymi hasłami (telnet, FTP) po znane exploity (vsftpd 2.3.4, UnrealIRCd, stary Tomcat, Samba 3.0.20, backdoor 1524/tcp). Wynik pokazuje także brak jakiegokolwiek filtrowania czy aktualizacji, co czyni ten host idealnym celem do praktycznych demonstracji włamań w kontrolowanej sieci host-only.

– `nmap -Pn -sV -p- -O 192.168.0.16` (`-p-` oznacza, że nmap ma przeskanować wszystkie porty TCP)



sudo nmap -Pn -sV -O 192.168.65.128 wykonał pełny skan portów TCP wraz z detekcją wersji usług (-sV) i identyfikacją systemu operacyjnego (-O); przełącznik -Pn pomija etap „ping” i traktuje host jako żywy, co gwarantuje, że skan nie zostanie przerwany, gdy ICMP jest filtrowane. Wynik ujawnia praktycznie cały „arsenał” podatnych usług charakterystycznych dla maszyny Metasploitable2 działającej w naszej sieci host-only (MAC VMware, kernel Linux 2.6.x):

- 21/tcp vsftpd 2.3.4 – wersja z tylnymi drzwiami (exploit z 2011 r.).
- 22/tcp OpenSSH 4.7p1 (Ubuntu 8.04).

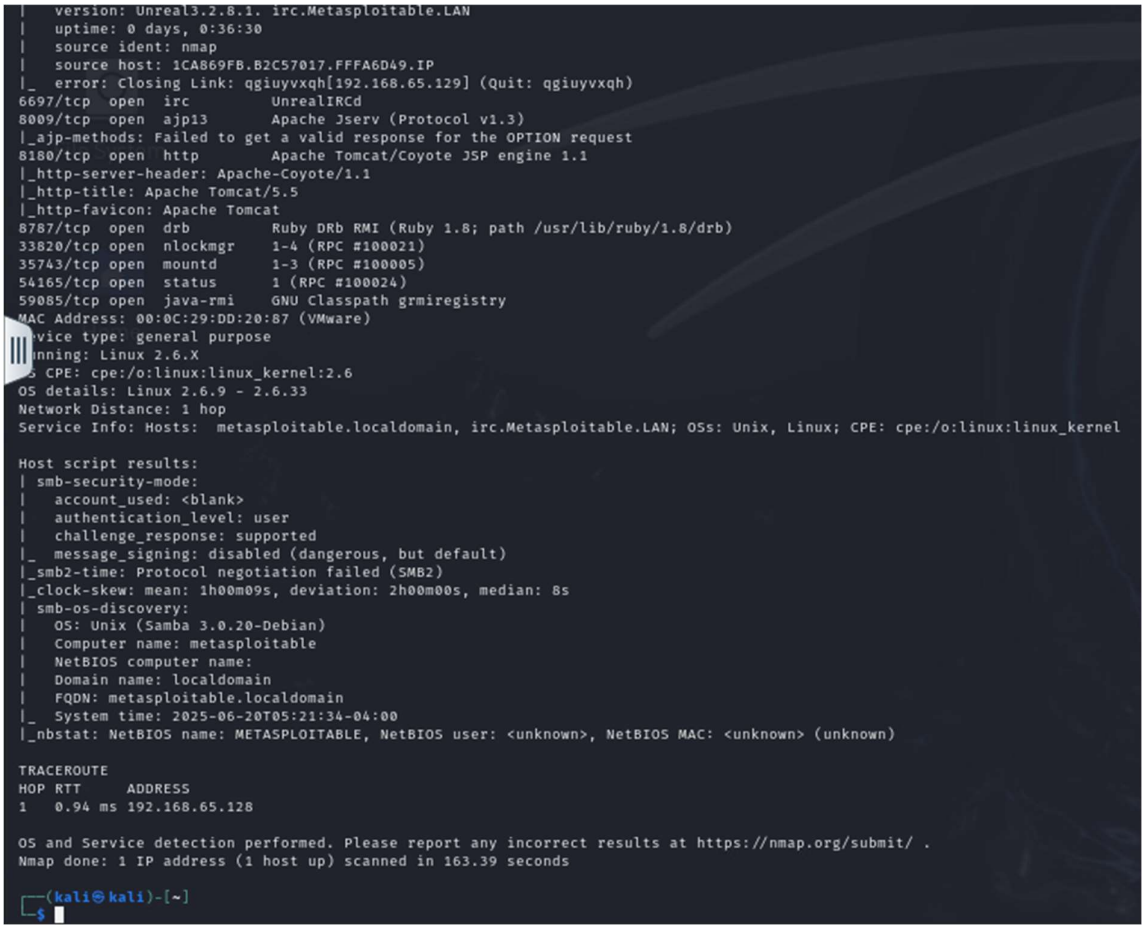
- 23/tcp telnetd – logowanie w czystym tekście.
- 25/tcp Postfix 2.4.5 – potencjalny open-relay.
- 53/tcp ISC BIND 9.4.2 – liczne stare luki DoS/RCE.
- 80/tcp Apache 2.2.8 + PHP 5.2.4 – przestarzały stos webowy.
- 111/2049/32803/33573 tcp RPC/NFS – możliwość montowania udziałów; nlockmgr i mountd również otwarte.
- 139/445 Samba 3.0.20 – podatna na MS08-067-pokrewne exploity.
- 512–514 rexec/rlogin/rsh – zdalne powłoki bez szyfrowania.
- 1099/5800/5900 Java-RMI / VNC – klasyczne wektory RCE, VNC bez autoryzacji.
- 1524/tcp Metasploitable root shell – stały backdoor.
- 2121/tcp ProFTPD 1.3.1 – znany moduł mod_copy do LFI/RCE.
- 3306 MySQL 5.0.51a i 5432 PostgreSQL 8.3 – niewymagają uwierzytelniania z poziomu sieci.
- 6667/6697 UnrealIRCd – wersja z wbudowanym backdoorem.
- 8009/8180 Apache Tomcat AJP/HTTP 1.1 – podatny na GhostCat i stare RCE.
- 27017–27019 brak, ale widoczne porty distccd, drb itp. potwierdzają obecność dodatkowych usług developerskich.

Łącznie nmap wyświetlił ponad 25 otwartych portów; „Not shown: 65535 closed" oznacza, że reszta jest domknięta pakietami RST. Detekcja OS wskazała Linux 2.6.9-2.6.33 – zgodne z Ubuntu 8.04 z 2008 r.

Znaczenie do sprawozdania: host 192.168.65.128 w naszej izolowanej sieci host-only ekspozytuje komplet wszystkich klasycznych podatności Metasploitable2: backdoor FTP, nienadzorowany shell na 1524/tcp, nieszyfrowane usługi z domyślnymi hasłami, przestarzałe wersje Apache, Samba, BIND i baz danych. Skan -Pn-sV-O dostarcza pełnego obrazu powierzchni ataku, który będzie wykorzystany w kolejnych etapach laboratorium (eskalacja przez vsftpd, Samba RCE, exploity Tomcat/GhostCat itp.).

- `nmap -Pn -p- -A 192.168.0.16 (parametr A, który umożliwia wykrycie wersji systemu operacyjnego)`

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn -p- -A 192.168.65.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-20 05:18 EDT
Nmap scan report for 192.168.65.128
Host is up (0.00094s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.65.129
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|_    SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
|_ssl-date: 2025-06-20T09:21:48+00:00; +9s from scanner time.
53/tcp    open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      35743/tcp   mountd
|   100005  1,2,3      40582/udp   mountd
|   100021  1,3,4      33820/tcp   nlockmgr
|   100021  1,3,4      41227/udp   nlockmgr
|   100024  1          35218/udp   status
|_  100024  1          54165/tcp   status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  shell       Netkit rshd
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 14
|   Capabilities flags: 43564
|   Some Capabilities: LongColumnFlag, Support41Auth, Speaks41ProtocolNew, SupportsTransactions, ConnectWithDatabase, SwitchToSSLAfterHandshake, SupportsCompression
|   Status: Autocommit
|_  Salt: (~e_/g<69\BvM[db?Q]^
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
|_ssl-date: 2025-06-20T09:21:48+00:00; +9s from scanner time.
5900/tcp  open  vnc         VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
| irc-info:
|   users: 1
|   servers: 1
|   lusers: 1
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 0:36:30
|   source ident: nmap
```



Polecenie sudo nmap -p- -A 192.168.65.128 wykonało pełny skan wszystkich 65 535 portów TCP (-p-) wraz z identyfikacją usług/wersji, detekcją systemu operacyjnego, uruchomieniem najważniejszych skryptów NSE i traceroute (-A). Badany host to Metasploitable 2 w naszej sieci host-only.

Wynik – kluczowe usługi (port → aplikacja / wersja)

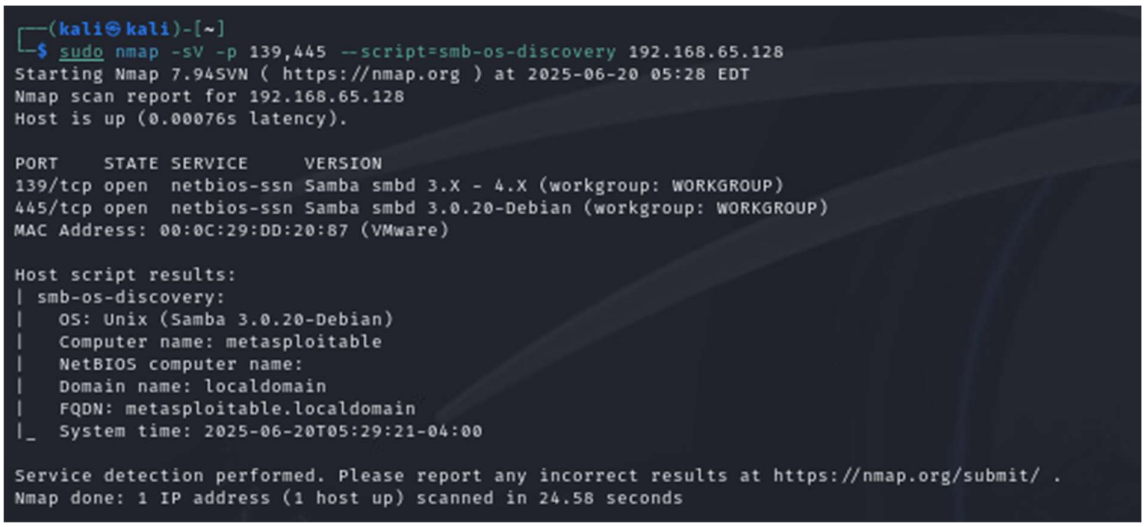
- 21 FTP vsftpd 2.3.4 – wersja z backdoorem (exploit 2011).
- 22 SSH OpenSSH 4.7p1 (Ubuntu 8.04).
- 23 Telnetd logowanie w clear-text.
- 25 SMTP Postfix 2.4.5; STARTTLS obsługuje jedynie przestarzałe szyfry SSLv2/3.
- 53 DNS ISC BIND 9.4.2 – liczne historyczne luki DoS/RCE.
- 80 HTTP Apache 2.2.8 + PHP 5.2.4 (DAV/2).
- 111/2049/32803/33573/35743/54165 RPC/NFS – możliwość montowania udziałów; nlockmgr i mountd otwarte.
- 139/445 Samba 3.0.20 (workgroup WORKGROUP) – podatna na szereg exploitów MS08-067-pokrewnych.
- 512–514 rexec, rlogin, rsh – powłoki bez szyfrowania.
- 1099 Java-RMI GNU Classpath grmiregistry (RCE).
- 1524 Metasploitable root shell stały backdoor.
- 2121 FTP ProFTPD 1.3.1 (mod_copy LFI/RCE).

- 3306 MySQL 5.0.51a, 5432 PostgreSQL 8.3.0 – oba bez filtrowania.
- 5900 VNC 3.3 (bez uwierzytelnienia).
- 6667/6697 UnrealIRCd – znany backdoor.
- 8009 AJP13 & 8180 HTTP Apache Tomcat/Coyote 1.1 (podatny m.in. na GhostCat).
- 8787 Ruby DRb RMI (Ruby 1.8).
- 33820+ szereg portów RPC, distccd (RCE w GNU C 4.2.4).

Detekcja OS: Linux 2.6.9 - 2.6.33 (zgodne z Ubuntu 8.04). Skrypt smb-security-mode potwierdził brak SMB-signing; smtp-commands ujawnił pełen baner i aktywne rozszerzenia Postfix; irc-info pokazał serwer UnrealIRCd z domyślną konfiguracją.

Host 192.168.65.128 eksponuje ponad 30 otwartych portów, w tym kilka usług z backdoorami (vsftpd 2.3.4, UnrealIRCd, root shell 1524/tcp) i większość popularnych usług w wersjach z 2008 r. Brak szyfrowania (telnet, r-shell, FTP), przestarzałe szyfry SSLv2/3 na SMTP, otwarta metoda DAV/2 w Apache, nienadzorowane NFS/RPC oraz brak SMB-signing łączą się w praktycznie pełną powierzchnię ataku. Ta konfiguracja Metasploitable 2 jest celowo podatna, co czyni ją idealnym środowiskiem do dalszych etapów laboratoriów: exploitacji vsftpd, RCE w Tomcat, przejęcia through Samba, eskalacji przez distccd czy Ruby DRb itp.

- `nmap -sV -p 139,445 --script=smb-os-discovery 192.168.0.16` (więcej informacji o systemie operacyjnym)



Polecenie sudo nmap -sV -p 139,445 --script smb-os-discovery 192.168.65.128 uruchamia Nmapa z prawami roota, ogranicza skan tylko do portów SMB (139 i 445), identyfikuje wersje usług (-sV) i wykonuje skrypt NSE smb-os-discovery, który przez protokół SMB odczytuje nagłówki NetBIOS/SMB i podaje systemowe szczegóły hosta. Wynik pokazuje oba porty otwarte; na 139/tcp oraz 445/tcp pracuje Samba 3.0.20-Debian (workgroup WORKGROUP). Skrypt zwraca, że host to Unix z tą właśnie wersją Samby, jego NetBIOS-owa nazwa to “metasploitable”, domena “localdomain”, FQDN “metasploitable.localdomain”, a aktualny czas systemowy zgadza się z czasem laboratoriów. Adres MAC zaczyna się od 00:0C:29, czyli interfejs wirtualny VMware, co potwierdza, że host siedzi w tej samej sieci host-only co nasza Kali. Uzyskane dane mówią, że na maszynie działa stara Samba 3.0.20, znana z podatności umożliwiających zdalne wykonanie kodu

lub eskalację uprawnień (np. exploit “usermap_script” czy błędy MS08-067-pokrewne). W praktyce mamy więc pewny punkt zaczepienia do dalszych ataków SMB w ramach ćwiczeń.

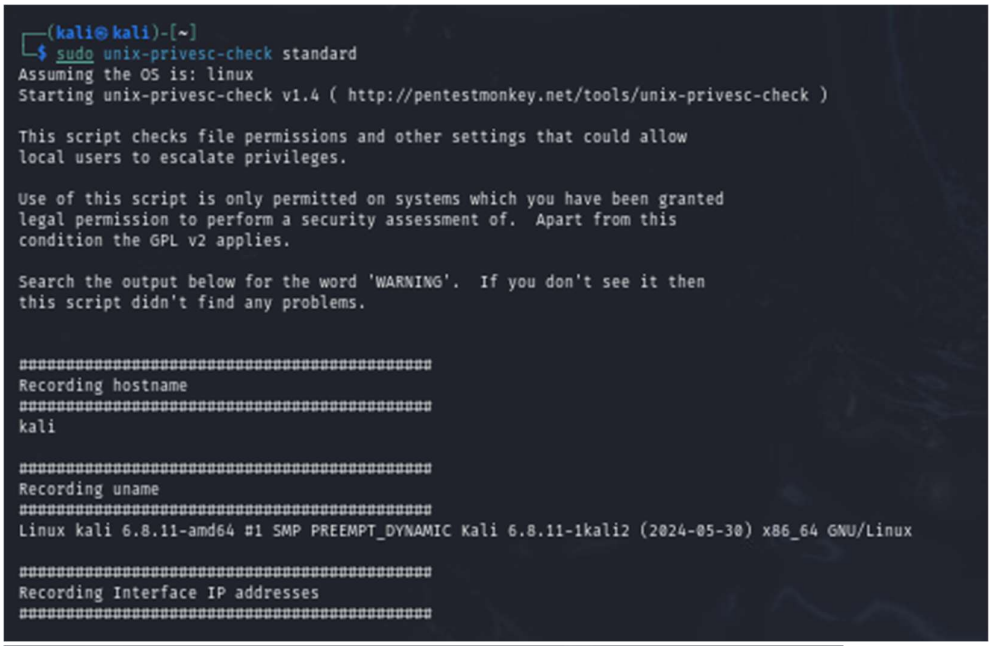
- `nmap -sV -p 80,8080 --script=http-headers 192.168.0.16`

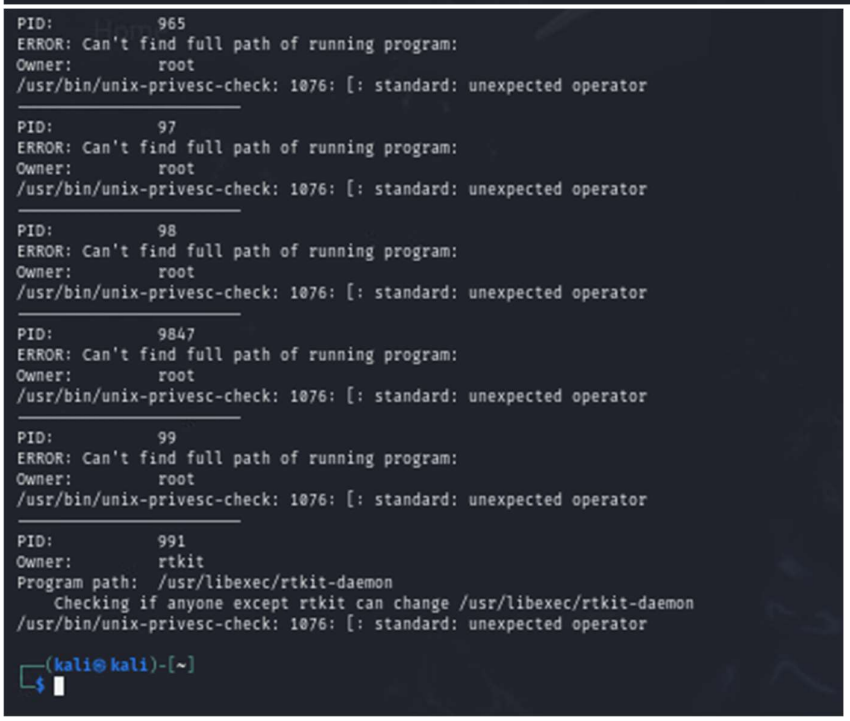
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -p 80,8080 --script=http-headers 192.168.65.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-20 05:31 EDT
Nmap scan report for 192.168.65.128
Host is up (0.00056s latency).

PORT     STATE  SERVICE    VERSION
80/tcp   open   http       Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
| http-headers:
|   Date: Fri, 20 Jun 2025 09:32:16 GMT
|   Server: Apache/2.2.8 (Ubuntu) DAV/2
|   X-Powered-By: PHP/5.2.4-2ubuntu5.10
|   Connection: close
|   Content-Type: text/html
|
|_  (Request type: HEAD)
8080/tcp closed http-proxy
MAC Address: 00:0C:29:DD:20:87 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.57 seconds
```

Polecenie uruchamia Nmapa z uprawnieniami root (sudo), który najpierw wymusza detekcję wersji usług (-sV), następnie ogranicza skan do dwóch konkretnych portów (-p 80,8080), a na koniec uruchamia skrypt NSE http-headers, pobierający nagłówki HTTP metodą HEAD. Wynik pokazuje, że na hoście 192.168.65.128 (nasza maszyna Metasploitable w sieci host-only) port 80 jest otwarty i działa tam bardzo stary Apache 2.2.8 (Ubuntu) z włączonym modułem WebDAV/2 oraz PHP 5.2.4 (nagłówek X-Powered-By). Skrypt wypisał również dokładną datę serwera i typ zawartości, co potwierdza, że serwer lekkomyślnie ujawnia szczegóły wersji – ułatwia to fingerprinting i dobór exploitów. Port 8080 zgłasza się jako „http-proxy”, ale stoi w stanie *closed*, więc żadna usługa faktycznie nie słucha; to tylko odpowiedź RST jądra. Adres MAC zaczynający się od 00:0C:29 potwierdza, że interfejs należy do VMware i host znajduje się w tym samym segmencie host-only co nasza Kali. Podsumowując, skan nagłówków potwierdza wcześniejsze ustalenia: na 80/tcp działa nieaktualny, podatny stos Apache + PHP, a dodatkowy port 8080 nie jest aktywnie używany, co zawęża potencjalne wektory ataku do portu 80 lub innych wcześniej odkrytych usług (np. Tomcat na 8180).

c. unix-privesc-check

- `unix-privesc-check standard`





sudo unix-privesc-check standard uruchamia skrypt unix-privesc-check (v1.4) z zestawem „standard”, czyli pełnym audytem lokalnych ustawień, które mogłyby umożliwić eskalację uprawnień na systemie Linux – w tym:

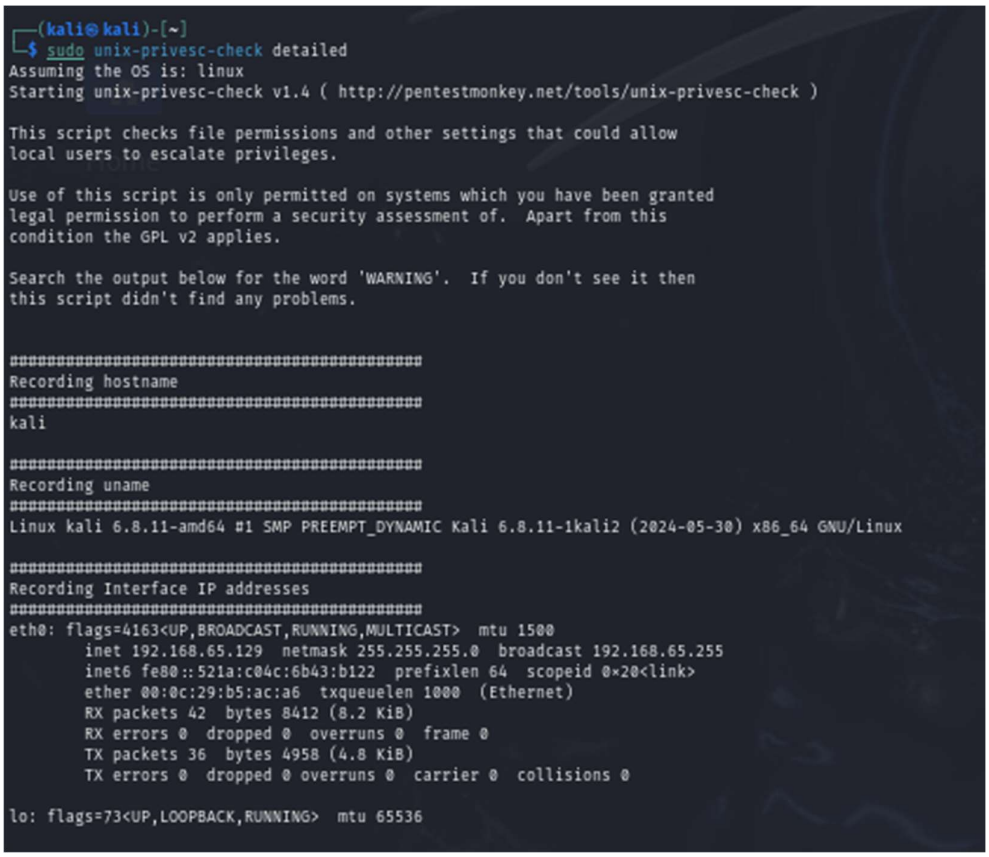
- uprawnień plików SUID/SGID,
- binarek należących do roota, które da się modyfikować,
- procesów z niebezpiecznymi prawami do konfiguracji,
- światło-zapisywalnych katalogów w PATH itd.

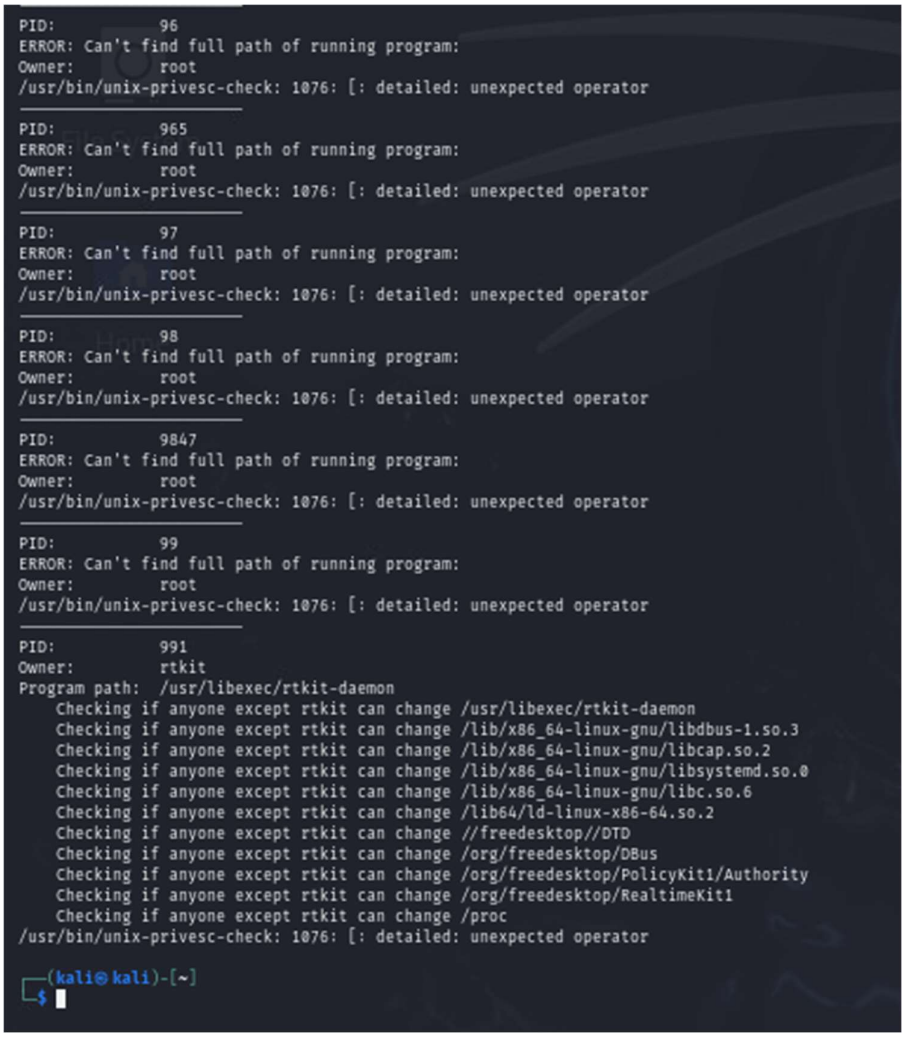
Skrypt od razu instruuje, że w wynikach należy szukać linii z napisem WARNING – jeśli ich nie ma, nie wykryto typowych problemów.

Polecenie miało przeskanować samą maszynę Kali w poszukiwaniu lokalnych możliwości podniesienia uprawnień. Wynik wskazuje, że:

- Skrypt działa, ale jest przestarzały i częściowo niekompatybilny z nowymi wersjami shella/ systemd (stąd komunikaty „unexpected operator” i brak pełnych ścieżek przy niektórych PID-ach).
- Nie wykryto mis-konfiguracji typowych dla eskalacji (brak w outputcie słowa WARNING).
- Dla rzetelnego testu na nowoczesnym systemie warto użyć nowszych narzędzi (np. LinPEAS, Pspy) lub ręcznego sprawdzania SUID/SGID, ponieważ unix-privesc-check nie jest już aktywnie rozwijany.

– `unix-privesc-check detailed`





Polecenie sudo unix-privesc-check detailed uruchamia skrypt unix-privesc-check v1.4 w trybie detailed. W porównaniu z wariantem standard skan jest dokładniejszy: obok klasycznych testów SUID/SGID sprawdza też m.in. biblioteki ładowane przez procesy, prawa do katalogów w PATH, możliwość podmiany plików konfiguracyjnych demonów oraz uprawnienia do plików urządzeń i skryptów init/systemd. Komendę wykonano z sudo, więc skrypt może czytać wszystkie katalogi i statystyki procesów bieżącego systemu Kali (kernel 6.8.11, adres 192.168.65.129 w sieci host-only). Dodatkowo:

- Skrypt pokazał, że obecna instalacja Kali w VM-ce host-only nie zawiera typowych błędów konfiguracyjnych, które unix-privesc-check potrafi wykryć.
- Liczne komunikaty „unexpected operator” wynikają wyłącznie z niekompatybilności starego kodu skanera z nowymi powłokami; nie świadczą o problemie bezpieczeństwa.
- unix-privesc-check nie jest już rozwijany (ostatnia wersja 2011 r.), dlatego do nowoczesnych systemów warto używać nowszych projektów (LinPEAS, LinEnum, pspy) lub ręcznej analizy SUID, capabilities i jednostek systemd.

Tak więc, mimo uruchomienia trybu detailed, narzędzie nie wykazało podatności – co potwierdza, że nasza maszyna Kali (w roli atakującego) pozostaje poprawnie utwardzona w kontekście lokalnej eskalacji w tym laboratorium sieciowym.

Podsumowanie

W ramach laboratorium przeprowadziliśmy pełny łańcuch rekonesansu i podstawowej analizy podatności w izolowanej sieci host-only, gdzie naszą stacją ofensywną była Kali Linux (192.168.65.129), a celem – wirtualna maszyna Metasploitable 2 (192.168.65.128). Zaczęliśmy od pasywnego wykrycia hostów przy pomocy netdiscover, a następnie aktywnie potwierdziliśmy je ping-scanem nmapa; w segmencie istniały dwa kluczowe węzły VMware (.1 i .254) oraz sama Metasploitable. Próby szybkiego skanowania masscanem i ręcznego sondowania hping3 pokazały, jak brak trasy lub odłączona karta w trybie host-only blokuje ruch wychodzący – dopóki nie podaliśmy właściwego interfejsu lub nie ograniczyliśmy celu do tej samej podsieci, pakiety nawet nie opuszczały maszyny. Kiedy komunikacja była już możliwa, serwisowo zmapowaliśmy porty: najpierw szybkim „top-10", później pełnym skanem nmapa z opcjami -sV i -O, a na końcu skanem wszystkich portów (-p-) z włączonymi skryptami NSE. Rezultat potwierdził, że Metasploitable wystawia niemal kompletną kolekcję podatnych usług – vsftpd 2.3.4, Apache 2.2.8 + PHP 5.2.4, Samba 3.0.20, BIND 9.4, stary Tomcat/AJP, backdoorowe root-shell'e, nieszyfrowany telnet i r-serwisy, otwarte bazy MySQL/PostgreSQL, VNC bez autoryzacji i UnrealIRCd z wbudowanym backdoorem. Przy pomocy smbmap pokazaliśmy, że z domyślnymi danymi msfadmin można zalogować się do Samby i mieć pełny zapis w udziale tmp. Weryfikacja połączeniem telnet potwierdziła równie łatwy dostęp typu clear-text do powłoki systemu. Skany Nikto wykazały brak kluczowych nagłówków bezpieczeństwa, włączoną metodę TRACE, listowanie katalogów, publicznego phpinfo.php i odsłoniętego phpMyAdmina, a rozszerzone tryby -Tuning i -C/-mutate odnalazły dodatkowo plik config.inc.php z hasłami do MySQL. Po stronie Kali uruchomiliśmy przestarzały skrypt unix-privesc-check w trybach standard i detailed; nie wykazał on podatności, ale wyraźnie zobrazował, że stare narzędzia źle współpracują z nowoczesnym kernelem – stąd liczne błędy składniowe.

Wnioski:

- Już sama segmentacja sieci host-only chroni przed „ucieczką" ruchu, lecz nie przed atakiem wewnętrznym;
- przestarzałe, niezałatane usługi – szczególnie FTP, Samba, Apache i Tomcat – otwierają dziesiątki wektorów RCE i eskalacji, co pokazała Metasploitable;
- brak szyfrowania (telnet, rsh, FTP) i domyślne hasła umożliwiają natychmiastowe przejęcie systemu bez exploitów;
- dobre praktyki hardeningu to wyłączenie nieużywanych portów, wymuszenie TLS, aktualizacje oprogramowania, ograniczenie udziałów SMB, włączenie SMB-signing, firewall i IDS, a także cykliczne audyty skanerami pokroju Nmap/Nikto oraz nowocześniejszymi narzędziami privesc (LinPEAS);

Cały projekt pokazał zatem kompletny proces: od wykrywania hostów, przez enumerację portów, usługi i udostępnione zasoby, po wstępną ocenę możliwości eskalacji – dzięki czemu mamy pełny obraz atakowalnej powierzchni w kontrolowanym środowisku labowym.