KARTA PRACY UCZNIA | 4 × 45 MINUT

Modele uprawnień i architektura systemu pod kątem typowych ataków

Ubuntu Linux • wirtualizacja • ochrona danych • reakcja na zagrożenia • odzyskiwanie plików

TWOJA MISJA

Jesteś administratorem szkolnego laboratorium. Sprawdzisz, dlaczego jeden użytkownik może czytać plik, a inny otrzyma odmowę dostępu. Zobaczysz też, jak zareaguje Windows Defender na próbkę testową i jak odzyskać dane po kontrolowanej symulacji incydentu.

Plan na cztery godziny lekcyjne

LEKCJA	CZYM SIĘ ZAJMUJEMY	CO BĘDZIESZ UMIEĆ
01 / 45 min	Architektura Linux, r / w / x, liczby 4–2–1	Odczytywać i obliczać 755, 644, 700 i 600
02 / 45 min	chmod, skrypty, konta, katalogi, umask	Wykazać rzeczywistą odmowę dostępu
03 / 45 min	Typowe ataki, AppArmor, Defender, EICAR	Rozpoznawać różne warstwy ochrony
04 / 45 min	Pliki testowe, archiwum, migawka VM	Wykonać kopię i sprawdzić odtworzenie

Imię i nazwisko: ..... **Klasa:** .....

Nazwa maszyny wirtualnej: .....

ZASADY LABORATORIUM

Pracuj tylko na szkolnej VM i plikach testowych w katalogu **~/TC_LAB**. Nie wykorzystuj prawdziwych danych uczniów. Nie wyłączaj Defendera, AppArmor ani innych zabezpieczeń. Nie udostępniaj katalogów VM komputerowi gospodarza. Kiedy zadanie wymaga uprawnień administratora lub przywrócenia migawki, wykonaj je za zgodą nauczyciela.

Ocena jest wystawiana na podstawie **samodzielnej pracy, wyjaśnień i obserwacji**; zadania nie mają punktacji.

TEORIA | PIERWSZA LEKCJA

01 Kto decyduje o dostępie w Linux?

W Linux działa wiele procesów, ale żaden program nie powinien zyskiwać więcej praw, niż potrzebuje. Dostęp do plików zależy m.in. od **tożsamości procesu**, uprawnień zasobu i dodatkowych reguł bezpieczeństwa.

WARSTWA	PRZYKŁAD	DLACZEGO TO WAŻNE?
Użytkownik	whoami, id, groups	Program zwykle działa z prawami użytkownika, który go uruchomił.
Plik / katalog	chmod, właściciel, grupa	Określa, kto może czytać, pisać i wykonywać.
Jądro Linux	system calls, procesy	Weryfikuje operacje i odmawia niedozwolonych.
AppArmor (MAC)	profile programów	Może ograniczyć program dodatkowo, niezależnie od chmod.
Ochrona AV	sygnatury, analiza zagrożeń	Może rozpoznać i zatrzymać podejrzany plik lub zachowanie.

Model: właściciel → grupa → pozostali

TRZY ZESTAWY UPRAWNIEŃ

Linux przy sprawdzaniu standardowych bitów dostępu używa odpowiedniej kategorii: **u** = właściciel (user), **g** = grupa (group), **o** = pozostali (others). Nie dodaje uprawnień z trzech kategorii do siebie. Może też istnieć dodatkowa lista ACL lub polityka MAC.

```
$ whoami
uczen
$ ls -l plan.txt
-rw-r----- 1 uczen pracownia 19 lis 2  plan.txt
```

Przeczytaj wiersz od lewej: **-** oznacza zwykły plik; **rw-** – prawa właściciela; **r--** – prawa grupy; **---** – brak praw dla pozostałych. „uczen” to właściciel, a „pracownia” to grupa.

Pomyśl: jeżeli plik ma prawa **600**, czy nieszkodliwy skrypt uruchomiony na koncie właściciela może go odczytać? Dlaczego?

TEORIA | LICZBY, KTÓRE POJAWIAJĄ SIĘ W CHMOD

02 Dlaczego 4, 2 i 1?

Każdy zestaw praw jest zbudowany z **trzech bitów**. W notacji ósemkowej wartość uprawnienia powstaje przez **dodanie liczb** odpowiadających włączonym prawom.

LITERA	LICZBA	DLA PLIKU	DLA KATALOGU
r	4	Odczyt zawartości pliku	Listowanie nazw elementów (zależnie też od x)
w	2	Zmiana zawartości pliku	Tworzenie / usuwanie wpisów (zwykle z x)
x	1	Uruchamianie programu / skryptu	Przechodzenie przez katalog, dostęp do znanej nazwy

Wszystkie kombinacje – od 0 do 7

CYFRA	OBLICZENIE	LITERY	ZNACZENIE
0	0	---	Brak praw
1	1	--x	Tylko wykonanie / przejście
2	2	-w-	Tylko zapis
3	2 + 1	-wx	Zapis i wykonanie
4	4	r--	Tylko odczyt
5	4 + 1	r-x	Odczyt i wykonanie
6	4 + 2	rw-	Odczyt i zapis
7	4 + 2 + 1	rwx	Odczyt, zapis, wykonanie

> **ZAPAMIĘTAJ**
>
> **7 = 4 + 2 + 1 = rwx**. **6 = 4 + 2 = rw-**. **5 = 4 + 1 = r-x**. **4 = r--**. Każda z trzech cyfr trybu chmod jest obliczana osobno dla właściciela, grupy i pozostałych.

Przykład: **chmod 640 raport.txt** oznacza 6 (rw-) dla właściciela, 4 (r--) dla grupy i 0 (---) dla pozostałych.

TEORIA | JAK ROZSZYFROWAĆ TRZY CYFRY

03 Czytamy zapis 755, 644 i 700

Schemat jest zawsze ten sam: **pierwsza cyfra – właściciel, druga – grupa, trzecia – pozostali**. W poleceniu **chmod 754 nazwa_pliku** każda cyfra działa niezależnie.

PRZYKŁAD KROK PO KROKU: 754

7 = 4 + 2 + 1 = **rwx** (właściciel). **5** = 4 + 1 = **r-x** (grupa). **4** = tylko odczyt = **r--** (pozostali). Otrzymujemy: **rwxr-xr--**.

TRYB	ZAPIS LITEROWY	PRZYKŁADOWE ZASTOSOWANIE
755	rwxr-xr-x	Katalog / publicznie wykonywany program
750	rwxr-x---	Katalog dostępny właścicielowi i grupie
700	rwx------	Prywatny katalog właściciela
644	rw-r--r--	Typowy plik odczytywany także przez innych
640	rw-r-----	Raport dostępny właścicielowi i grupie
600	rw-------	Prywatny plik, np. dane poufne (testowe)
777	rwxrwxrwx	Każdy może pisać – ryzykowne; nie ustawiaj bez potrzeby

Uważaj: katalog i plik to nie to samo

Katalog 700: właściciel może oglądać nazwy, tworzyć / usuwać elementy i wejść do środka. **Plik 600**: właściciel może czytać i edytować, ale nie uruchamia pliku bez nadania **x**.

Pułapka: nawet gdy plik jest odczytywalny, brak prawa **x** do któregoś katalogu na ścieżce może uniemożliwić dotarcie do niego. Uprawnienia katalogu wpływają także na możliwość usunięcia pliku.

```
chmod 600 sekret.txt       # prywatny plik
chmod 700 moje_dane         # prywatny katalog
chmod 755 publiczny_katalog # wejście dla innych
ls -l sekret.txt            # litery praw
stat -c "%a %n" sekret.txt  # cyfry praw
```

ĆWICZENIE | SAMODZIELNE OBLICZENIA

04 Kalkulator uprawnień bez kalkulatora

Wypełnij tabelę. Najpierw rozbij prawa na trzy grupy po trzy znaki. Następnie policz każdą cyfrę. Nie korzystaj z gotowych odpowiedzi z internetu.

DANE	PRAWA UŻYTKOWNIKA	GRUPY	INNYCH	WYNIK
rw-r-----	6 = 4 + 2	4 = 4	0	**640**
rwxr-x---	.....	.....	.....	.....
rw-------	.....	.....	.....	.....
rwxrwxr-x	.....	.....	.....	.....
r--r--r--	.....	.....	.....	.....
rwx------	.....	.....	.....	.....

Przelicz w drugą stronę

ZAPIS CYFROWY	DLA WŁAŚCICIELA	DLA GRUPY	DLA INNYCH	ZAPIS LITEROWY
754	rwx	r-x	r--	rwxr-xr--
751	.....	.....	.....	.....
604	.....	.....	.....	.....
711	.....	.....	.....	.....
664	.....	.....	.....	.....

Sytuacja 1: plik z fikcyjnym hasłem ma być widoczny tylko dla właściciela i nie może być wykonywany. Jakie prawa ustawisz?

Sytuacja 2: katalog z materiałami ma być dostępny do odczytu i wejścia dla wszystkich, ale edytować go może jedynie właściciel. Jaki tryb jest najczęściej odpowiedni?

Sytuacja 3: dlaczego polecenie **chmod 777 sekret.txt** jest złym pomysłem?

LEKCJA 01 | ZADANIA PRAKTYCZNE

05 Poznaj swój system i utwórz pliki

Zadanie 1. Zidentyfikuj konto i system

Uruchom terminal Ubuntu. Przepisz polecenia i wyjaśnij, co oznaczają UID, GID oraz wersja jądra. Nie zmieniaj uprawnień systemowych.

```
whoami
id
groups
uname -r
id -u
```

Wpisz nazwę konta, jego UID i grupy. Czy pracujesz jako root (UID 0)?

Zadanie 2. Przygotuj laboratoryjne dane

```
mkdir -p "$HOME/TC_LAB/jawne" "$HOME/TC_LAB/prywatne"
printf 'Plan lekcji - dane testowe\n' > \
  "$HOME/TC_LAB/jawne/plan.txt"
printf 'HASLO_FIKCYJNE_123\n' > \
  "$HOME/TC_LAB/prywatne/sekret.txt"
ls -ld "$HOME/TC_LAB" "$HOME/TC_LAB/prywatne"
ls -l "$HOME/TC_LAB/jawne/plan.txt"
```

Katalog **~/TC_LAB** jest Twoją przestrzenią laboratoryjną. Znak **~** to skrót do katalogu domowego, **$HOME** – jego ścieżka, a **-p** pozwala tworzyć brakujące katalogi.

Zadanie 3. Ustaw prawa i zobacz ich wynik

```
chmod 600 "$HOME/TC_LAB/prywatne/sekret.txt"
chmod 644 "$HOME/TC_LAB/jawne/plan.txt"
chmod 700 "$HOME/TC_LAB/prywatne"
stat -c '%a %n' "$HOME/TC_LAB/prywatne/sekret.txt" \
 "$HOME/TC_LAB/jawne/plan.txt" "$HOME/TC_LAB/prywatne"
```

Odpowiedz: który obiekt ma prawa 600, który 644, a który 700? Po co taki podział?

PUNKT KONTROLNY

Pokaż nauczycielowi wynik **stat** oraz samodzielnie rozszyfruj wybrany tryb na litery. Nie wystarczy samo przepisanie komend.

LEKCJA 02 | ZADANIA PRAKTYCZNE

06 Co blokują uprawnienia?

Zadanie 4. Skrypt, którego nie można uruchomić

Wykonaj polecenia po kolei. Pierwsza próba uruchomienia skryptu powinna zakończyć się odmową dostępu. To poprawny wynik testu.

```
mkdir -p "$HOME/TC_LAB/skrypty"
printf '#!/bin/bash\necho Skrypt uruchomiony!\n' > \
 "$HOME/TC_LAB/skrypty/test.sh"
chmod 644 "$HOME/TC_LAB/skrypty/test.sh"
"$HOME/TC_LAB/skrypty/test.sh"
chmod u+x "$HOME/TC_LAB/skrypty/test.sh"
"$HOME/TC_LAB/skrypty/test.sh"
ls -l "$HOME/TC_LAB/skrypty/test.sh"
```

Co robi chmod u+x? u = właściciel, **+** = dodaj, **x** = wykonanie. To zapis symboliczny. W odróżnieniu od **chmod 755** nie zastępuje od razu wszystkich trzech zestawów praw.

Dlaczego pierwsza próba nie zadziałała? Jaki dokładnie znak pojawił się po wykonaniu chmod u+x?

```
chmod 644 "$HOME/TC_LAB/skrypty/test.sh"
bash "$HOME/TC_LAB/skrypty/test.sh"
```

Czy bash potrafi odczytać skrypt bez bitu x? Co to pokazuje o roli interpretera?

Zadanie 5. Inny użytkownik nie widzi mojego pliku

To ćwiczenie wykonaj, jeśli prowadzący pozwala na użycie **sudo**. Konto **nobody** reprezentuje tu innego użytkownika, a wszystkie pliki leżą w tymczasowym katalogu.

```
KATALOG=$(mktemp -d /tmp/izolacja.XXXXXX)
chmod 755 "$KATALOG"
printf 'PUBLICZNE\n' > "$KATALOG/dane.txt"
chmod 644 "$KATALOG/dane.txt"
sudo -u nobody cat "$KATALOG/dane.txt"
chmod 600 "$KATALOG/dane.txt"
sudo -u nobody cat "$KATALOG/dane.txt"
```

Przy których prawach odczyt zadziałał? Jaką odpowiedź zobaczyłeś przy odmowie dostępu?

LEKCJA 02 | ĆWICZENIA UZUPEŁNIAJĄCE

07 Katalog, maska umask i zasada minimum

Zadanie 6. Co oznacza x w katalogu?

Najpierw stwórz katalog i testowy plik. Następnie odbierz katalogowi prawo przejścia, a po sprawdzeniu przywróć je. Nie zmieniaj katalogów systemowych.

```
mkdir -p "$HOME/TC_LAB/przejscie"
printf 'OK\n' > "$HOME/TC_LAB/przejscie/test.txt"
chmod 600 "$HOME/TC_LAB/przejscie"
cat "$HOME/TC_LAB/przejscie/test.txt"
chmod 700 "$HOME/TC_LAB/przejscie"
cat "$HOME/TC_LAB/przejscie/test.txt"
```

Dlaczego prawo r do katalogu nie wystarczyło, aby otworzyć znany plik wewnątrz?

Zadanie 7. Skąd biorą się prawa nowych plików?

umask jest maską bitów, które mają zostać odebrane podczas tworzenia obiektu. Typowe punkty wyjścia to **666 dla zwykłego pliku** i **777 dla katalogu**. **umask 077** wyłącza prawa grupy i pozostałych; nowy plik otrzymuje zwykle 600, nowy katalog 700.

```
umask
( umask 077
  touch "$HOME/TC_LAB/prywatne/nowy.txt"
  mkdir -p "$HOME/TC_LAB/prywatne/nowy_katalog"
)
stat -c '%a %n' "$HOME/TC_LAB/prywatne/nowy.txt" \
 "$HOME/TC_LAB/prywatne/nowy_katalog"
```

WAŻNY SZCZEGÓŁ

Nie licz umask jak zwykłego odejmowania dziesiętnego. Maska wyłącza **bity**: dla pliku **666 & ~077 = 600**, dla katalogu **777 & ~077 = 700**. Nawiasy w komendzie tworzą podpowłokę, więc zmiana umask nie zostaje na stałe w Twojej powłoce.

Dlaczego plik nie uzyskał prawa wykonania? Podaj wynik komendy stat dla obu obiektów.

Porównanie poleceń

POLECENIE	DZIAŁANIE
chmod 640 raport.txt	Ustaw wszystkie prawa: właściciel rw-, grupa r--, inni ---.
chmod g+r raport.txt	Dodaj grupie prawo odczytu.
chmod o-r raport.txt	Usuń pozostałym prawo odczytu.
chmod u+x test.sh	Dodaj właścicielowi bit wykonania.

LEKCJA 03 | ARCHITEKTURA I ZAGROŻENIA

08 Złośliwy proces a uprawnienia właściciela

System może poprawnie blokować dostęp obcym kontom, a jednak program uruchomiony przez samego właściciela będzie miał prawo czytać jego pliki. To jeden z powodów, dla których potrzebujemy wielu zabezpieczeń naraz.

Zadanie 8. Nieszkodliwy audyt uprawnień

Poniższy skrypt **tylko sprawdza**, czy bieżący użytkownik może odczytać pliki. Nie wysyła danych, niczego nie zmienia ani nie próbuje podnosić uprawnień.

```
cat > "$HOME/TC_LAB/skrypty/audyt.sh" <<'KONIEC'
#!/bin/bash
echo "Konto procesu: $(id -un)"
for p in "$HOME/TC_LAB/jawne/plan.txt" \
         "$HOME/TC_LAB/prywatne/sekret.txt"; do
    if [ -r "$p" ]; then
        echo "ODCZYT DOZWOLONY: $p"
    else
        echo "BRAK DOSTEPU: $p"
    fi
done
KONIEC
bash "$HOME/TC_LAB/skrypty/audyt.sh"
```

Czy skrypt na Twoim koncie zobaczy plik 600? Dlaczego oznacza to ograniczenie samego chmod?

Zadanie 9. AppArmor: dodatkowa warstwa MAC

```
cat /sys/module/apparmor/parameters/enabled
sudo aa-status
```

Jeżeli polecenie **aa-status** nie jest dostępne lub wymaga uprawnień, zapisz ten fakt. **DAC** to klasyczne prawa właściciel/grupa/pozostali. **MAC** to dodatkowa, narzucona polityka ograniczająca działanie programu.

Podaj różnicę między odmową wynikającą z chmod a ograniczeniem aplikacji przez profil AppArmor.

WNIOSEK

Różne zagrożenia wymagają różnych zabezpieczeń: zasady najmniejszych uprawnień, izolacji procesów, aktualizacji, ochrony antywirusowej, nadzoru i poprawnych kopii zapasowych.

LEKCJA 03 | WSPÓLNY POKAZ WINDOWS VM

09 Zobacz prawdziwą reakcję Defendera

Podczas tej części nauczyciel pokazuje Windows uruchomiony w oddzielnej maszynie wirtualnej. Nie będziemy uruchamiać prawdziwego złośliwego oprogramowania. Użyjemy **oficjalnej, nieszkodliwej próbki EICAR** i scenariuszy testowych Microsoft Defender, które potrafią wywołać autentyczne reakcje ochrony.

> **PRZYGOTOWANIE VM**
>
> Przed pokazem: migawka bazowa, wyłączone foldery współdzielone, wyłączony schowek dwukierunkowy i przeciąganie plików. W systemie pozostaw włączone zabezpieczenia. Nie korzystaj ze szkolnych poświadczeń. O przywróceniu migawki decyduje nauczyciel.

Pokaz A. EICAR – test antywirusa, a nie infekcja

1. Nauczyciel otwiera w Windows **Zabezpieczenia Windows → Ochrona przed wirusami i zagrożeniami** i sprawdza stan ochrony.

2. Otwiera oficjalną stronę https://www.eicar.org/download-anti-malware-testfile/ i wybiera plik testowy, np. **eicar.com.txt**.

3. Uczniowie obserwują reakcję. Blokada może nastąpić już w przeglądarce, przy zapisie pliku albo po wykryciu przez Defendera.

4. Wspólnie otwierają **Historia ochrony** i odczytują nazwę wykrycia oraz działanie systemu, jeśli wpis jest dostępny.

Co wykryto? Co zrobił system? Dlaczego alert EICAR nie oznacza, że komputer został zainfekowany?

Pokaz B. Oficjalny test ochrony przed ransomware

Microsoft publikuje **nieszkodliwe scenariusze** dla funkcji Controlled Folder Access (Kontrolowany dostęp do folderów). Prowadzący może uruchomić odpowiednią demonstrację ze strony:

learn.microsoft.com/en-us/defender-endpoint/defender-endpoint-demonstrations

Scenariusz może zależeć od konfiguracji systemu i dostępnych funkcji Defender. Jeżeli dana funkcja nie jest dostępna, analizujemy oficjalny opis i ekran zabezpieczeń, zamiast zmieniać ustawienia ochrony.

Czym różni się wykrycie sygnatury pliku od zablokowania nieautoryzowanej zmiany chronionego folderu?

LEKCJA 03 | DEMONSTRACJA ZACHOWANIA BEZ PRAWDZIWEGO MALWARE

10 „Watykańczyk EDU” – symulacja incydentu

Złośliwe programy mogą wyświetlać niepokojące komunikaty, zmieniać pliki, próbować uzyskać większe uprawnienia lub uruchamiać się ponownie. Nasza **symulacja** pokazuje tylko pierwsze dwa efekty – na sztucznych danych. Nie ma w niej szyfrowania, autostartu ani komunikacji sieciowej.

Pokaz nauczyciela w PowerShell (Windows VM)

Wprowadź polecenia pojedynczo do PowerShell na **specjalnie przygotowanej VM**. Skrypt działa tylko w nowo utworzonym katalogu **INCYDENT_DEMO** na pulpicie. Nie wymaga zmiany polityki wykonywania skryptów ani wyłączania antywirusa.

```
$Lab = Join-Path $env:USERPROFILE 'Desktop\INCYDENT_DEMO'
New-Item -Path $Lab -ItemType Directory -Force | Out-Null
'Projekt szkolny - DEMO' | Set-Content "$Lab\projekt.txt"
'Wyniki testowe - DEMO' | Set-Content "$Lab\wyniki.txt"
Copy-Item -Path $Lab -Destination "${Lab}_KOPIA" -Recurse
Get-ChildItem -Path $Lab -Filter '*.txt' -File |
  Rename-Item -NewName { $_.Name + '.DEMO' }
'To jest TYLKO symulacja szkolna.' |
  Set-Content "$Lab\CZYTAJ_DEMO.txt"
Get-ChildItem -Path $Lab
```

CO WŁAŚNIE ZROBILIŚMY?

Program stworzył dwa pliki fikcyjne, wykonał ich kopię, następnie zmienił im rozszerzenia i pozostawił komunikat. **Żadna treść nie została zaszyfrowana**. Samo nadanie rozszerzenia „.DEMO” nie jest ransomware.

Odwracamy efekt bez „płacenia okupu”

```
Get-ChildItem -Path $Lab -Filter '*.DEMO' -File |
  Rename-Item -NewName { $_.Name -replace '\.DEMO$', '' }
Get-Content "$Lab\projekt.txt"
Get-ChildItem -Path "${Lab}_KOPIA"
```

Co zauważyłeś? Dlaczego w prawdziwym incydencie sama zmiana rozszerzeń nie wystarczyłaby do odzyskania zaszyfrowanych danych?

CZEGO NIE ROBIMY

Nie pobieramy nieznanych próbek, nie wyłączamy antywirusa, nie próbujemy omijać SmartScreen ani zasad PowerShell i nie testujemy na komputerze gospodarza. Migawka VM nie jest gwarancją izolacji ani bezpieczną kopią na wypadek utraty całego dysku.

LEKCJA 04 | KOPIA BEZPIECZEŃSTWA

11 Zabezpiecz dane PRZED incydentem

Zanim pojawi się jakikolwiek problem, musisz mieć **sprawdzoną kopię**. W laboratorium użyjemy archiwum **.tar.gz** i sumy kontrolnej SHA-256. Pracujemy wyłącznie z trzema sztucznymi plikami.

Zadanie 10. Utwórz dokumenty testowe i archiwum

```
mkdir -p "$HOME/TC_LAB/incydent" "$HOME/TC_BACKUP"
printf 'Projekt ucznia - A\n' > \
 "$HOME/TC_LAB/incydent/projekt.txt"
printf 'Baza testowa - 3 rekordy\n' > \
 "$HOME/TC_LAB/incydent/baza.txt"
printf 'Wyniki - fikcyjne\n' > \
 "$HOME/TC_LAB/incydent/wyniki.txt"
tar -czf "$HOME/TC_BACKUP/dane-czyste.tar.gz" \
 -C "$HOME/TC_LAB" incydent
tar -tzf "$HOME/TC_BACKUP/dane-czyste.tar.gz"
sha256sum "$HOME/TC_BACKUP/dane-czyste.tar.gz"
```

Co robi każda część polecenia?

ELEMENT	WYJAŚNIENIE
tar	Narzędzie do tworzenia i odczytywania archiwów.
-c / -x / -t	Utwórz / wypakuj / wyświetl zawartość archiwum.
-z / -f	Kompresja gzip / wskazanie nazwy pliku archiwum.
-C katalog	Przejdź do wskazanego katalogu przed archiwizacją / wypakowaniem.
sha256sum	Oblicza skrót; porównanie sum pozwala wykryć zmianę zawartości archiwum.

Jakie trzy pliki widzisz po tar -tzf? Zapisz początek sumy SHA-256.

Zadanie 11. Utwórz migawkę VM

Po zgodzie nauczyciela w VirtualBox przejdź do **Migawki / Snapshots → Utwórz** i zapisz stan pod nazwą **PRZED_INCYDENTEM**. Opcje interfejsu mogą różnić się wersją VirtualBox. Nie przywracaj migawki jeszcze.

DLACZEGO TO DWA RÓŻNE MECHANIZMY?

Archiwum przywraca wybrane pliki. **Migawka** cofa stan całej maszyny wirtualnej do wcześniejszego momentu. Archiwum zapisane wewnątrz tej samej VM nadal może zniknąć wraz z jej dyskiem – w praktyce potrzebna jest niezależna kopia.

LEKCJA 04 | REAGOWANIE NA INCYDENT

12 Wykryj zmianę i odzyskaj pliki

Zadanie 12. Symulacja zmiany nazw w Ubuntu

Wszystkie działania dotyczą wyłącznie plików w **~/TC_LAB/incydent**. Poniższa pętla nie szyfruje danych i nie modyfikuje żadnych innych katalogów.

```
for f in "$HOME/TC_LAB/incydent/"*.txt; do
  mv -- "$f" "$f.symulacja"
done
printf 'SZKOLNA SYMULACJA - brak szyfrowania\n' > \
 "$HOME/TC_LAB/incydent/CZYTAJ_TO.txt"
ls -l "$HOME/TC_LAB/incydent"
```

Jak zmieniły się nazwy plików? Czy ich treść stała się nieczytelna? Co odróżnia ten test od prawdziwego ransomware?

Zadanie 13. Przywróć dane do osobnego katalogu

```
mkdir -p "$HOME/ODZYSK"
tar -xzf "$HOME/TC_BACKUP/dane-czyste.tar.gz" \
 -C "$HOME/ODZYSK"
ls -l "$HOME/ODZYSK/incydent"
cat "$HOME/ODZYSK/incydent/projekt.txt"
cat "$HOME/ODZYSK/incydent/baza.txt"
```

Odnowione pliki znajdują się w **~/ODZYSK/incydent**. Stary katalog **~/TC_LAB/incydent** zachowujemy do analizy – nie nadpisujemy go bez potrzeby.

Podaj liczbę i nazwy odzyskanych plików. W jaki sposób sprawdziłeś ich zawartość?

Zadanie 14. Przywrócenie migawki – wyłącznie z nauczycielem

Zapisz swoje odpowiedzi i dowody poza odtwarzaną VM w miejscu wyznaczonym przez prowadzącego. Nauczyciel zatrzyma maszynę, przywróci migawkę **PRZED_INCYDENTEM**, a potem porównacie stan systemu. Zmiany wykonane po migawce mogą zniknąć.

WNIOSKI | BEZPIECZEŃSTWO I ODZYSKIWANIE

13 Dlaczego jeden mechanizm nie wystarcza?

MECHANIZM	CHRONI PRZED / POMAGA	OGRANICZENIE
chmod / DAC	Dostępem innych użytkowników do plików.	Nie blokuje programu z prawami właściciela.
AppArmor / MAC	Wykraczaniem programu poza profil.	Skuteczność zależy od reguł i tego, co jest objęte profilem.
Microsoft Defender	Niektórymi złośliwymi plikami i działaniami.	Nie jest gwarancją wykrycia każdego zagrożenia.
Migawka VM	Szybkim cofnięciem stanu maszyny.	Nie zastępuje niezależnego backupu.
Backup	Odtworzeniem danych po awarii lub incydencie.	Musi być dostępny, aktualny i przetestowany.

Odpowiedz własnymi słowami

1. Dlaczego 600 nie oznacza, że plik jest odporny na każde złośliwe oprogramowanie?

2. Jak sprawdzisz, czy odmowa dostępu wynika z uprawnień pliku czy z katalogu?

3. Czym różni się reakcja programu antywirusowego na EICAR od odmowy dostępu „Permission denied”?

4. Co trzeba sprawdzić po odtworzeniu kopii zapasowej?

Do pokazania nauczycielowi: wynik stat (600 / 644 / 700), co najmniej jedna odmowa dostępu, opis testu Defender/EICAR, dowód odtworzenia trzech plików.

DLA UCZNIÓW, KTÓRZY KOŃCZĄ WCZEŚNIEJ

14 Poziom EXPERT

Wyzwanie A. Zaprojektuj prawa od zera

W nowym katalogu **~/TC_LAB/ekspert** utwórz trzy pliki: **regulamin.txt**, **raport.txt** i **haslo_testowe.txt**. Przygotuj prawa dla kolejnych sytuacji, uzasadnij wybór i udowodnij go komendą **stat**.

OBIEKT	WYMAGANIE	TWÓJ CHMOD
regulamin.txt	Właściciel pisze; inni mogą tylko czytać.	.....
raport.txt	Właściciel pisze, grupa czyta; reszta bez dostępu.	.....
haslo_testowe.txt	Tylko właściciel czyta i zapisuje.	.....
katalog ekspert	Tylko właściciel ma pełen dostęp.	.....

Wyzwanie B. Czy najważniejsze jest prawo do pliku?

Zaprojektuj test pokazujący, że prawa **w + x** do katalogu mogą umożliwiać usunięcie wpisu, mimo że sam plik nie przyznaje prawa zapisu jego zawartości. Używaj jedynie własnych plików w **~/TC_LAB/ekspert**. Przedstaw przewidywanie nauczycielowi przed wykonaniem.

Jak różnią się „modyfikacja zawartości pliku” i „usunięcie nazwy z katalogu”?

Wyzwanie C. Model ochrony szkolnego serwera

Zaproponuj cztery warstwy ochrony: np. poprawne prawa, AppArmor, aktualizacje, monitoring logów, kontrola aplikacji i niezależne kopie zapasowe. Dla każdej podaj przykład ataku, którego skutki mogą zostać ograniczone.

Wyzwanie D. Kopie zapasowe 3–2–1

Narysuj schemat: **3 kopie danych, 2 typy nośników lub środowisk, 1 kopia poza lokalizacją**. Oznacz dodatkowo wariant offline / niezmienialny i sposób regularnego testowania odtwarzania.

POMOC NA PRZYSZŁOŚĆ

15 Ściąga poleceń i źródła

POLECENIE	ZASTOSOWANIE
whoami / id / groups	Kim jestem, UID/GID i przynależność do grup.
ls -l / ls -ld	Prawa pliku / samego katalogu.
stat -c "%a %n" PLIK	Wyświetlenie praw liczbowych.
chmod 600 PLIK	Ustaw rw-------.
chmod 755 KATALOG	Ustaw rwxr-xr-x.
chmod u+x PLIK	Dodaj prawo wykonania właścicielowi.
chmod g-r PLIK	Usuń grupie prawo odczytu.
umask	Sprawdź maskę uprawnień nowych plików i katalogów.
sudo -u nobody cat PLIK	Test odczytu jako inny użytkownik – z pozwoleniem.
cat /sys/module/apparmor/parameters/enabled	Sprawdź, czy moduł AppArmor jest aktywny.
tar -czf / tar -xzf	Utwórz / wypakuj archiwum .tar.gz.
sha256sum PLIK	Wylicz skrót SHA-256.

Źródła i oficjalne demonstracje

chmod – dokumentacja systemowa: https://man7.org/linux/man-pages/man1/chmod.1.html
EICAR – oficjalny plik testowy: https://www.eicar.org/download-anti-malware-testfile/
Microsoft Defender – demonstracje: https://learn.microsoft.com/en-us/defender-endpoint/defender-endpoint-demonstrations
VirtualBox – dokumentacja Guest Additions: https://docs.oracle.com/en/virtualization/virtualbox/7.2/user/guestadditions.html

CO POWINIENEŚ UMIEĆ PO TYCH LEKCJACH

1. Obliczyć prawa 0–7 i rozszyfrować chmod 755 / 644 / 600 / 700. 2. Odróżnić prawa pliku od praw katalogu. 3. Wyjaśnić DAC, MAC i rolę ochrony antywirusowej. 4. Rozpoznać rzeczywistą blokadę i nieszkodliwy test EICAR. 5. Samodzielnie przygotować archiwum, sprawdzić zawartość i odzyskać dane.