Zabezpieczenie systemów i usług sieciowych
Laboratorium 2

Uwagi ogólne:
1. Wszystkie zadania wykonywane są na systemie Ubuntu Server 20.04.1 64bit zainstalowany w środowisku emulatora VirtualBox
2. Zaimportowane na pierwszych zajęciach serwery będą potrzebne do wykonania kolejnych ćwiczeń.

Zadania:
1. Import systemu (*)

- ❏ plik ovf do pobrania z adresu: https://zsius-pliki.justdoit.tech/lab_zsius_2020_3.ova
- ❏ w programie Virtualbox wybieramy “Plik” -> “Importuj urządzenie wirtualne”
- ❏ na następnym ekranie wskazujemy plik ovf i postępujemy zgodnie z instrukcjami na ekranie
- ❏ domyślne konto: student, hasło: student
- ❏ w ustawieniach sieciowych maszyny wirtualnej wybieramy zaawansowane -> przekierowywanie portów i sprawdzamy czy jest dodana reguła “host port 2222 -> guest port 22”

2. Logowanie do zdalnego serwera
    - ❏ pobrać program Putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
    - ❏ w pole ‘Host Name (or IP address) wpisać adres ip 127.0.0.1
    - ❏ jako ‘Connection type’ wybrać SSH
    - ❏ jako port wpisujemy 2222
    - ❏ w sekcji ‘Translation’ wybrać charset UTF-8
    - ❏ w sekcji ‘Session’ wpisać dowolną nazwę w pole ‘Saved Sessions’ i kliknąć ‘Save’
    - ❏ wybrać ‘Open’ i jako nazwę użytkownika podać nazwę wybraną podczas instalacji podać hasło wybrane podczas instalacji
    - ❏ **aby uzyskać pełne uprawnienia przed poleceniem dodajemy sudo**

3. Poruszanie się po systemie plików i edycja plików (*)
    - ❏ przejść do katalogu głównego systemu (/)
      komenda cd <katalog>
    - ❏ wyświetlić zawartość katalogu
      komenda ls [przełączniki] [katalog]
        - -l long listing format
        - -a all
        - -h sizes in human readable format
    - ❏ przejść do katalogu /etc i wyświetlić zawartość
    - ❏ otworzyć i wyedytować plik /etc/legal w edytorze vim (sudo vim /etc/legal) lub innym wg preferencji
      komenda vim [plik]
        - przejście do trybu edycji
            - i insert
            - a append
            - Esc wraca do trybu komend
        - najbardziej przydatne komendy edytora

:w zapisz
:q wyjdź
/<tekst> znajdź

4. Przegląd działających procesów

- ❏ - wyświetlić procesy uruchomione w systemie
  komenda ps [przełączniki]
  -e wszystkie procesy w systemie
  -f rozszerzone informacje o każdym procesie
- ❏ - wyświetlić jedynie procesy użytkownika root za pomocą mechanizmu pipe | i komendy grep [przełączniki] <poszukiwane wyrażenie> [źródło danych]. Bez podania źródła polecenie grep oczekuje danych na <STDIN> (zazwyczaj <STDIN> to nasza klawiatura). Przykład komendy: ps -ef | grep student

5. Zarządzanie oprogramowaniem (*)

W systemie Ubuntu programem do obsługi zainstalowanego oprogramowania jest apt. Posiada on szereg sub komend, najczęściej używane to:

apt-get update - aktualizuje bazę dostępnych pakietów oprogramowania
apt-get upgrade - aktualizuje zainstalowane pakiety oprogramowania
apt-cache search - wyszukiwanie pakietu
apt-get install - instalacja pakietu

Zadanie:

Zaktualizować system oraz wyszukać i upewnić się, że są zainstalowane pakiety:

vim
mc
telnet
wget
tcpdump
ufw

6. Instalacja, podstawowa konfiguracja i uruchomienie przykładowej usługi (http) (*)

- ❏ wyszukać i zainstalować pakiet apache2
- ❏ przejść do katalogu /var/www/html i wyedytować plik index.html umieszczając w nim dowolne treści
- ❏ domyślnie w systemie Ubuntu wszystkie porty zapory sieciowej są otwarte. Na tym etapie nie musimy tego zmieniać.
- ❏ uruchamiamy nasz serwer www za pomocą komendy systemctl <akcja> <usługa>. Podstawowe akcje to: start, stop, restart, enable, disable, status a nasza usługa to apache2. Aby wystartować serwer wywołujemy akcje start (przykład: systemctl start apache2)
- ❏ dodajemy kolejny port w ustawieniach sieciowych maszyny wirtualnej (virtualbox): zaawansowane -> przekierowywanie portów ->dodajemy regułę “host port 2280 -> guest port 80”
- ❏ sprawdzamy czy serwer działa poprawnie otwierając stronę: http://127.0.0.1:2280 w przeglądarce zainstalowanej na naszej stacji roboczej.

7. Lokalizacja i przeglądanie logów systemu i aplikacji

- ❏ przejść do katalogu /var/log/apache2
- ❏ za pomocą komendy tail [przełączniki] wyświetlić końcówkę pliku access.log, przydatne przełączniki:
  - -<n> wyświetla n ostatnich linii pliku
  - -f ciągłe wyświetlanie pliku w miarę jego przyrastania
- ❏ użyć mechanizmu pipe | do ciągłego wyświetlania tylko tych linii pliku logu które zawierają frazę 'GET' (patrz przykład w zadaniu numer 4)

8. Podsłuchanie ruchu usługi

- ❏ uruchomić program tcpdump [przełączniki] [filtr], przydatne przełączniki:
  - -n nie rozwiązuje nazw hostów na podstawie adresów IP
  - -A wyświetla złapane pakiety wraz z zawartością
  - -i <interfejs> wybiera interfejs sieciowy serwera
- ❏ Jako filtrów używamy kolejno wyrażeń 'tcp port 80' dla nie szyfrowanego ruchu http i 'tcp port 22' dla szyfrowanego ssh. (przykład: "tcpdump -n -A -l -i enp0s3 tcp port 80")
- ❏ Przy uruchomionym poleceniu tcpdump ponownie otwieramy naszą stronę w przeglądarce i obserwujemy "złapany" ruch sieciowy. Potem porównujemy z ruchem ssh (port 22)

9. Podstawy automatycznej analizy logów

Celem zadania jest zapoznanie się z metodami automatycznej analizy logów systemu operacyjnego i aplikacji. Podstawową aplikacją do automatycznej analizy jest program logwatch (należy zainstalować. W konfiguracji postfix wybieramy 'Local only'). Raport dotyczący wszystkich logów systemu można uzyskać używając przełącznika **--range All**, aby wysłać wyniki na email używamy przełącznika --mailto (jako adres najlepiej podać root@localhost, te maile można przeczytać uruchamiając program mutt)

Ręczne uruchamiania narzędzia jest jednak kłopotliwe. Dlatego też użyjemy programu cron do automatycznego uruchamiania raportowania o określonej godzinie. Aby dodać nową komendę do harmonogramu cron wydajemy polecenie **crontab -e** co uruchomi edytor tekstu (najlepiej wybrać nr 2 vim.basic). Każda linia pliku to oddzielna pozycja harmonogramu. Składnia pojedynczej linii ma postać:
<minuta> <godzina> <dzień miesiąca> <miesiąc> <dzień tygodnia> <komenda do wykonania>
Nieużyte pola zastępujemy *. Na przykład aby uruchamiać program logwatch codziennie o 1:05 w nocy należy wpisać:
5 1 * * * /usr/sbin/logwatch --range All --mailto root@localhost
Ustawiamy kolejne sprawdzenie na obecną godzinę + 3 minuty zapisujemy zmiany i zamykamy edytor (Esc :wq Enter). Aby sprawdzić obecną godzinę maszyny wirtualnej używamy polecenia date, godzina wewnątrz VM nie musi zgadzać się z rzeczywistą.

10. Poszukiwanie rootkitów

Celem zadania jest zapoznanie studentów z podstawowym narzędziem do poszukiwania złośliwego oprogramowania w systemach z rodziny Unix. Program rkhunter (należy zainstalować) analizuje system poszukując śladów które mogło pozostawić złośliwe oprogramowanie. W większości przypadków nie stwierdza on jednak jednoznacznie czy w systemie działa złośliwe oprogramowanie, jest jednak cennym narzędziem pomagającym wyłapać błędy konfiguracji systemu. W pierwszej kolejności uruchamiamy program rkhunter z przełącznikiem -c, wygeneruje to wstępny raport o stanie systemu. Następnie uruchamiamy rkhunter --propupd co spowoduje zapisanie aktualnego stanu plików jako "normalnego", każda ingerencja w pliki systemowe spowoduje wykrycie zmiany przy następnym uruchomieniu rkhunter.

Dodajemy komendę /usr/bin/rkhunter --report-warnings-only --cronjob do pliku harmonogramu i

ustawiamy ją tak aby uruchomiła się za 3 minuty. Wyniki można będzie znaleźć w skrzynce e-mail użytkownika root (do sprawdzenia za pomocą programu mutt).

11. Podstawowe zabezpieczenie usługi SSH (*)

Celem zadania jest ograniczenie dostępu bezpośredniego do uprzywilejowanego konta użytkownika root. Aby to osiągnąć należy przejść do katalogu /etc/ssh. Następnie wykonać kopię pliku sshd_config na sshd_config.bak (cp sshd_config sshd_config.bak). Otworzyć plik sshd_config do edycji i znaleźć linię zawierającą “PermitRootLogin”, zmienić wartość z “prohibit-password” na “no”. Następnie zapisujemy zmiany i zamykamy edytor. Aby zmiany zadziałały konieczne jest ponowne uruchomienie procesu sshd. W tym celu używamy komendy **systemctl restart sshd**

12. Konfiguracja zapory sieciowej (*)

Celem zadania jest zapoznanie studentów z podstawami obsługi zapory sieciowej w systemie Ubuntu. Komendy należy wykonywać z użyciem mechanizmu sudo. Do konfiguracji zapory zostanie użyty uproszczony interfejs UFW. Pierwszym krokiem jest sprawdzenie aktualnego stanu zapory i połączenie się na stronę www utworzoną w zadaniu nr 6. W tym celu wydajemy komendę: ufw status. Powinna wskazywać status inactive. Następnie zezwalamy na dostęp przez ssh (Putty), aby tego dokonać wydajemy polecenie:

ufw allow 22/tcp. Następnie możemy włączyć zaporę poprzez: ufw enable. Ponownie otwieramy naszą stronę www, tym razem nie powinniśmy jej zobaczyć. Aby zezwolić na dostęp dodajemy do zapory port 80 tcp analogicznie jak port ssh.