1. Wprowadzenie

Szybki rozwój dziedziny nowoczesnych technologii sprawił, że dostęp do globalnych zasobów sieci Internet stał się możliwy praktycznie w każdym zakątku ziemi. Już dziś trudno sobie wyobrazić funkcjonowanie oddziałów nowoczesnych przedsiębiorstw, często umiejscowionych w oddalonych od siebie miastach a nawet i różnych kontynentach bez wzajemnej komunikacji.

W dzisiejszych czasach kwestia cyber-bezpieczeństwa stała się kwestią priorytetową, nie tylko dla wielkich korporacji i banków lecz także - lub może raczej przede wszystkim - dla małych firm i zwykłych użytkowników komputerów. W celu zapewnienia pewnego pewnego stopnia bezpieczeństwa sieci niezbędne jest wcześniejsze zbadanie istniejących luk i błędów w jej aktualnym stanie zabezpieczeń. W tym celu bardzo często wykorzystuje się skanery portów sieciowych lub też bardziej rozbudowane narzędzia, przedstawiające dodatkowe informacje o badanej siecji jak na przykład wersja systemu operacyjnego na komputerach w sieci czy też działające usługi sieciowe. Taki oględny skan jest nie tylko wstępem do ataku na daną sieć, lecz może być także świetnym punktem startowym w przypadku, gdy chcemy stworzyć lub wzmocnić zabezpieczenia już istniejące. Aktualnie na rynku znajduję się sporo narzędzi, które można wykorzystać w tym celu, w tej pracy jednak nacisk położony zostanie na zobrazowanie wydajności poszczególnych metod skanowania sieci oraz na przystępną wizualizaję przeprowadzonej analizy sieci z wykorzystaniem interaktywanych diagramów, wykresów i map sieci.

Konieczne zatem stało się zapewnienie poufności oraz auten- tyczności, jednym słowem bezpieczeństwa przesyłanych tym medium danych. Mam na myśli zapobiegnięcie ich podsłuchaniu, nieautoryzowanemu spreparowaniu, przejęciu se- sji lub też uniemożliwieniu w ogóle komunikacji sieciowej (Atak DoS - Deny Of Service). Skutki takich ataków podczas komunikacji np. z bankiem, czy też wewnątrz firmy mogą być paraliżujące na dużą skalę, w konsekwencji - wyjątkowo kosztowne.

1.1. Cele pracy

Celem poniższej pracy jest zaprojektowanie i stworzenie rozbudowanego narzędzia do prowadzenia testów penetracyjnch na sieciach komputerowych małych i średnich rozmiarów, opartych na skanowaniu obecności usług sieciowych z wykorzystaniem jak największej ilości protokołów sieciowych i technik wyszukiwania. Dodatkowo ńależyśtworzyć podsystem wizualizacji wyników (w postaci diagramów lub map topologii sieci), umożliwiający interaktywne prowadzenie dalszych testów penetracyjnych na wybranych jednostkach. Dodatkowo należy dokonać analizy porównawczej wyników procesów skanowania

prowadzonych z przyjęciem różnych priorytetów i kryteriów (czas, wydajność, ąnonimowośćśkanera, poziom ingerencji).

Wybrane rozwiązania zostaną zaprojektowane, wdrożone i zweryfikowane w przykładowej sieci korporacyjnej firmy mającej 6 oddziałów

1.2. Zawartość pracy

W rodziale **??** przedstawiono podstawowe informacje dotyczące struktury dokumentów w LaTeXu. Alvis [**?**] jest językiem

Praca niniejsza składa się z 11 rozdziałów. W rozdziale pierwszym zaprezentowano genezę sieci komputerowych, opis technologii używanych do budowy sieci rozległych, Internetu i protokołu IP. Rozdziały drugi i trzeci poruszają tematy związane z zagrożeniami przesyłania danych poprzez sieć publiczną oraz opis procesów bezpieczeństwa. W rozdziale czwartym znajdują się szczegółowe informacje o technologiach umożliwiających zabezpieczenie danych podczas tranzytu.

2. Implementacja

2.1. pierwsze podejście do implementacji

2.1.1. Pierwsze próby wytworzenia prostego skanera za pomocą jęzkua python 2.7 i biblioteki pylibcap

Po wnikiliwej analizie istniejących rozwiązań w dziedzinie tworzenia skanerów portów sieciowych, zauważyłem, że większość z nich korzysta z języka pyton [źródło]. Z tego też powodu w początkowej fazie projektu postanowiłem skorzystać z prostej Pthon'owej biblioteki pylibcap i napisać krótki skrypt w języku ptyon w wersji 2.7 w celu sprawdzenia możliwości tej biblioteki. Jednakże już po pierwszych chwilach zorientowałem się, że zaimplementowanie za jej pomocą dużego projektu może okazać się kłopotliwe, ponieważ biblioteka ta jest już dość stara a także niezbyt rozbudowana. Mimo to udało mi się wytworzyć narzędzie mogące przysłużyć się w dalszej części mojego projektu.

Conn scan

Na samym pocątku bardzo szubko udało mi się utworzyć narzędzie, które za pomocą najprostszej metody - sprawdzeniu czy podany host odpowiada na próbę połączenia, tak zwany Conn scan - było w stanie uzyskać informacje na temat dostępnych portów na podanej maszynie.

```
def tcp_connect(ip, ports):
    #create a socket
    try:
        #AF_INET -> Internet Protocol v4 addresses, STREAMing socket
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error,err_msg:
      print 'Cannot create a socket'
      sys.exit()

    #checking ports
    for port in ports:
        try:
            #try to connect, if success port is opened
            result = s.connect_ex((ip,port))
            if result == 0:
```

```
            print 'port ' + str(port) + ' open'
            pass
        #close scoket and prepare new one
        s.close()
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error:
        pass
```