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 przeanalizowanie 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 sieci, jak na przykład wersja systemu operacyjnego 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 oraz na przystępną wizualizaję przeprowadzonej analizy z wykorzystaniem interaktywanych diagramów, wykresów i map sieciowych.

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 o skanowanie obecności usług sieciowych z wykorzystaniem jak największej liczby protokołów sieciowych i technik wyszukiwania. Dodatkowo stworzony zostanie 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).

W celu zaprezentowania wiarygodnych wyników analizy porównawczej przeprowadone zostaną testy na bazie sieci komputerowej niewielkich roziarów liczącej kilka maszyn. Porównianiu poddane zostaną parametry wszystkich omawianych w tej pracy metod i stworzone zostaną dane wizualizujące wyniki testów w celu głębszego zrozumienia różnic między algorytmami.

Źródłem omawianego problemu badawczego jest chęć porównania metod skanowania usług sieciowych pod różnym kątem, w sposób przystępny dla mniej zaawansowanych osób zajmujących się dziedziną bezpieczeństwa sieci komputerowych. Obecnie istnieje spora liczba oprogramowania zajbującego się podobną problematyką, lecz większości z istniejących rozwiązań brakuje klarownego rozróżnienia pomiędzy poszczególnymi metodami skanowania.

W zamierzeniu praca ta ma pomóc zrozumieć różnicę między omawianymi w tej prazcy metodami pod względem różnych parametrów, takich jak inwazyjność w badaną sieć czy szybkość działania metody. Na podstawie uzyskanych wyników autor postara się wyłonić metody wyróżniające się od innych pod danym względem.

1.2. Zawartość pracy

Praca niniejsza składa się z rozdziałów. W drugim pierwszym przedstawiono problematykę związaną ze zwrostem zagrożeń w globalnej sieci internet oraz co ze wzrostem tym się wiąże. Wyjaśniono czym jest i na czym polega prowadzenie testów penetracyjnych. Omówiniono także podstawy działania i budowy protokołów TCP, UDP oraz [datagramów] IP.

2. Podstawy teoretyczne

W rozdziale tym przedstawione zostaną zagadnienia ściśle związane z tematem tej pracy, których znajomość niezbędna jest do zrozumienia kolejnych rozdziałów.

2.1. Wzrost zagrożeń w globalnej sieci Internet

Obecnie sieci IP są w znaczym stopniu zabezpieczone w sposób niewystarczający, co za tym idzie, hakerzy są w stanie wykorzystać istniejące luki w ich infrastrukturze do uzyskania nieautoryzowanego dostępu do wrażliwych danych przedsiębiorstw w celu zakłócenia prawidłowego ich działania

Zgodnie z danymi przedstawionymi w raporcie amerykańskiego Federalnego Biura Śledczego, departament Internet Crime Complaint Center (IC3), [1] z roku na rok wzrasta ilość strat poniesionych w wyniku cyber-przestępstw ustalając się w roku 2015 na poziomie 1,070.71 milionów dolarów amerykańskich co w porównaniu z rokiem 2001 (17.8 milionów USD) wynosi wzrost o ponad 600 procent. W związku z tak gwałtownym wzrostem strat poniesionych przez przedsiębiorstwa narasta potrzeba zabezpieczenia wrażliwych na ataki infrastuktur informatyczych.

Instnieją różne sposoby sprawdzenia poziomu bezpieczeństwa instniejącej sieci. Dla przykładu ńetwork vulnerability assessments"badają każdy komponent sieci próbując "determineśzeroką gamę podatności i luk. Ąutomated vulnerability scanners"mogą być użyte do rutynowej kontroli sieci i ostatecznie testy penetracyjne testują, czy zabezpieczenia danej sieci mogą być złamane w danym przedziale czasowym.

2.2. Czym są testy penetracyjne

Zgodnie z definicją [**?**], testy penetracyjne sieci są sposobem dla przedsiębiorstw i innych organizacji do odnalezienia luk w zabezpieczenia na tyle wcześnie, aby zapobiec wykorzystaniu ich przez hakerów do włamania się i kradzieży istotnych dla firmy danych i informacji. Informacje te mogą przejawiać się w wielu różnych formach, takich jak własność intelektualna, kod źródłowy lub jego dokumentacja czy też nazwy użytkowników i ich hasła, niezależnie jednak od formy, są to dane na których organizacja ta w znaczym stopniu się opiera i powinny być one chronione (?)[].

Ważnym pojęciem w tym temacie są tak zwani ętyczni hakerzy- osoby zatrudniane do włamania się do sieci a następnie przeprowadzenia licznych ataków przy użyciu technik, które mogą wykorzystać przestępcy internetowi (?).

Bazowo rozróżniamy dwa typy testów penetracyjnych [**?**]:

1. testy zapowiedziane (announced testing) - próba przechwycenia wyszczególnionych wcześniej plików będących "flagamiłub też próba skompromitowania systemów w sieci klienta korzystając z pełnej kooperacji z zespołem technicznym, dokumentacji projektowej lub kodu źródłowego,

2. testy niezapowiedziane (unannounced testing) - różni się od testów zapowiedzianych tym, że jedynie "upper levels of management"jest poinformowany o tym fakcie. Tego typu testy sprawdzają procedury bezpieczeństwa, instniejącą infrastrukturę bezpieczeństwa a także żeaktywność"pracowników.

Ponadto możemy rozrózniać [**?**],[**?**]:

1. test penetracyjny z minimalną wiedzą (black box) w którym konsultanci bezpieczeństwa nie mają wcześniejszej wiedzy na temat sieci którą mają penetrować, poza informacją, która sieć jest ich celem. Wymaga to poświęcenia sporej ilości czasu w fazie przygotowawczej, jest to jednak etap wartościowy dla klienta, ponieważ dostarcza informacji na temat danych, które są dostępne na jego temat dla outside world

2. test penetracyjny z pełną wiedzą (crystal box, white box) gdzie zespół posiada kluczowe informacje na temat sieci będącej celem, jej konfiguracji a także dane na temat oprogramowania i sprzętu będącego w użyciu firmy. W cały proces zaangażowany jest także presonel firmy zapewniający szczegółowe informacje na temat systemu i kontolujący jego działanie w czasie trwania testów aby nie sprawiały one problemów dla zwykłych użytkowników.

3. testy penetracyjne z częściową wiedzą (grey box) znajdujący się pomiędzy wcześniejszymi dwoma. W tym typie testu, zespół posidada podstawowe informacje na temat testowanej sieci takie jak adres IP czy nazwa hosta. Dzięki temu testerzy nie muszą poświecać tak wiele czasu na uzyskiwanie podstawowych informacji i mogę dokładniej zbadać inne istotne moduły systemu.

istotne jest aby po skończoncyh testach penetracyjnych dostarczyć do klienta dokładny raport z wynikami, jakie po testach tych udało się osiągnąć

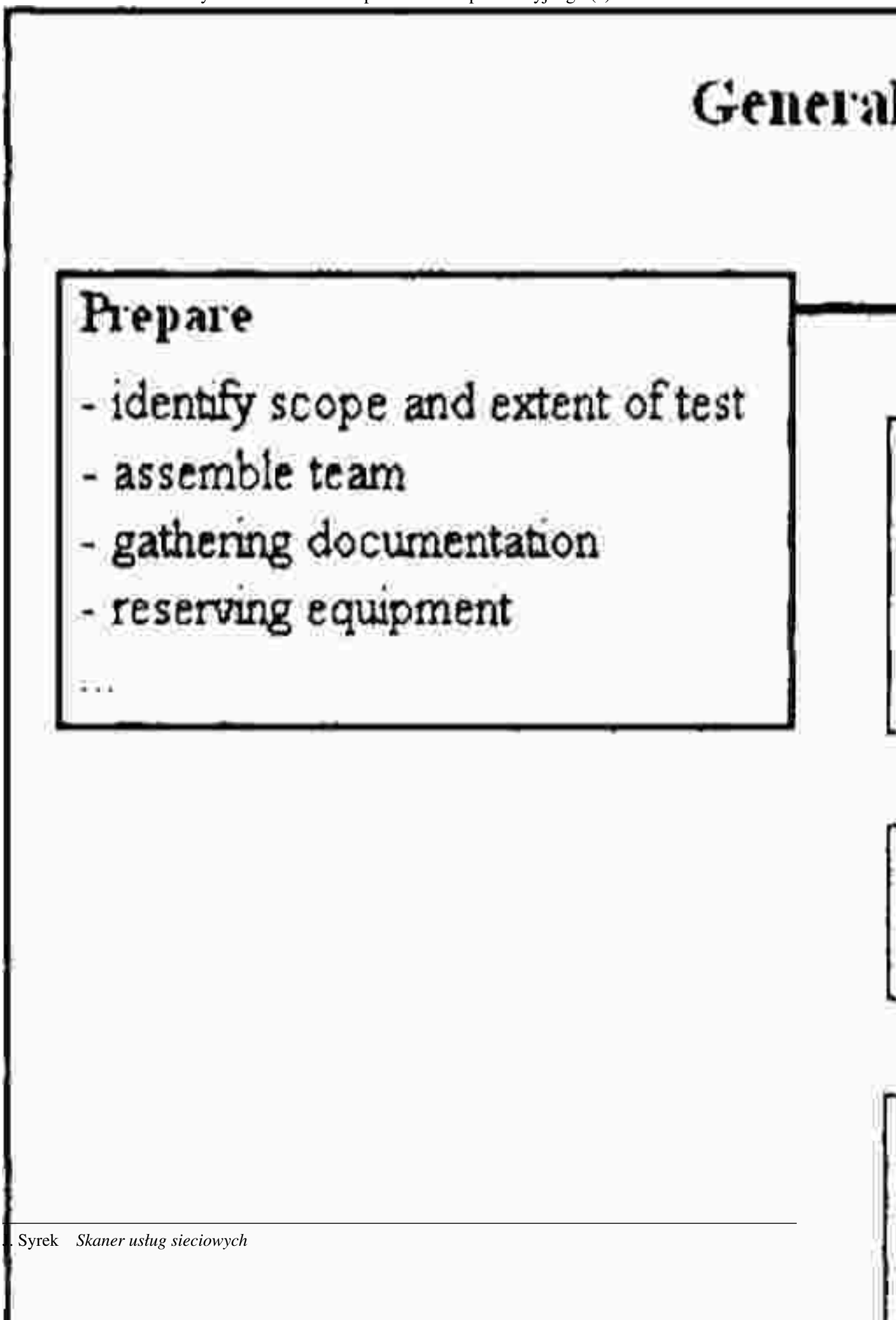
2.2.1. Rozwój testów penetracyjnych

<charakter> testów penetracyjnych w znacznej mierze zależy od ewolucji narzędzi służących do hackingu a używanych przez cyber-przestępców. Bazowa struktura pentestingu jest jednak stała i składa się z czterech faz (Przygotowania, Analiza, Dokumentacja i Cleanup). Jednakże używane techniki oraz nasick <balans rozkład> mogą ulegać zmianie zależnie od pojawiania się nowych wątków w tematyce bezbieczeństwa sieciowego pojawiających się se wględu na rozwój coraz bardziejzłożonych narzędzi służącyhc ddo hackingu, potrzeb redukowania kosztów jak rówież wprowadzania nowych typów sieci

2.3. przykład scenariusza pentestingu

todo

Rysunek 2.1: Schemat procesu testu penetracyjnego (?)



3. Implementacja

3.1. pierwsze podejście do implementacji

3.1.1. 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):
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error,err_msg:
      print 'Cannot create a socket'
      sys.exit()

    for port in ports:
        try:
            result = s.connect_ex((ip,port))
            if result == 0:
                print 'port ' + str(port) + ' open'
                pass
            s.close()
            s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
        except socket.error:
            pass
```

4. Skanowanie sieci

Skan sieci [**?**] jest czynnością podstawową dla uzyskiwania informacji o aktualnym stanie systemu komputerowego lub sieci. Co za tym idziem, jest stosowany jako pierwszy, podstawowy krok potencjalnych napastników względem wybranych przez nich celów. W pewnych okolicznościach skan uważany jest także za atak sam w sobie.

Jednakże może one też służyć słusznym celom, takim jak na przykład sprawdzanie konfiguracji systemów, koregowanie reguł bezpieczeństwa czy monitorowanie wielko-skalowych środowisk sieciowych. W tym sensie/pod tym względem, skanowanie jest efektywną metodą przy sprawdzaniu środków bezpieczeństwa sieci/kontroli

W rozdziale tym przedstawione zostaną podstawowe typy skanerów, ich charakterystyki jak również różnice między nimi

4.1. Na czym polega scanning?

Bez wcześniejszego zeskanowania danej sieci hakerzy nie posiadają wystarczających informacji na ich temat i w większosci wypadków nie są w stanie przejść do kolejnego etapu ich strategi ataku.

Najbardziej skomplikwane scenariusze są praktycznie niemożliwe do wykonania bez wcześnieszego dokładnego zbadania sieci będącej celem ataku. Z innej strony, techniki skanowania sieci zostały zaadaptowane w dziedzinie bezpieczeństwa IT i służą do wykrywania słabości i wad danego systemu lub sieci.

W wyniku ciągłego rozowju branży IT infrastruktury sieciowe stają się coraz bardziej skomplikowane. W tej persepktywie skaning jawi się jako efektywne narzędzie do kontroli i badania tak złożonych systemów. Wraz ze wzostem wymagań ze strony zarówno osób atakujących jak i osób odpowiedzialnych za bezpieczeństwo, w ostatnich latach doszło do szybkiego rozwoju dziedziny metod i technik skanowania.

Liczne metody skanowania odróżniają od siebie ich kompetencje (w funkcjonalności?), a co za tym idzie, skutkują różnymi rodzajami implementacji. Ich wyniki są zazwyczaj odmienne i nie machine-readable. Łącząc kilka typów skanerów można *jednak) uzyskać gruntowne lecz jednocześnie łatwe w interpretacji wyniki.

4.2. attack graph

Attack Graphs have been proposed for years as a formal way to simplify the modeling of complex attacking scenarios [9]. An attack graph describes not only one possible attack, but also many potential ways for an attacker to reach a goal. the workflow of an attack graph construction tool consists of three in- dependent phases: Information Gathering, Attack Graph Construction, as well as Visualization and Analysis.

4.3. Przegląd metod skanowania sieci

Techniki skanowania sieci klasyfikuje się w następujący sposób (kategorie):

1. pasywne - brak wysyłania jakichkolwiek pakietów,
2. wyszukujące (discovery) - pakiety wysyłane są wyłącznie w celu odnalezienia hostów i wykryciu łączności,
3. skan portów - systematyczne sprawdzenie indywidualych portów lub ich przedziałów. Pakiety z odpowiedzią mogą być sprawdzone względem posiadania software banners,
4. probing - wykryte serwisy sieciowe sprawdzane są pod kątem wersji i słabości/podatności(vulnerabilities),
5. exploiting - wykryte luki są aktywnie wykorzystywane lub sprawdzane,
6. inne - aktywne, jednak nie mogą być zakwalifikowane do żadnej z poprzednich kategorii.

Większość z emerged technik skanu są zaimplementowane na podstawie przedstawionej powyżej klasyfikacji. W praktyce obejmuje to:

1. skaner topologii - użyty do sprawdzania struktury sieci składającej się na wykrycie hostów, maski sieciowej i bramek sieciowych (gateways). W przypadku zaawansowanych skanerów, liczne sieci są skanowane aby w rezultacie przedstawić jednorodny, kompletny diagram sieci do sprawdzenia ich wzajemych połączeń.
2. skaner portów sieciowych -
3. skaner ódcisu palca"(Fingerprint Scanner) -
4. skaner podatności (Vulnerability Scanner)
5. skaner przyrostowy (Incremental Scanner) -
6.

4.4. Protokół TCP

W sieci komputerowej każdy host jest niejako zamkniętą przestrzenią. Sposobem na komunikację z innymi maszynami w sieci jest port. W modelu OSI[przypis] port przynależy do warstwy transportowej. Warstwa ta identyfikuje każdy serwis działający na urządzeniu poprzez jeden ustalony numer portu, będący 16-bitowym adresem. Rozróżnia się dwa typy adresów portów [**?**]:

1. numery uniwersalne - będące rozdzielone pomiędzy serwisy sieciowe przez organizację ICAAN[link], posiadajce wartości z zakresu 0 do 1023,
2. numery zwykłe (ordinary port number) - w sposób losowy mogą być przydzielane pomiędzy procesy klienckie (can random distribute the customer process to the requested service)

W warstwie transportowej znajdują się dwa istotne protokoły transmisji danych:

1. transmission control protocol (TCP)
2. user Datagram Protocol (UDP)

Podstawa działania skanera portów jest bardzo prosta. Narzędzie to próbuje nawiązać połączenie TCP lub UDP za pomocą tak zwanego gniazda (socket) będącego parą portu o danym numerze oraz adresu IP. Na podstawie odpowiedzi hosta na przesłany sygnał (lub jej braku) można stwierdzić, czy dany port jest otwarty, a co za tym idzie, czy dany serwis sieciowy do niego "podpięty"działa na danej maszynie.

4.4.1. Budowa TCP/IP

Adres IP hosta wraz z numerem portu tworzą parę będącą gniazdem (socket) dla połącznia z inną maszyną w sieci. Aby uzyskać dostęp do serwisu sieciowego za pomocą protokołu TCP, połaczenie musi być explicitly ustanowione pomiędzy gniazdem na maszynie wysyłajacej oraz maszynie odbierające. Dzięki temu połączenia TCP są identyfikowane bezpośrednio za pomocą ich dwóch punktów końcowych (gniazda na obydwu maszynach). Między punktami końcowymi dochodzi do wymiany danych za pomocą segmentów [**?**].

Segment TCP składa się z obowiązkowego nagłówka oraz opcjonalnych danych. Nagłówek natomiast zawiera w sobie sześć bitów flag z których jeden lub więcej może być zwróconych? w tym samym czasie [1,2]:

– SYN - synchronizuje numery sekwencji aby zainicjować połaczenie,

– FIN - nadawca zakończył przesyłanie danych,

– RST - zresetuj połączenie,

– URG - informuje o istotności pola "priorytet",

– ACK - informuje o istotności pola Ńumer potwierdzenia"

– PSH - odbiorca wymaga natychmiastowego przesłania danych,

– NS – (ang. Nonce Sum) jednobitowa suma wartości flag ECN (ECN Echo, Congestion Window Reduced, Nonce Sum) weryfikująca ich integralność

– CWR – (ang. Congestion Window Reduced) flaga potwierdzająca odebranie powiadomienia przez nadawcę, umożliwia odbiorcy zaprzestanie wysyłania echa.

– ECE – (ang. ECN-Echo) flaga ustawiana przez odbiorcę w momencie otrzymania pakietu z ustawioną flagą CE

4.4.2. Nawiązywanie połączenia TCP

Protoków TCP jest zorientowanym połączeniowo, niezawodnym, (strumieniowym) protokołem komunikacyjnym. Zorientowanie połączeniowo oznacza, że dwie aplikacje użuwające TCP muszą ustanowić wzajemne połączenie TCP przed wymianą danych. Niezawodność zapewniona jest poprzez użycie sum kontrolnych, liczników, sekewncjonowania danych i potwierdzeń. Dzięki przypisywaniu numerów sekwencji do każdego przesyłanego bajtu oraz wymaganiu potwierdzenia po otrzymaniu, TCP gwarantuje niezawodność w dostarczaniu danych.. Numery sekwencji używane są do zapewnienia właściwego porządku (otrzymywanych) danych oraz do wyelimnowania zduplikowanych danych. Zazwyczaj w sesji TCP działają (jednocześnie) strumienie danych (każdy punkt końcowy otrzymuje oraz przesyła dane). W związku z tym inicjacyjne numery sekwencji (?) musza być przydzielone do każdego strumienia kiedy połączenie jest ustanawiane.

<conn establish process»

Szczegóły implementacyjne TCP

Większość implementacji TCP/IP stosuje (się) następujące zasady:

– kiedy odbierany jest segment SYN (lub FIN) dla zakmniętego portu (i.e., for which no TCB exists), TCP odrzuca segment i odpowiada nadawcy segmentem RST,

– kiedy segment RST dostarczany jest do portu nasłuchującego, jest po prostu odrzucany,

– kiedy segment RST dostarczany jest do portu zamkniętego, jest (po prostu) odrzucany,

– kiedy segment zawierający flagę ACK jest dostarczany do portu nasłuchującego, TCP odrzuca segment i odpowiada nadawcy segmentem RST,

– kiedy segment z "wyłączoną"flagą SYN jest dostarczany do portu nasłuchującego, the normal three.way handshake continues by replying SYN/ACK,

– When a FIN segment arrives for a listening port, it is simply dropped. "FIN behavior"(closed port = RST, listening port = dropped) can also be seen with the PSH and URG flags, with a TCP segment with no flags, and with all the combinations of FINIPSHIURG [5].

Bibliografia

[1] ???. *https://pdf.ic3.gov/2015_IC3Report.pdf*. 2015r?.

[2] Klevinsky, T.J., Laliberte S., Gupta A. *Hack I.T.*. Addison - Wesley, December 2004.

[3] Geer, D., Harthorne, J. *Penetration Testing: A Duet*. Proceedings of the 18 th Annual Computer Security Applications Conference (ACSAC'02), pp 185-198.