SKANER BEZPIECZEŃSTWA JAKO NARZĘDZIE DIAGNOSTYCZNE

Dariusz LASKOWSKI, Ireneusz KRYSOWATY, Paweł NIEDZIEJKO

Wojskowa Akademia Techniczna, Wydział Elektroniki, Instytut Telekomunikacji
00-908 Warszawa, ul. Gen. S. Kaliskiego 2, tel.: (+48)(22) 683 73 53, fax: (+48)(22) 683 90 38,
email: dlaskowski / ikrysowaty / pniedziejko @wset.edu.pl

Streszczenie

Identyfikacja i utrzymanie zadanego poziomu stanu bezpieczeństwa systemu teleinformatycznego (ang. *network*) jest jednym z najważniejszych elementem polityki bezpieczeństwa (ang. *security policy*). Wymaga ono od osób odpowiedzialnych za zapewnienie bezpieczeństwa oprócz stosowania narzędzi ochrony (tj. zapory ogniowe, systemy wykrywania włamań i antywirusowe oraz systemy kontroli treści) również nieustannego nadzorowania i analizowania zjawisk zachodzących w systemie. W celu uzyskania wiarygodnych informacji diagnostycznych wykorzystuje się dedykowane oprogramowanie (skanery bezpieczeństwa) czasu rzeczywistego. Ułatwia ono wykrycie potencjalnych luk w zabezpieczeniach systemu oraz zawiera mechanizmy ich naprawy. Uwzględniając ważność i aktualność oraz wieloaspektowość problemu bezpieczeństwa, dokonano przeglądu darmowych narzędzi (ang. *Open Source*) stosowanych do technicznego audytu bezpieczeństwa systemu.

Słowa kluczowe: system teleinformatyczny, bezpieczeństwo, diagnostyka komputerowa, narzędzia diagnostyczne.

SECURITY SCANNERS AS DIAGNOSTIC TOOL

Summary

Identification and maintenance established network security level is one of the most important security policy elements. It is requires from people responsible for security assurance apart using security tools (e.g. firewalls, intrusion detection and antivirus systems and content control systems) also continuous monitoring and analyzing phenomena occurring in the system. Dedicated real-time software (security scanners) is developed to get reliable diagnostic information. That software facilitates potential gaps in the system's protection as well as concerns repair mechanisms. Review of the open source tools was made taking into consideration importance and topicality as well as multiaspect of the security problem. Open source tools are used for technical security audit.

Keywords: network, security, computer diagnostic, diagnostic tool.

WSTĘP

Skanery bezpieczeństwa, nazywane również programami audytu bezpieczeństwa, ze względu na posiadane właściwości wykorzystywane są zarówno przez dostawcę usług sieciowych (oficera bezpieczeństwa lub administratora systemu) jak i przez nieupoważnione osoby[1]. Te dwa podzbiory osób mają na celu pozyskanie informacji, o występujących w systemie teleinformatycznym (*STI*) lukach, dla odmiennych celów.

Problem wykorzystania narzędzi bezpieczeństwa jest złożony, dlatego też obecnie oferowane są różnego rodzaju oprogramowania rozwijane przez destruktorów. Kolejnym etapem w procesie ewolucji skanerów bezpieczeństwa jest ciągłe adoptowanie ich do potrzeb osób odpowiedzialnych za bezpieczeństwo sieciowe. Istnieją również sporadyczne przypadki wystąpienia procesu odwrotnego, czyli skanery napisane pod kątem zapewnienia bezpieczeństwa *STI*, ze względu na swoją funkcjonalność i prostotę są używane przez włamywaczy komputerowych. Praktycznie skanery bezpieczeństwa możemy podzielić z uwzględnieniem wielu kryteriów, jednym z nich mogą być koszty:

– narzędzia komercyjne są drogie i dostępne dla nielicznej grupy użytkowników (banki, instytucje rządowe),
– narzędzia dostępne na zasadach *Open Source* posiadają przyjazny dla użytkownika interfejs graficzny umożliwiający i można uzyskać bezpłatnie.

Wykorzystanie pojedynczych narzędzi audytu bezpieczeństwa w sposób przypadkowy i nieprzemyślany nie daje gwarancji uzyskania wiarygodnej informacji w postaci raportu o lukach w zabezpieczeniach i możliwościach włamania. Skuteczność działań polityki bezpieczeństwa występuje przy zastosowaniu testów penetracyjnych na podstawie, których odwzorowuje się w dość jednoznaczny sposób typowe postępowanie

[1] Rozumiane jako destruktor, włamywacz komputerowy, hacker, cracker.

destruktorów[2]. Najbardziej uogólniony schemat realizacji działania hacker'ów zakłada istnienie następujących etapów:
- przygotowanie ataku: rozpoznanie, identyfikacja podatności, przygotowanie odpowiednich narzędzi,
- atak właściwy: wykorzystanie zidentyfikowanych podatności, uzyskanie dostępu do zasobów, ich kradzież, modyfikacja lub zniszczenie, ewentualne pozostawienie ukrytego wejścia,
- zatarcie śladów nieuprawnionej działalności.

Najistotniejsze znaczenie ma etap przygotowania ataku, czyli rozpoznania badanego (atakowanego) systemu oraz samo przygotowanie (wybór) odpowiednich narzędzi dających duże prawdopodobieństwo powodzenia ataku.

1. SKANERY SIECIOWE I SYSTEMOWE

Dla administratora systemu najgroźniejsze są włamania odnajdywane i rozpoznawane z dużym późnieniem, w których występuje kradzież danych. Dla licznego grona organizacji (ubezpieczeniowych, finansowych, itp.) o wiele bardziej korzystne jest utracenie danych, możliwych w pewnym stopniu do odzyskania z kopii bezpieczeństwa, niż możliwość ich wykorzystania przez konkurencję. Dlatego bezpieczeństwo *STI* od strony sieci jest niezwykle istotnym problemem. Przez punkty dostępu do *STI* (a zwłaszcza lokalną sieć komputerową) można do atakowanego systemu dostać się wykorzystując trzy podstawowe techniki:
- poprzez podsłuch (ang. *sniffer*),
- poprzez użycie skanera (ang. *scanning*),
- przy pomocy podszywania (ang. *spoofing*).

Z obserwacji praktycznych zjawisk sieciowych, pierwszym popularnym i groźnym działaniem jest uruchamianie skanera automatycznie wyszukującego luki w systemie. Programy tego typu dzieli się je na dwie kategorie:
- skanery systemowe badają stację lokalną (ang. *host*), poszukując luk w wynikających z przeoczeń, drobnych błędów w administracji, konfiguracji, przykładem skanera dla Linux'a jest *Computer Oracle and Password System*,
- skanery sieciowe testuje stację lokalną przez łącza sieciowe w zakresie dostępności usług i portów, poszukując potencjalnych luk do przeprowadzenia ataku, przykładem skanera jest *Internet Security Scanner*.

Skaner wysyła zapytania do portów serwera i zapisuje odpowiedź od niego otrzymaną. Gromadzi on cenne informacje na temat wybranego serwera. Wymagania systemowe skanerów to:
- duży rozmiar pamięci operacyjnej RAM,
- posiadanie odpowiednich bibliotek oraz usług przez platformę systemową,
- połączenie z zasobami sieci.

[2] Celem destruktorów jest włamanie się do systemu i jego zdestabilizowanie lub też kradzież albo modyfikacja informacji.

Przed skanerem nie da się ukryć zasobów systemowych w procesie diagnozowania, ponieważ skaner ujawnia większość słabych punktów systemu, wspomaga proces utrzymania i przywracania stanu zdatności przez kontrolę bezpieczeństwa oraz jest ważnym narzędziem służącym zabezpieczeniu sieci.

Do obserwacja funkcjonowania skanerów służą następujące programami:
- *courtney* - to skrypt napisany w Perl'u, który w połączeniu z *tcpdump* wykrywa procesy skanowania realizowane przez *SATAN'a* i *SAINT'a*,
- *IcmpInfo* - wykrywa podejrzane działania (skanowanie, bombardowanie, itp.) z wykorzystaniem ICMP,
- *scan-detector* - wykrywa skanowanie TCP/ UDP,
- *portSentry* - zaawansowane narzędzie rozpoznające atak i próbę jego zablokowania.

Wypracowanie skutecznej metody ochrony przed szkodliwymi skutkami oddziaływania destruktora jest realizowane poprzez dokładną analizę zastosowanego narzędzia. Dlatego też administratorzy *STI* z reguły rozpoczynają badania efektywności postawionych zapór przez stosowanie tych samych narzędzi, które używane są przez hackerów.

Legalność skanerów sieciowych to temat dyskusyjny. Niektórzy uważają, że przeczesywanie systemu to naruszenie prywatności. Inni stoją na stanowisku, że uruchamiając serwer w Internecie wyraża się przynajmniej domniemaną zgodę na skanowanie.
W końcu adres sieciowy można porównać do numeru telefonicznego, a ten każdy ma prawo wybrać. Przepisy prawne nie regulują jeszcze tej kwestii, więc według prawa programy skanujące nie są nielegalne. Jednak dla administratorów sieci każde wykorzystanie takich aplikacji przez osoby z zewnątrz jest działaniem nielegalnym.

Przykładowym narzędziem analizy stopnia zabezpieczenia sieci jest hacker „z referencjami". Znane są organizacje zatrudniające takich ludzi i skłonne ponosić duże nakłady finansowe za obronę sieci przed atakami innych hacker'ów. Posiadający odpowiedni zasób wiadomości hacker wie, co jest aktualnie „na topie", jak to wykorzystać i jak się przed tym bronić. Problem polega jedynie na tym jak odróżnić hackera „z referencjami" od przestępcy.

Chyba najbardziej popularnym i jednocześnie najsławniejszym skanerem jest *SATAN*, uznawany przez znacząca większość organizacji do analizy bezpieczeństwa systemu, na podstawie opracowań *National Computer Security Institute*.

1.1. Skaner *SATAN*

Security Administrator's Tool for Analyzing Networks (*SATAN*) autorstwa Dan Farmer i Weitse Venema po raz pierwszy ukazał się 5 kwietnia 1995 roku i jest dostępny w ogólnoświatowych zasobach sieciowych. *SATAN* to pierwszy skaner portów

TCP/IP umożliwiający złożone testowanie *STI* za pomocą prostych procedur w systemie operacyjnym UNIX i Windows i inne platformy systemowe.

Składa się on z kilku modułów skanujących, wykrywających luki w zabezpieczeniach zdalnych stacji roboczych, poprzez badanie m.in.:

- kontrolę dostępu do serwera,
- dostęp do zdalnych powłok,
- dostęp do usług (np. usługi FTP),
- eksportowane systemy plików NFS,
- hasła NIS,
- luki *sendmaila* i protokołu TFTP.

SATAN zyskał dużą popularność dzięki przejrzystości i ścisłości raportów wyświetlanych z wykorzystaniem możliwości Perla i zwykłej przeglądarki interpretującej język HTML'a. Kategorie danych wyjściowych raportowanych to:

- *vulnerabilities* - zawiera spis wrażliwych punktów sieci lub stacji roboczej i miejsc ich występowania (np. serwer, terminal, itp.),
- *host information* - zawiera informacje o: lokalizacji serwerów w sieci, każdym komputerze opisanym przez program, podsieciach i domenach itp.,
- *trust* - pomaga w poznaniu relacji ufności pomiędzy systemami, bada te relacje za pomocą komputerów obsługujących zdalne logowanie, udostępniających systemy plików itp.

Należy podkreślić, iż skanery to nie tylko wyszukiwacze luk, ale przede wszystkim nieocenione narzędzie programowe przypominające o konieczności ich ciągłego wyszukiwania i usuwania. Przykładami innych skanerów są: *Security Administrator's Integrated Network Tool* (*SAINT*), *Internet Security Scanner* (*ISS*), *Nessus*, *nmap*, *CGI scanner v1.0*, itd.

1.2. Skaner *SAINT*

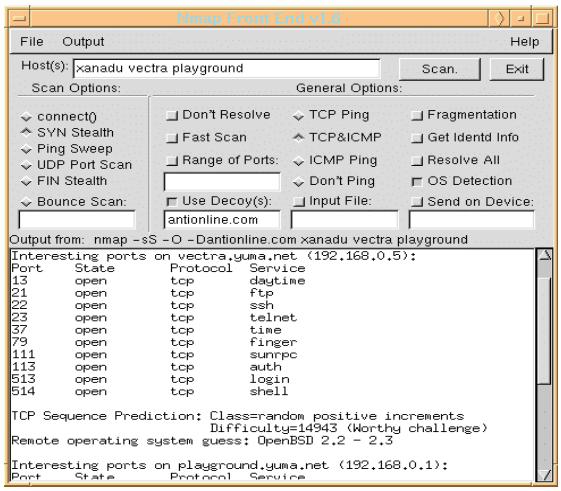
Security Administrator's Integrated Network Tool jest rozszerzeniem i udoskonaleniem programu *SATAN*. Program testuje wszelkie rodzaje usług oferowanych w stacjach sieciowych stanowiących potencjalną lukę w bezpieczeństwie, począwszy od serwerów WWW, FTP, e-mail poprzez usługi DNS, SMB i wiele innych.

Producent udostępnia program bez konieczności uiszczania opłat licencyjnych, nie ponosi jednak żadnej odpowiedzialności za błędne lub nieuczciwe jego wykorzystywanie. Celem projektantów pakietu *SAINT* było wykonanie narzędzia grupującego wiele procedur testujących stopień zabezpieczenia systemu. Ułatwiono dostęp do najnowszych informacji o błędach w oprogramowaniu, mogących zwiększyć ryzyko włamania do systemu. *SAINT* działa w graficznym środowisku użytkownika X Windows. Interfejs programu stanowi przeglądarka stron WWW (np. *Netscape*), poprzez którą obsługuje się go i otrzymuje wyniki funkcjonowania.

1.3. Skaner *NMAP*

The Network Mapper jest kolejnym skanerem portów udostępniany w narzędzie *Open Source*, służy zarówno do skanowania portów w rozległych *STI*, jak i pojedynczych stacji roboczych opartych o systemy Unix'owe i Windows'owe. Skaner znajduje zastosowanie dla różnych protokołów sieciowych (m.in. UDP, TCP, ICMP). Głównym zadaniem programu jest wykrywanie oferowanych usług, np. FTP, POP3 oraz określanie numerów portów, na których działają (Rys. 1). Cechą decydującą o wyjątkowej przydatności *nmap'a* jest możliwość wyboru jednej z wielu technik skanowania łącznie z ukrywającymi fakt skanowania (tzw. *stealth*).

Programu używa się między innymi do identyfikowania rodzaju i wersji systemu operacyjnego zainstalowanego na danym systemie sprzętowym (ang. *TCP/IP fingerprinting*) Program automatycznie również sprawdza, czy komputery znajdujące się w danej podsieci są uruchomione (ang. *ping*). Ważną funkcją narzędzia jest określanie rodzaju ściany ogniowej zainstalowanej w systemie.



Rys. 1. Graficzny interfejs skanera nmap

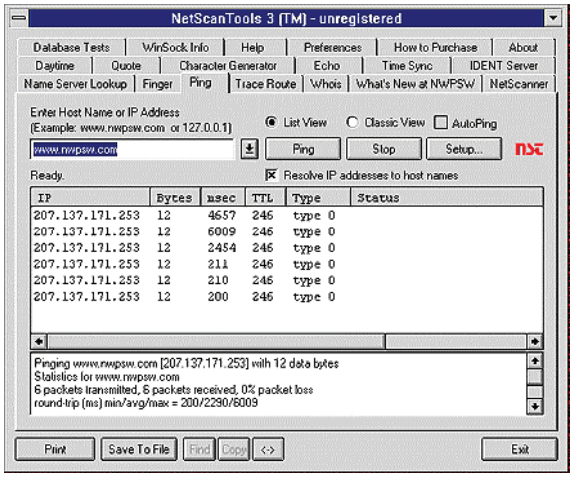
Dzięki różnorodności technik skanowania (od *TCP Connect* do *Stealth*) ustala się reguły przyjmowania i/lub odrzucaniau przychodzących pakietów.

1.4. Skaner *NETSCANTOOLS*

NetScanTools jest narzędziem służącym do skanowania w systemie Windows. Jest to produkt komercyjny, jednak ze strony producenta można pobrać 30-dniową standardową wersję próbną. Skaner ten jest wygodną graficzną nakładką (Rys. 2) dla wielu dostępnych za darmo narzędzi obsługiwanych z wiersza poleceń, takich jak *ping, traceroute, finger* oraz *whois* i *fwhois*. Program wykorzystuje także kilka starszych usług, takich jak *echo, daytime, quote* i *chargen*, do których dostęp jest blokowany w większości systemów dołączonych do Internetu. Dodatkowo zaimplementowano wiele z narzędzi / podprogramów takich jak *TimeSync*,

Database Tests, *WinSock Info*, *NetBios Info* oraz *Ident Server*. *NetScanTools* posiada dużą funkcjonalność przez wygodny i zintegrowany pakiet narzędzi do zdobywania istotnych informacji sieciowych oraz przeprowadzania skanowania sieci. Do najbardziej przydatnych narzędzi należy:

- *NetScanner* - skanowanie wskazanego przedziału adresów i odszukiwanie działających sieci i stacji roboczych,
- *Port Scanner* - właściwe skanowanie portów wraz z „łapaniem nagłówków" (ang. *banner grabbing*),
- *TCP Term* - bezpośrednie łączenie się ze wskazanym portem na wskazanym zdalnym systemie i komunikowanie się z nasłuchującą na tym porcie usługą.



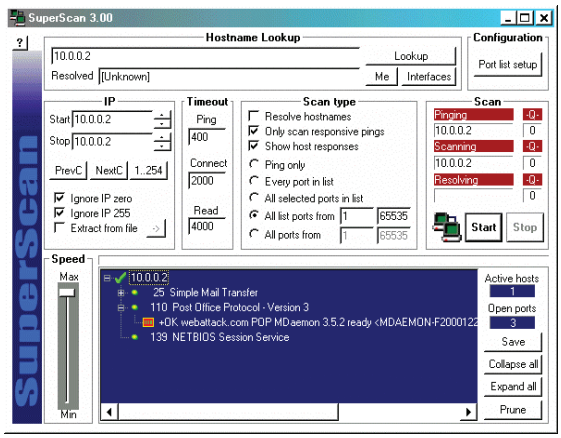
Rys. 2. Graficzny interfejs skanera NetScanTools

1.5. Skaner *SUPERSCAN*

SuperScan jest kolejnym graficznym narzędziem skanującym funkcjonującym także w systemie Windows. W przeciwieństwie do programu *NetScanTools*, jest to darmowe narzędzie *Open Source*, umożliwiające skanowanie portów UDP i TCP oraz zdobywanie dodatkowych informacji pozwalający na identyfikację diagnozowanego systemu. Zwiększona funkcjonalność ograniczyła jednak możliwości stosowania programu *SuperScan* do nowszych systemów operacyjnych Windows[3].

Wygląd interfejsu użytkownika aktualnej wersji programu *SuperScan* posiada zakładki oddzielających funkcje oraz ekrany przeznaczone do konfiguracji (Rys. 3).

[3] *SuperScan* nie działa w systemach Windows 95 i 98.



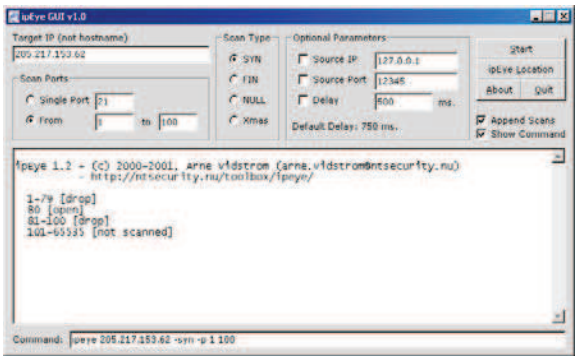
Rys. 3. Graficzny interfejs skanera SuperScan

SuperScan używa się do skanowania portów, pozyskiwania ogólnych informacji o sieci (np. nazw stacji roboczych, danych o trasowaniu) oraz szczegółowych informacji o zdalnych komputerach z systemem Windows (z danymi o użytkownikach, grupach i usługach włącznie).

1.6. Skaner *IPEYE SKANER*

IPEye jest skanerem portów uruchamianym z wiersza poleceń systemów Windows 2000/XP. Program udostępnia takie same tryby ukrytego skanowania TCP jak *nmap* (ze skanowaniem *SYN*, *FIN*, *Xmas tree* oraz *null*). *IPEye* jest darmowy i posiada małe wymagania sprzętowe. Wadą narzędzia jest możliwość uruchamiania tylko w systemach Windows.

Opcje programu *IPEye* (Rys. 4) są podobne do możliwości innych skanerów omówionych powyżej. Można w nim wydłużyć czas skanowania oraz zmodyfikować źródłowy adres IP i numer portu (chociaż nie są obsługiwane bardziej wyszukane opcje podszywania się pod inny adres).



Rys. 4. Graficzny interfejs skanera IPEyeScanner

2. SKANERY SŁABYCH PUNKTÓW

W ogólności, skaner słabych punktów składa się z modułu skanującego i katalogu. Katalog zawiera listę najczęściej spotykanych plików, plików znanych z wrażliwości na atak oraz listy często spotykanych luk w zabezpieczeniach dla wielu współczesnych serwerów i host'ów. Skaner słabych punktów obsługuje odczytywanie katalogu słabych

punktów, wysyłanie żądań do badanego systemu oraz interpretowanie otrzymywanych odpowiedzi celem określenia, czy badany system jest podatny na atak. Zadaniem tych narzędzi jest zwykle wykrywanie słabych punktów, które mogą być łatwo usunięte przez poprawienie konfiguracji komputera, zainstalowanie uaktualnień i wyczyszczenie odpowiednich katalogów.

2.1. Skaner *NIKTO*

Program *Nikto* jest skanerem funkcjonującym na podstawie języka Perl, dzięki czemu działa w systemach Unix, Windows oraz Mac OS X. Narzędzie wykorzystuje standardowe biblioteki Perla zawarte w jego domyślnej instalacji. Narzędzie wymaga także biblioteki *LibWhisker*. Już od pierwszej wersji program oferował obsługę *Secure Sockets Layer*, serwerów pośredniczących (ang. *proxy*) oraz funkcję skanowania portów.

Moduł skanujący wykrywa potencjalne słabe punkty serwerów WWW i wyświetla dane wyjściowe wyjaśniające, dlaczego znalezione luki w systemie zabezpieczeń mogą stanowić zagrożenie.

Program wyposażono w wiele opcji - w większości przypadków opcje te poszerzają funkcjonalność narzędzia do tego stopnia, że wykracza ona poza możliwości typowe dla programów skanujących.

2.2. Skaner *STEALTH*

Stealth jest narzędziem do skanowania słabych punktów wykorzystującym graficzny interfejs użytkownika systemu Windows. Zaleta programu *Stealth* leży w liczbie przeprowadzanych testów i w łatwości aktualizowania jego bazy danych. Testy dotyczą rozmaitych elementów: od adresów URL łamiących zabezpieczenia urządzeń z wbudowanymi serwerami WWW po ostatnio odkryte luki w serwerach IIS.

Narzędzie *Stealth* próbuje uprościć proces wykrywania słabych punktów, udostępniając narzędzie pomocnicze, *Stealth Exploit Development Tool* - program z graficznym interfejsem użytkownika,
w którym należy podać wartość dla każdego możliwego pola wykorzystywanego podczas konstruowania testu słabych punktów.

Kolejną ciekawą techniką wykorzystywaną przez *Stealth* jest test przepełnienia bufora. Tego typu atak można przeprowadzić przeciwko dowolnemu adresowi URL udostępnianemu przez aplikację WWW, dla którego definiuje się listę parametrów.

Większość testów przeprowadzanych przez narzędzie *Stealth* opiera się na zwrotnych kodach HTTP generowanych przez skanowany serwer. Takie rozwiązanie jest korzystne, gdy sprawdza się, czy na serwerze istnieją źle zabezpieczone skrypty, jednak nie zawsze daje wiarygodną odpowiedź na pytanie, czy dany skrypt rzeczywiście jest podatny na atak.

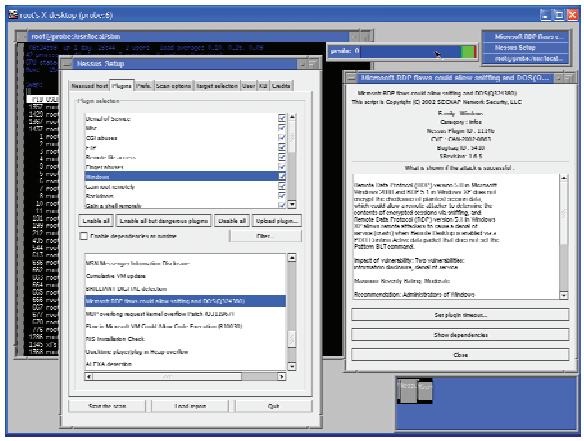
2.3. Skaner *NESSUS*

NESSUS to narzędzie dostępne na platformie Unix na zasadach *Open Source*. Działanie *Nessus'a* opiera się na architekturze klient-serwer (Rys. 5), a program składa się z dwóch części:

- systemu (ang. *nessusd*) - będąca serwerem, może zostać zainstalowana na systemie zdalnym, wykonuje ona większą część zadań wysyłanych do niej przez klienta,
- klienta - z tego poziomu wydaje się polecenia i łączy się z serwerem za pomocą protokołu TCP/IP. Klient może być uruchamiany na tej samej maszynie lub na innej, zlokalizowanej w obrębie sieci (lub tez poza nią). Korzystanie z serwera wymaga autoryzacji przy użyciu systemu jednorazowych haseł i kluczy.

Wraz z narzędziem dostarczany jest bardzo zaawansowany graficzny interfejs użytkownika, znacznie ułatwiający pracę. W praktyce najczęściej wykorzystuje się do testów narzędzie *NessusWX* - klienta *Nessus'a* w wersji dla MS Windows. Powodem jego wyboru, w przeciwieństwie do klienta uniks'owego, jest jego lepsza organizacja i ergonomia użycia.

Nessus umożliwia szczegółowe określanie pojedynczego rodzaju wykonywanego testu lub wybranie całej grupy testów (np. *Denial of Service*) lub tylko niektórych, istotnych dla badanego systemu.



Rys. 5. Konsola programu *Nessus*

Nessus może współpracować z trzema zalecanymi programami znacznie zwiększającymi funkcjonalność testów (*nmap, Hydra, nikto*).

3. PROGRAMY *SNIFFINGU*

Sniffer to program przełączający kartę sieciową w tryb bezwładny i nasłuchujący pracę w sieci. Przy jego pomocy możemy wywołać: przełączenie interfejsu w tryb bezładny, nasłuchiwanie, zapisywanie i wyświetlanie nasłuchiwanego ruchu TCP. Najpopularniejsze programy to: *TCPDUMP I WINDUMP, ETHEREAL, DSNIFF*, *ETTERCAP*, itp.

Niebezpieczeństwo dla systemu ze strony *sniffer'ów* jest bardzo duże, gdyż można tą drogą

przechwycić hasła. A jak wiadomo dostanie się do systemu nawet jako zwykły użytkownik to połowa sukcesu dla włamywacza. Ataki przeprowadzane przy użyciu tego rodzaju programów to najgroźniejsze i najbardziej niebezpieczne ataki na system i poufne informacje.

Wykrycie *sniffer'a* nie jest łatwe, ponieważ najczęściej nie zostawiają one żadnych śladów - działają na komputerze atakującego i tylko przyjmują pakiety - żadnych danych podczas prowadzenia nasłuchu nie wpuszczają do sieci (chyba, że komputer zostanie zapytany).

Sprawdzenia, czy dana karta pracuje w trybie bezładnym można dokonać posługując się poleceniem *ifconfig*. Aby sprawdzić wszystkie karty pod względem trybu pracy można też posłużyć się poleceniem *ifstatus*. W przypadku ustawionego interfejsu w tryb bezładny otrzymamy stosowny komunikat. Poważną wada tego rozwiązania jest konieczność sprawdzania ustawienia każdej stacji roboczej.

Dobrym rozwiązaniem jest użycie programu *Network Promiscuous Detector* (*NEPED*). *NEPED* skanuje sieć w poszukiwaniu interfejsów znajdujących się w trybie *promiscous*. W tym celu *NEPED* wykorzystuje błąd w implementacji *arp*. W jądrach linux'owych powyżej 2.0.36 naprawiono już ten błąd.

Najlepszą ochroną przed *sniffer'ami* jest zastosowanie szyfrowania zarówno transmisji *loginu* i hasła jak i samych danych.

3.1. Sniffer'y *TCPDUMP* i *WINDUMP*

Program *tcpdump* jest *sniffer'em* pakietów dla systemu Unix, uruchamianym w wierszu poleceń. *WinDump* jest odpowiednikiem *tcpdump* dla systemów Windows i ma niemal taką samą funkcjonalność jak *tcpdump*. *Tcpdump* został napisany ściśle pod kątem monitorowania sieci, analizowania i testowania ruchu sieciowego oraz przechwytywania pakietów.

Tcpdump jest popularnym analizatorem protokołów sieciowych wymagającym *kernel'a* z opcją *Berkeley Packet Filter* i urządzenia z dostęp do */dev/bpf*. Właściwe funkcjonowanie wymaga zdefiniowania interfejsu zbierania pakietów i miejsca ich składowania. Zaletą *tcpdump* jest możliwość konstruowania warunków logicznych sprawdzający określoną właściwość pakietu. Jeśli wynik sprawdzenia wzorca to logiczna prawda - pakiet zostaje zarejestrowany.

Podstawowy typ skanowania *tcpdump* to wysłanie pustego pakietu UDP (*udp 0*) na port i oczekiwanie na odpowiedź w postaci komunikatu ICMP.

Tcpdump i *WinDump* to właściwie bardziej analizatory pakietów sieciowych niż *sniffer'y*. Ich możliwości filtrowania pakietów są znacznie większe niż wielu innych dostępnych narzędzi, ale mechanizm przechwytywania danych z pakietów nie jest najłatwiejszy w użyciu. Programy pozwalają uzyskać wiele interesujących niskopoziomowych informacji o pakietach przechodzących przez sieć i mogą pomóc zdiagnozować wszelkiego rodzaju słabe punkty a także przechwycić informacje wrażliwe.

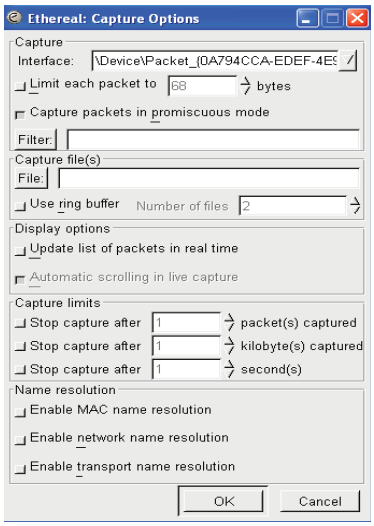
Oba narzędzia, *tcpdump* i *WinDump*, używają biblioteki *pcap*, zestawu procedur służących do przechwytywania pakietów. Procedury biblioteki *pcap* udostępniają interfejs i zestaw funkcji umożliwiających filtrowanie pakietów na poziomie systemu operacyjnego oraz demontowanie pakietów IP na dane surowe.

Analiza wyników dostarczanych przez *tcpdump* jest bardzo trudna, dlatego też rzadko się z niego korzysta. Jednakże *tcpdump* stał się standardem (zarówno wśród *sniffer'ów*, jak i sieciowych systemów detekcji intruzów). Oparta jest na nim większość narzędzi analizy sieciowej posiadających interfejs graficzny. Format zapisu pakietów za pomocą *tcpdumpa* jest rozpoznawany przez wiele narzędzi. Co więcej, filtry z jakich korzysta ten pakiet, zostały także zaakceptowane przez innych twórców. Efektem funkcjonowania pakietu jest kompatybilność wielu narzędzi sieciowych z *tcpdumpem*.

3.2. *ETHEREAL*

Ethereal to graficzny interfejs do odczytywania plików z wynikami przechwytywania pakietów, tworzonych przez kilka różnych *sniffer'ów* pakietów, z *tcpdump* i *WinDump* włącznie. Potrafi samodzielnie i w czasie rzeczywistym przechwytywać pakiety, przy użyciu narzędzia *tethereal* i biblioteki *pcap*. Wykorzystując *Ethereal* na stworzonych wcześniej plikach z przechwyconymi danymi, można przeglądać szczegóły przechwyconej sesji, łącznie z danymi pakietów.

Ethereal jest dostępny za darmo dla systemów Windows, Unix i niektórych systemów operacyjnych dla komputerów Macintosh. Możliwości Ethereala są duże, gdyż program rozpoznaje w zasadzie wszystkie popularne formaty zapisu pakietów. Program umożliwia analizę strumienia TCP, co jest bardzo pomocne podczas dekodowania sesji TELNET, SMTP, FTP czy HTTP. Pakiet ułatwia także zbieranie i analizowanie statystyk ruchu.



Rys. 6.Okno parametrów przechwytywania

Ethereal składa się z kilku wbudowanych narzędzi, zainstalowanych domyślnie w systemach Unix i Windows:

- *Tethereal* - wersja programu *Ethereal* pozbawiona graficznego interfejsu, wykorzystywana w systemach bez środowiska X Windows lub w Win32,
- *Editcap* - pozwala na modyfikację plików zawierających przechwycone pakiety. Niestety możliwości edycji są bardzo ograniczone. Można wybrać liczbę pakietów, jakie mają się znaleźć w pliku wynikowym, zmodyfikować czas ich przechwycenia,
- *Mergacap* - umożliwia konsolidację plików zawierających przechwycone pakiety w jeden plik, pakiety są włączane do pliku wynikowego chronologicznie,
- *Text2pcap* - wczytuje dane w formacie szesnastkowym, zapisanym jako znaki ASCII, i konwertuje je do formatu *lipcap*, umożliwia ponadto uzupełnienie danych o brakujące nagłówki, np. IP czy UDP.

3.3. *DSNIFF*

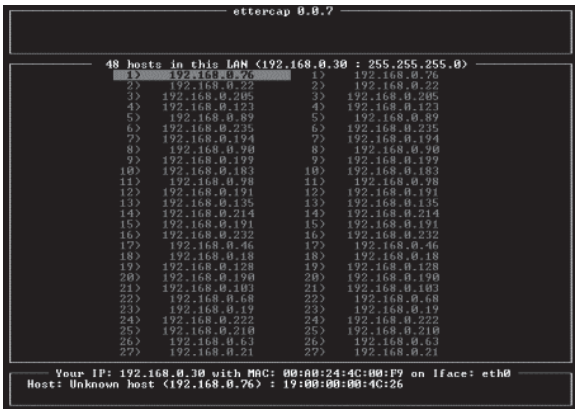
Dsniff jest zestawem darmowych narzędzi, które zostały oryginalnie stworzone dla celów testowania sieci i możliwości jej infiltracji:

- *Arpspoof* umożliwia podsłuchiwanie przełączanej sieci przez podrabianie odpowiedzi ARP dla docelowej stacji roboczej,
- *Dnsspoof* działa podobnie do *arpspoof*. Pozwala podrobić odpowiedzi DNS dla serwera DNS znajdującego się w sieci lokalnej,
- *Dsniff* jest zaawansowanym *sniffer'em* haseł - używanym, gdy chcemy uzyskać tylko nazwy użytkowników i hasła,
- *Macof* zasypuje lokalną sieć losowymi, wymyślonymi adresami MAC (atak typu *Flood*) w nadziei, że spowoduje awarię przełącznika i zacznie on działać jako koncentrator, dając programowi *dsniff* większe szansę w środowisku sieciowym wykorzystującym przełączniki,
- *Mailsnarf* składa z powrotem podsłuchane wiadomości poczty elektronicznej z protokołów SMTP i POP,
- *Sshmitm* jest jednym z bardziej złośliwych narzędzi dostępnych w zestawie *dsniff*. sshmitm (*SSH Man in the Middle*) może podsłuchiwać ruch SSH przekierowywany do komputera intruza. Obsługuje tylko SSH w wersji 1,
- *Tcpkill* próbuje przerwać trwające połączenie TCP przez sfałszowanie pakietu resetującego i wstrzyknięcie go do prawdziwego połączenia,
- *Tcpnice* pozwala spowolniać połączenia. Wykorzystuje te same opcje, co *tcpkill*,
- *Urlsnarf* działa tak samo jak wszystkie inne programy do przechwytywania w tym zestawie narzędzi, z wyjątkiem tego, że działa dla sieciowych adresów URL,
- *Webmitm* wykonuje dla HTTPS (ruch WWW z szyfrowaniem SSL) to samo, co *sshmitm* dla SSH,
- *Webspy* z pakietu *dsniff* umożliwia podsłuchiwanie ruchu WWW pochodzącego ze wskazanego komputera. Za każdym razem, kiedy komputer skieruje się do nowego adresu URL, *webspy* załaduje ten sam URLw przeglądarce intruza.

3.4. *ETTERCAP*

Program *Ettercap* działa w systemach Linux, BSD, Solaris 2.x, większości odmian Windows (Rys. 7) i Mac OS X. *Ettercap* tworzenie własnych modułów dodatkowych. Te moduły mogą służyć do rozszerzenia możliwości *ettercap*.

Uruchomienie *ettercap* bez żadnych opcji, powoduje zainicjowanie skanowania *ARP* badanej sieci LAN i wyświetlenie tabeli wszystkich komputerów znalezionych w sieci LAN. Podsłuchiwanie z wykorzystaniem *ARP* jest wizytówką programu *ettercap*. Ten tryb wymaga wybrania przynajmniej źródła albo celu.



Rys. 7. Konsola programu EtherCap

Umożliwia on przechwytywanie ruchu, nawet w sieci opartej na przełącznikach. *Ettercap* podrobi pakiety *ARP* docelowego komputera w taki sposób,

że każde żądanie *ARP* dla adresu IP wybranego celu otrzyma w odpowiedzi adres MAC komputera podsłuchującego, co pozwoli na przechwycenie ruchu przez *sniffer* zanim *ettercap* przekaże go dalej. Procedura zatruwania ARP wykorzystywana przez *ettercap* może niekiedy spowodować zakłócenia w działaniu sieci LAN. Ze względu na swoją naturę, tryb podsłuchiwania z wykorzystaniem ARP w programie *ettercap* wykorzystuje atak typu „*man-in-the-middle*" podobny do tego, jaki *sshmitm* z narzędzi pakietu *dsniff*.

WNIOSKI

Z analizy zjawisk zachodzących w systemach teleinformatycznych wynika konieczność zapewnienia wymaganego poziomu bezpieczeństwa, poprzez realizację procesu diagnozowania, zarówno dla danych przechowywanych jak i znajdujących się w obiegu przed nieupoważnionymi osobami. uwzględniając aktualność i ważność tego problemu, na rynku teleinformatycznym, oferowanych jest wiele rozwiązań programowych i sprzętowych do badania bezpieczeństwa w systemach tego rodzaju. Dlatego też w artykule tym wymieniono i dokonano charakterystyki powszechnie wykorzystywanych darmowych narzędzi wyszukujących najmniej bezpieczne miejsca.

Jak wynika z analizy oferowanych rozwiązań, do badania bezpieczeństwa, istnieje obecnie duża grupa narzędzi, zarówno sprzętowych jak i programowych, pozwalających na wieloaspektowe diagnozowanie sieciowe. Są to głównie narzędzia pozwalające analizować zjawiska zachodzące sieci i badanie odporności na różnego rodzaju ataki.

Z powyższego wynika, że za pomocą dowolnego z przedstawionej grupy stosowanych w praktyce narzędzi (Tabela 1), konieczność podkreślenia jak nie mało wiedzy i umiejętności potrzeba posiadać, żeby spowodować nieodwracalne szkody w zakresie utraty poufności, integralności, dostępności informacji przechowywanej, przetwarzanej i przesyłanej we współczesnych systemach teleinformatycznych.

Tabela 1.

Zestaw wybranych narzędzi do zdobywania informacji o systemach TI

Narzędzie	Zastosowanie	Strona www
Ethereal	Analizowanie pakietów	http://www.ethereal.com
Tcpdump	Analizowanie pakietów (Linux)	http://www.tcpdump.org/
Windump	Analizowanie pakietów (Windows)	http://www.winpcap.org/windump/
Nmap	Wszechstronne skanowanie sieci	http://www.insecure.org/nmap/
Dsniff	Testowanie sieci - zestaw narzędzi	http://naughty.monkey.org/~dugsong/dsniff/
Ngrep	Filtrowanie zawartości pakietów	http://ngrep.sourceforge.net/
Nikto	Skanowanie podatności serwerów WWW	http://www.cirt.net/code/nikto.shtml
Amap	Rozpoznawanie usług	http://www.thc.org/thc-amap
POf	Pasywne narzędzie do fingerprintingu	
Cheops-ng	Tworzenie mapy sieci	http://cheops-ng.sourceforge.net/
Firewalk	Tester firewalli	http://www.packetfactory.net/firewalk/
Sing	Generowanie pakietów ICMP	http://www.sourceforge.net/projects/sing
Hping2	Generowanie pakietów	http://www.hping.org/
Fragroute	Testowanie zachowania firewalli, systemów IDS i stosu TCP/1P	http://www.monkey.org/~dugsong/fragroute/
Nessus	Skanowanie podatności	http://www.nessus.org/
WebServer FP	Rozpoznawanie serwerów WWW	http://www.computec.ch/request.php7457
N-Stealth	Skanowanie podatności serwerów WWW	http://www.nstalker.com/nstealth/
Nemesis	Generator pakietów	http://nemesis.sourceforge.neV
Kismet	Skanowanie sieci WiFi	http://www.kismetwireless.net/
NetStumbler	Wykrywanie sieci WiFi	http://www.netstumbler.com/
AirCrack	Testowanie sieci WiFi - zestaw narzędzi	http://freshmeat.net/projects/aircrack/

Ciągły i nieustanny rozwój nowych metod i narzędzi ataku oraz fakt ich ogólnej dostępności w zasobach sieci Internet powodują, że znacznie wzrasta ryzyko potencjalnego zagrożenia.

LITERATURA

[1] Shema M. & Bradley C. Johnson: *Anti-Hacker TOOL KIT*, *Edycja polska*, Helion, 2004.
[2] Fadia A., *Etyczny hacking: Nieoficjalny przewodnik*, Mikom, 2003.
[3] Klevinsky T. J., Laliberte S., etc.: *Testy bezpieczeństwa danych*, Helion, 2003.
[4] Erickson J., *Hackin: Sztuka penetracji*; Helion 2004.
[5] Liderman K.: *Podręcznik administratora bezpieczeństwa teleinformatycznego;* MIKOM, 2003.
[6] Tanger J., Lane P. T., Danielyan E.: *Hack Proofing Linux*; HELION, 2003.
[7] Praca zbiorowa: *Hack Proofing Network;* HELION, 2003.
[8] Witryny internetowe poświęcone poszczególnym narzędziom (tabela 1).

Mgr inż. **Ireneusz KRYSOWATY** pracownik Wydziału Elektroniki Wojskowej Akademii Technicznej i absolwent kierunku „Elektronika i telekomunikacja". Jego obszar zainteresowań obejmuje biometrię oraz bezpieczeństwo systemów IT.

Aktualnie uczestniczy w pracy badawczej dotyczącej sieciowych systemów ochrony w oparciu o TCP/IP.

Dr inż. **Dariusz LASKOWSKI** w 1997 roku ukończył Wydział Elektroniki Wojskowej Akademii Technicznej, gdzie obecnie pracuje w Instytucie Telekomunikacji na stanowisku asystenta naukowo - dydaktycznego. Członek *Polish Safety and Reliability Association*, stopień doktora nauk technicznych otrzymał w dyscyplinie telekomunikacja o specjalność sieci teleinformatyczne. Autor wielu publikacji krajowych i zagranicznych, współwykonawca sześciu prac naukowo-badawczych oraz osiemnastu autorskich opracowań *Programu Zapewnienia Niezawodności* urządzeń i systemów łączności.

Mgr inż. **Paweł NIEDZIEJKO,** pracownik Wydziału Elektroniki Wojskowej Akademii Technicznej i absolwent kierunku „Elektronika i telekomunikacja". Prowadzi wykłady i szkolenia z zakresu bezpieczeństwa systemów IT, telekomunikacji światłowodowej. zarządzania bezpieczeństwem. Jego zainteresowania skupiają się biometrycznym uwierzytelnianiu i podpisie cyfrowym w infrastrukturze klucza publicznego. Obecnie prowadzi pracę badawczą dotyczącą sieciowych systemów ochrony w oparciu o TCP/IP.