+CD Advanced Archive Password Recovery · Comdom antispam for Linux · Fast Report Server

nr 5/2008 (37) cena 29,80 PLN stawka VAT: 0% INDEKS: 382396 ISSN: 1731-7150

Hakin9 5/2008 (37)

HAKIN9

JAK SIĘ OBRONIĆ HARD CORE IT SECURITY MAGAZINE

Przemycanie ukrytych informacji
Format GIF okiem hakera

Phishing, DDos, Deface
Atak na reputację

Peach 2.0
Rozbudowany fuzzing

Suhosin
Bezpieczeństwo aplikacji WWW

FILM INSTRUKTAŻOWY
Includowanie kodu PHP z innego serwera
część 7

Mechanizmy przeciwdziałania zagrożeniom

HAKOWANIE PAKIETÓW BIUROWYCH

NA PŁYCIE CD

WERSJE BEZ OGRANICZEŃ CZASOWYCH
ADVANCED ARCHIVE PASSWORD RECOVERY
ADVANCED RAR PASSWORD RECOVERY
ADVANCED ZIP PASSWORD RECOVERY
COMDOM ANTISPAM FOR LINUX
FAST REPORT SERVER
WERSJE 3 MIESIĘCZNE
PANDA INTERNET SCURITY 2008
USERGATE

ISSN 1731-7150 05
9 771731 715051

PLUS Wywiad z Jackiem Pokraśniewiczem

M2 NET

NOWA WIZJA BEZPIECZEŃSTWA

SMI! Series

SECURE MAIL INTELLIGENCE!®

SMI! for Gateway

– chroni ruch SMTP dzięki kilku silnikom antywirusowym
oraz ponad 20 technikom antyspamowych.

SMI! for Domino

– wykorzystuje polityki kontroli treści, wiele silników antywirusowych
oraz modułów antyspamowych do ochrony IBM Lotus Domino / Notes

SMI! for Exchange

– wykorzystuje wiele silników antywirusowych oraz innowacyjnych metod antyspamowych
do ochrony Microsoft Exchange.

SMI! Managed Service

– umożliwia filtrowanie poczty klienta przez aplikacje rodziny SMI!
na bezpiecznym serwerze M2 NET SA.

SMI! Encryption Key Manager

– umożliwia zastosowanie silnych, bezpiecznych i kwalifikowanych kluczy szyfrujących,
zgodnych ze standardem x.509 w. 3, wewnątrz aplikacji IBM Lotus Domino / Notes.

www.bezpiecznapoczta.pl

SEPATON®

Wirtualne Biblioteki Taśmowe (VTL)

W obecnych środowiskach zabezpieczania danych, ochrona informacji znaczy o wiele więcej niż tylko backup.

To jest właśnie powodem, że wielu klientów zaufało rozwiązaniom wirtualnych bibliotek taśmowych SEPATON oraz technologii de-duplikacji danych. Unikalna architektura ContentAware ™ umożliwia zminimalizowanie ryzyka utraty danych oraz zwiększa możliwości operacyjne związane z ochroną danych.

Aby dowiedzieć się więcej na temat rozwiązań Sepaton, zapraszamy do kontaktu z firmą **S4E S.A. – Autoryzowanym Dystrybutorem firmy Sepaton.**

S4E
storage 4 enterprise

S4E S.A., ul. Wadowicka 8W, 30-415 Kraków, www.s4e.pl/sepaton

SPIS TREŚCI

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

Dyrektor wydawniczy: Sylwia Pogroszewska
Redaktor naczelny: Katarzyna Juszczyńska
katarzyna.juszczynska@software.com.pl
Redaktor prowadzący: Robert Gontarski
robert.gontarski@software.com.pl

Kierownik produkcji: Marta Kurpiewska
marta.kurpiewska@software.com.pl
DTP Manager: Robert Zadrożny
Okładka: Agnieszka Marchocka

Dział reklamy: adv@software.com.pl
Prenumerata: Marzena Dmowska
pren@software.com.pl

Wyróżnieni betatesterzy:
Rafał Łysik, Marcin Kulawinek
Opracowanie CD: Rafał Kwaśny

Druk: 101 Studio, Firma Tęgi
Nakład wersji polskiej 6 000 egz.

Adres korespondencyjny:
Software–Wydawnictwo Sp. z o.o.
ul. Bokserska 1, 02-682 Warszawa, Polska
Tel. +48 22 427 36 77, Fax +48 22 244 24 59
www.hakin9.org

Osoby zainteresowane współpracą prosimy o kontakt:
cooperation@software.com.pl

hakin9 ukazuje się w następujących krajach:
Hiszpanii, Argentynie, Portugalii, Francji, Belgii, Luksemburgu, Kanadzie, Maroko, Niemczech, Austrii, Szwajcarii, Polsce, Czechach, Słowacji.

Prowadzimy również sprzedaż kioskową w innych krajach europejskich.

Magazyn hakin9 wydawany jest w 7 wersjach językowych:
PL ES CZ EN IT FR DE

NARZĘDZIA

POCZĄTKI

ATAK

STAŁE RUBRYKI

W SKRÓCIE

ABY INTERNET BYŁ BEZPIECZNIEJSZY...

12 lutego 2008 już po raz czwarty obchodzony był *Dzień Bezpiecznego Internetu*, organizowany w Polsce przez NASK oraz Fundację Dzieci Niczyje. Głównym Partnerem przedsięwzięcia była Fundacja Grupy TP. Dzień Bezpiecznego Internetu (DBI), ustanowiony z inicjatywy Komisji Europejskiej w ramach programu *Safer Internet*, ma na celu propagowanie działań na rzecz bezpieczeństwa dzieci i młodzieży w Internecie. Wśród tegorocznych Partnerów DBI znaleźli się UPC, Związek Producentów Audio-Video, Komenda Główna Policji, agencja badawcza Gemius S.A. Akcję wspiera również szereg mediów ogólnopolskich i lokalnych. *Głównym wydarzeniem obchodów DBI w Polsce jest konferencja, która obędzie się w Bibliotece Uniwersyteckiej w Warszawie. Dzień Bezpiecznego Internetu jest okazją do podsumowań oraz do przedstawienia nowych działań podejmowanych w ramach projektu Saferinternet.pl. W tym roku pragniemy zwrócić szczególną uwagę na ofertę e-learningową dla szkół podstawowych oraz nową odsłonę kampanii „Dziecko w Sieci", realizowaną pod hasłem STOP cyberprzemocy* – zapowiada Agnieszka Wrzesień, koordynator Projektu Awareness z Fundacji Dzieci Niczyje.

RAPORT ROCZNY CERT POLSKA 2007

Zespół CERT Polska opublikował raport podsumowujący rok 2007, zawierający analizę incydentów naruszających bezpieczeństwo teleinformatyczne, zgłaszanych do zespołu CERT Polska w ubiegłym roku. Raport znajduje się pod adresem: *http://www.cert.pl/PDF/Raport_CP_2007.pdf*

CZTERY LATA WIĘZIENIA DLA FAŁSZERZA Z TAJWANU

Tajwański sąd skazał na cztery lata pozbawienia wolności lidera grupy przestępczej Huang'a Jer-sheng'a, odpowiedzialnego za 90 proc. pirackich

kopii produktów koncernu z Redmond na światowym rynku. Podrabiane oprogramowanie sprzedawane było w przynajmniej 22 krajach całego świata: Australii, Austrii, Kanadzie, Chinach, Chorwacji, Etiopii, Francji, w Niemczech, Irlandii, we Włoszech, w Malezji, Paragwaju, na Filipinach, w Katarze, Singapurze, Hiszpanii, Szwajcarii, na Tajwanie, w Trynidadzie i Tobago, Wielkiej Brytanii, Stanach Zjednoczonych. Pirackie kopie trafiały również na rynek polski, gdzie były rozprowadzane głównie za pośrednictwem internetowych serwisów aukcyjnych.

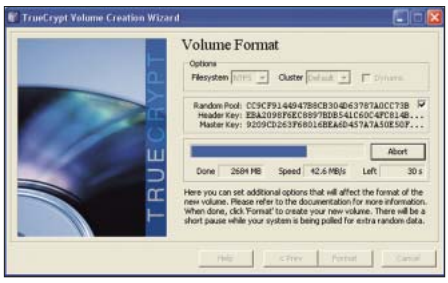
Trzej współpracownicy Huang'a Jer-sheng'a zostali skazani na kary od osiemnastu miesięcy do trzech lat pozbawienia wolności.

Wyroki, które zapadły w procesie na Tajwanie mogą być przestrogą dla potencjalnych fałszerzy oprogramowania komputerowego, ale jednocześnie mamy nadzieję, że zwrócą uwagę klientów na problem. Mamy nadzieję, że wydarzenia ostatnich tygodni przyczynią się do tego, że klienci będą bardziej uważnie dokonywać zakupów oprogramowania, omijając oferty na portalach aukcyjnych, a zamiast tego zwracając się do autoryzowanych dystrybutorów oprogramowania, dzięki czemu będą mogli uniknąć oszustwa. Zawsze podkreślamy, że użytkowanie oprogramowania pochodzącego z nielegalnego źródła jest nie tylko niezgodne z prawem, ale niesie również ze sobą ryzyko związane z narażeniem na niebezpieczeństwo własnych komputerów i danych w nich przechowywanych – powiedział Krzysztof Janiszewski, odpowiedzialny za ochronę własności intelektualnej w polskim oddziale Microsoft.

Na całym świecie piraci komputerowi narażają producentów oprogramowania na straty rzędu 40 miliardów dolarów rocznie. Szacuje się, że ponad połowa programów specjalistycznych, używanych w księgowości i kadrach polskich firm, jest nielegalna.

TRUECRYPT 5.0 WYDANY

Pojawiło się nowe wydanie programu TrueCrypt – w pełni darmowej aplikacji do szyfrowania danych. TrueCrypt jest darmowym oprogramowaniem Open Source dla systemów Microsoft Windows 2000/XP/2003 oraz UNIXów, umożliwiającym szyfrowanie całych partycji dyskowych *w locie*. Oprogramowanie wraz z kodem źródłowym udostępnione jest na licencji TRUECRYPT LICENSE przez TrueCrypt Foundation. Program TrueCrypt umożliwia m. in. utworzenie wirtualnego, zaszyfrowanego dysku, który widoczny będzie obok innych dysków zainstalowanych w komputerze, a także zaszyfrowanie całej partycji wybranego dysku lub innego urządzenia do magazynowania danych. TrueCrypt może korzystać z następujących symetrycznych algorytmów szyfrujących: AES-256, Blowfish (klucz 448-bitowy), CAST5 (CAST-128), Serpent (klucz 256-bitowy), Triple DES (3DES), Twofish (klucz 256-bitowy).



NOWY ALGORYTM SZYFRUJĄCY AL-KAIDY

Na jednym z forów internetowych związanych z Al-Kaidą ukazała się informacja, że zakończono prace nad nową wersją algorytmu szyfrującego, przeznaczonego dla użytkowników tego forum i znanego jako *Sekret Mudżahedina* – teraz w wersji 2. Autorzy kodu, którzy pozostają anonimowi, twierdzą, że wersja druga aplikacji jest o wiele bezpieczniejsza od poprzedniej.

Pierwsze wydanie Mujahideen Secrets wykorzystywało niezbyt dobry algorytm szyfrujący, było słabo zaprojektowane i łatwo było je złamać. Jeszcze nie miałem okazji zbadać wersji 2.0, ponieważ do jej pobrania niezbędne jest podanie hasła – jestem jednak pewien, że mogła ona zostać daleko poprawiona – powiedział Paul Henry, specjalista z firmy Secure Computing.

MICROSOFT CHCE PRZEJĄĆ YAHOO! ZA 44,6 MLD DOLARÓW

Firma Microsoft – największy światowy producent oprogramowania komputerowego – myśli nad kupnem Yahoo!, jednej z największych wyszukiwarek internetowych a zarazem firmy oferującej mnóstwo usług on-line. Gigant z Redmond złożył zarządowi Yahoo! oficjalną propozycję przejęcia wszystkich akcji będących w obiegu po cenie 31 dolarów za akcję, co razem daje sumę ponad 44,6 miliarda. Microsoft chce w ten sposób uzyskać dostęp do 130 milionów internautów odwiedzających miesięcznie portal Yahoo.

Mamy ogromny szacunek dla Yahoo! i razem jesteśmy w stanie zaoferować nowy, ekscytujący zestaw rozwiązań dla klientów, wydawców i reklamodawców. Wierzymy, że to połączenie podniesie wartość firm w oczach akcjonariuszy, a klientom i partnerom zapewni lepszy wybór i większą innowacyjność – mówił Steve Ballmer, dyrektor zarządzający Microsoftu.

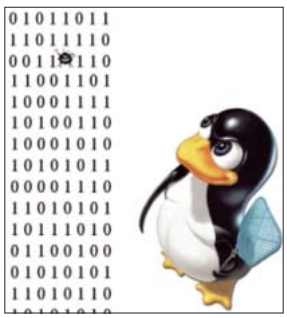
Jeśli transakcja dojdzie do skutku, będzie zdecydowanie największym przejęciem w historii Microsoftu i dokona trwałych zmian na szerokim rynku internetowym. Thomas Radinger, zarządzający Pioneer Investments w

Monachium uważa, że: *Microsoft jest pod potężną presją, by rozwijać swój internetowy biznes i odeprzeć ataki konkurentów takich jak Google, a ta oferta pokazuje, jak bardzo jest zdesperowany.*

Firma Yahoo! z siedzibą w Sunnyvale w stanie Kalifornia (USA) jest jednym z najpopularniejszych i największych serwisów internetowych na świecie, posiadającym wersje w kilkunastu językach. Twórcami Yahoo! byli David Filo i Jerry Yang – doktoranci fizyki na Wydziale Inżynierii Elektrycznej Uniwersytetu w Stanford. Obecnie Yahoo! świadczy internautom następujące usługi: e-mail, komunikator internetowy, radio internetowe, wyszukiwarka i katalog internetowy, weblog i chat.

LINUX 2.6.24 DOSTĘPNY

Pojawiła się nowa wersja jądra systemu GNU/Linux serii 2.6, oznaczona jako 2.6.24. Nowy kernel niesie ze sobą dziesiątki znaczących poprawek, których całkowitą listę zobaczyć można w ChangeLog. W nowej wersji jądra wprowadzono mechanizm zapobiegania fragmentacji pamięci, a także rozszerzono obsługę tzw. funkcji tickless na architektury x86-64, PPC, UML, ARM i MIPS. Programiści poprawili usprawnienia w zarządcy CFS, mające na celu jeszcze lepszą interakcyjność systemu. Dodano obsługę zarządzania zasilaniem SATA oraz funkcję dostępu w trybie tylko do odczytu do systemów plików zamontowanych w trybie zarówno odczytu, jak i zapisu.



WINDOWS SERVER 2008 TRAFIŁ DO TŁOCZNI

Po prawie pięciu latach rozwoju najnowszy, serwerowy system operacyjny firmy Microsoft wszedł w fazę RTM, która wyznacza kolejny ważny krok na drodze ku największej w historii firmy premierze produktów dla przedsiębiorstw. *Już od 4 lutego udostępniliśmy w Polsce wersję RTM Windows Server 2008. Obecnie jest on dostępny dla klientów korzystających z programu Connect, ale już wkrótce wersja ewaluacyjna pojawi się na witrynie Microsoft.com dla wszystkich osób zainteresowanych poznaniem najnowszej wersji systemu Windows Server* – powiedział Dariusz Korzun, Product Marketing Manager w polskim oddziale Microsoft. Oficjalna premiera systemu Windows Server 2008 odbędzie się 27 lutego w Los Angeles. WS2008 trafi do sprzedaży na początku marca. Ponieważ system Windows Server 2008 był opracowywany równolegle z kodem systemu Windows Vista, wyposażony jest w większość spośród zaawansowanych funkcji administracji tego systemu.

Nowa wersja przynosi sporo ulepszonych narzędzi do ochrony systemu. Jednym z najważniejszych jest Windows Firewall, którego zadaniem jest filtrowanie ruchu sieciowego przychodzącego oraz wychodzącego z systemu Windows Server 2008, co wyklucza takie niebezpieczeństwa, jak: próby nawiązania połączenia z systemami zewnętrznymi wywołane przez wirusy i robaki, które przeniknęły do systemu, próby zainfekowania kolejnych komputerów sieciowych lub próby rozsyłania spamu. Nowa zapora zawiera pokaźną grupę dodatkowych rozszerzeń oraz funkcji, jedną z nich jest możliwość automatycznego blokowania nieautoryzowanych pakietów wysyłanych przez serwer. Kolejną nowością jest integracja zapory z protokołem IPSec, który umożliwia ochronę ruchu sieciowego przez

W SKRÓCIE

weryfikację integralności pakietów oraz szyfrowanie danych. Technikę tę warto zastosować wobec skanerów wyszukujących słabe punkty w systemie oraz aplikacji skanujących inne programy, komputery oraz sieci w poszukiwaniu luk w bezpieczeństwie (jako przykład można tu podać skaner portów).

SKANDAL WOKÓŁ SPRZEDAŻY BILETÓW

To, że zawiódł internetowy system sprzedaży biletów na tegoroczne finały piłkarskich mistrzostw Europy, to jeszcze nic. Właśnie okazało się, że do Internetu trafiły tysiące danych osobowych kibiców, którym jednak udało się zarejestrować na stronie *kupbilet.pl* – informuje radio ZET. Rzecznik PZPN Zbigniew Koźmiński przyznał Radiu ZET, że operator serwisu miał problem z zabezpieczeniami. *Dostaliśmy taki sygnał od firmy nadzorującej przyjmowanie zamówień. To było w nocy, w pierwszych godzinach działania serwisu. Najpierw zablokowano na pewien czas cały serwis, a po usunięciu awarii strona działała już bez zarzutu i żadne dane nie były widoczne* – tłumaczył Koźmiński. Nie wiadomo ile danych wyciekło z systemu i w czyje ręce trafiły. Sprawą zajmie się Generalny Inspektor Ochrony Danych Osobowych.

NIELETNI HANDLOWAŁ W SIECI PIRACKIMI PŁYTAMI

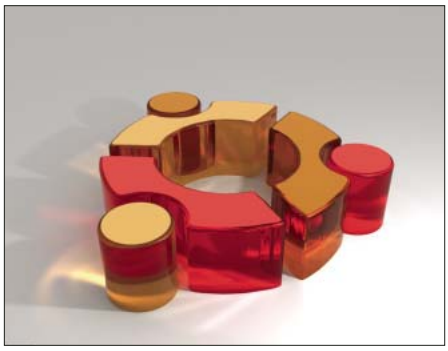
Dwa komputery i ponad 400 płyt DVD z nielegalnymi kopiami filmów zabezpieczyli policjanci w domu 16-letniego mieszkańca powiatu krośnieńskiego. Nastolatek pirackie płyty sprzedawał na zamówienie lub za pośrednictwem internetowych portali aukcyjnych po kilka złotych za sztukę. Chcąc podnieść swoją wiarygodność w Internecie, chłopak zawierał fikcyjne transakcje z kolegami, a później wymieniał się z nimi pozytywnymi opiniami. Przyjmując przelewy za sprzedane płyty, 16-latek korzystał z konta bankowego swojej mamy, do którego miał upoważnienie. Sprawa została przekazana do sądu rodzinnego i nieletnich, który zadecyduje o dalszym losie chłopca.

ROSJA NAJWIĘKSZYM PRODUCENTEM SZKODLIWEGO KODU

Jak wynika z najnowszego raportu firmy PC Tools, Rosja przoduje w produkowaniu szkodliwego oprogramowania – 28% światowego spamu pochodzi właśnie z Rosji. Chiny są źródłem 26,5% tego typu programów, a trzecie miejsce, z wynikiem poniżej 10% zajmują Stany Zjednoczone. Cieszyć może fakt, że Polska nie znalazła się w czołówce. Pierwsza dziesiątka największych wydawców szkodliwego oprogramowania wygląda następująco: 1. Rosja – 27,89%, 2. Chiny – 26,52%, 3. USA – 9,98%, 4. Brazylia – 6,77%, 5. Ukraina – 5,45%, 6. Wielka Brytania – 5,34%, 7. Francja – 3,81%. 8. Niemcy – 2,14%. 9. Szwecja – 1,60%, 10. Hiszpania – 1,37%.

UBUNTU ALPHA 5 JUŻ JEST

Pojawiło się piąte wydanie Alpha systemu Ubuntu 8.04 Hardy Heron. W nowej edycji systemu Ubuntu znajdziemy m.in. Brasero – program służący do nagrywania płyt CD/DVD, Transmission – klient BitTorrent, PulseAudio – serwer audio oraz PolicyKit – całkiem nową zaporę sieciową. Wydanie finalnej wersji 8.04 planowane jest na koniec kwietnia. Osoby, które już teraz chciałby przetestować nową wersję mogą pobrać obrazy ISO dowolnej wersji dystrybucji: Ubuntu, Kubuntu, Edubuntu, Ubuntu JeOS, Xubuntu, Gobuntu czy UbuntuStudio.



SKUTECZNOŚĆ SZYFROWANIA DANYCH PODWAŻONA

Dane zaszyfrowane za pomocą wyspecjalizowanych programów stworzonych w celu ochrony prywatnych informacji (np. Microsoft's BitLocker i Apple's FileVault), mogą zostać odzyskane poprzez odtworzenie klucza szyfrującego z ulotnej pamięci RAM. Testy przeprowadzone przez grupę badaczy, w skład której wchodzili m.in. członkowie e Electronic Frontier Foundation i naukowcy z Uniwersytetu w Princeton i wykazały, że używając dość niekonwencjonalnego sposobu można bez większych problemów odszyfrować dane.

DZIURA W VMWARE

Specjaliści z Core Security Technologies wykryli poważną lukę w oprogramowaniu VMware, która pozwala na nieograniczony dostęp do systemu macierzystego. Błąd

występuje w środowiskach Windows w momencie aktywacji *opcji obsługi folderu współdzielonego*. Na skutek błędu wirtualizowany system ma dostęp do wszystkich plików hosta. Podane na występowanie błędu są następujące produkty: VMware Workstation, VMware Player, VMware ACE. Podatność nie występuje w VMware Server, ponieważ brak tam opcji współdzielenia katalogów między systemami. Bezpieczne są również wersje VMware Fusion oraz VMware dla systemów GNU/Linux. W tej chwili nie istnieje jeszcze poprawka eliminująca błąd dlatego producenci oprogramowania zalecają wyłączenie funkcji współdzielonych folderów.

196 POLAKÓW W DRUGIEJ RUNDZIE IMAGINE CUP 2008

Microsoft podsumował wyniki pierwszej rundy międzynarodowego konkursu technologicznego dla studentów Imagine Cup 2008. W tegorocznej edycji wystartowało 2090 uczestników z Polski, czyli o ponad 500 osób więcej niż w roku ubiegłym. Po podsumowaniu pierwszej rundy konkursu, Polacy mają swoją reprezentację w siedmiu z dziewięciu kategorii. W sumie do drugiej rundy rozgrywek konkursowych zakwalifikowało się 196 osób z Polski.

Cieszy nas rosnące zainteresowanie konkursem Microsoft Imagine Cup. Ubiegłoroczne zwycięstwo studentów z Polski w aż 3 kategoriach z pewnością pomogło w rozpropagowaniu tego przedsięwzięcia, a pozytywne wrażenia i doświadczenia, jakie przywieźli z sobą studenci przekonały i zainspirowały ich kolegów i koleżanki. Polacy są bardzo utalentowani i od kilku lat z dużymi sukcesami startują w Imagine Cup oraz innych konkursach informatycznych. Tu szansę dostają nie tylko osoby związane z informatyką, ale również innymi dziedzinami nowych technologii, stąd wielu studentów z kierunków innych niż informatyka. To kolejne duże osiągnięcie i jesteśmy bardzo zadowoleni, widząc jak studenci różnych kierunków, połączeni pasją do technologii, wzajemnie uczą się od siebie i współpracują – powiedział Karol Wituszyński z polskiego oddziału firmy Microsoft.

Temat przewodni tegorocznej edycji konkursu Imagine Cup brzmi: *Wyobraź sobie świat, w którym technologia pomaga chronić środowisko*. Zmagania są realizowane w dziewięciu kategoriach: Projektowanie Oprogramowania, Projektowanie Systemów Wbudowanych, Projektowanie Gier, Projekt Hoshimi – Bitwa Programistyczna, Technologie Informatyczne, Algorytmy, Fotografia, Film Krótkometrażowy i Projekt Interfejsu.

Microsoft Imagine Cup jest największym międzynarodowym konkursem technologicznym dla studentów. W ubiegłorocznej edycji konkursu, której międzynarodowy finał odbył się w Korei Płd., Polacy startowali w 7 z 9 kategorii konkursowych i zajęli pierwsze miejsca w trzech z nich, co zapewniło nam pierwszą lokatę w klasyfikacji generalnej konkursu.

DZIURA W LIGHTTPD

W serwerze www – *lighttpd* znaleziono poważną lukę, która pozwala na zdalne doprowadzenie do awarii oprogramowania. Luka polega na błędzie występującym podczas obliczeń przeprowadzanych w czasie dodawania deskryptorów plików do globalnej tablicy, co prowadzi do usterek zapisu i zakłócenia pracy serwera. Błąd występuję w wersji 1.4.18 i dotyczy także wcześniejszych edycji, ale tylko do wersji 1.4.8. Oficjalna poprawka nie została jeszcze przygotowana.

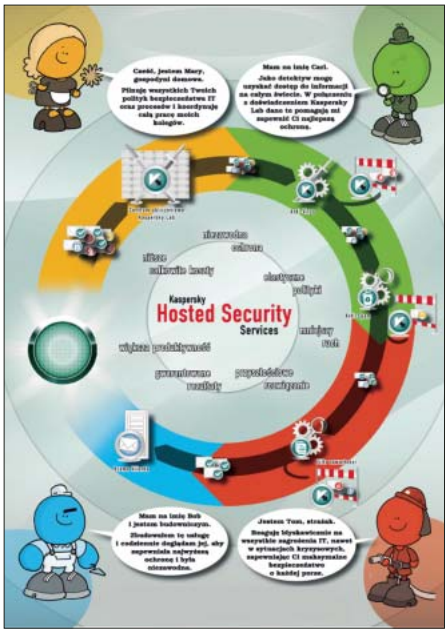
TREND MICRO PRZEJMUJE IDENTUM

Firma Trend Micro (globalny lider wśród producentów oprogramowania antywirusowego) poinformowała o przejęciu angielskiego producenta oprogramowania do szyfrowania poczty elektronicznej – Identnum. Trend Micro planuje zintegrować oprogramowanie Private Post (szlagierowy produkt Identnum) z istniejącą linią produktów oraz zmianę brand-u na Trend Micro (Bristol) Ltd.

KONFERENCJA KASPERSKY LAB

Ataki na firmy przeprowadzane są w większości za pośrednictwem Internetu, a nowe szkodliwe programy rozprzestrzeniają się w coraz szybszym tempie. Poważne epidemie wirusów wymagają interwencji administratorów, którym z powodu coraz większego obciążenia pracą często brakuje czasu oraz wiedzy niezbędnej do wykonania tego zadania. Pociąga to za sobą różne konsekwencje, z których przeciążenie sieci oraz przestój w pracy to najmniejsze zmartwienia firm. Dlatego Kaspersky stworzył Hosted Security Services, narzędzie zostało zaprezentowane na konferencji w lutym 2007 r. Usługi są dostępne od drugiego kwartału 2008 roku. Zanim wirusy i spam przeniknął do sieci korporacyjnej za pośrednictwem poczty elektronicznej, stron WWW czy komunikatorów internetowych, zostaną odfiltrowane przy pomocy sprawdzonych, wysoce skutecznych algorytmów.

Usługi Kaspersky Hosted Security Services to innowacyjne podejście do zwalczania zagrożeń internetowych, które są skutecznie filtrowane na długo przed dotarciem do bramy internetowej firmy. Dzięki temu niebezpieczna zawartość nigdy nie trafia do sieci korporacyjnej. Usługi Kaspersky Hosted Security Services zapewniają nie tylko szybszą reakcję na nowe zagrożenia, ale również obniżenie całkowitych wydatków firmy na bezpieczeństwo internetowe. Prosta i szybka integracja pozwala na błyskawiczne wdrożenie usługi.



ZAWARTOŚĆ CD

Na dołączonej do pisma płycie znajduje się dystrybucja hakin9.live (h9l) w wersji 4.0.3 on BackTrack2.0, 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.

PROGRAMY

Wersje programów:

- Advanced Archive Password Recovery – bez ograniczeń czasowych,
- Advanced RAR Password Recovery – bez ograniczeń czasowych,
- Advanced ZIP Password Recovery – bez ograniczeń czasowych,
- Comdom antispam for Linux – bez ograniczeń czasowych,
- Fast Report Server – bez ograniczeń czasowych,
- Panda Internet Security 2008 – wersja 3-miesięczna,
- UserGate – wersja 3 miesięczna.

FILMY INSTRUKTAŻOWE

Siódmy odcinek: *Includowanie* kodu PHP z innego serwera

Kolejny odcinek z serii filmów instruktażowych, przedstawiający najpopularniejsze metody ataków na strony internetowe.

BACKTRACK2.0 NA TWOIM PENDRIVIE

Utwórz partycję na pendrivie:

```
# fdisk /dev/sda
```

Uwaga: Jeśli posiadasz dyski SCSI lub SATA, sprawdź gdzie są umieszczone – `/dev/sda` może być Twoim dyskiem systemowym!

Wykasuj wszystkie istniejące partycje (wciśnij *d* oraz *Enter*, później wprowadź ilość partycji – od 1 do 4). Aby sprawdzić obecny stan partycji, wprowadź *p*. Później zacznij tworzyć nową partycję FAT32 – o wielkości około 800 MB. W tym celu wciśnij *n*, zatwierdzając klawiszem *Enter*. Zacznij od początku i ustal wielkość tworzonej partycji lub wciśnij jeszcze raz *Enter*, aby użyć całego urządzenia. Rodzaj partycji musi zostać zmieniony na FAT32 – wprowadź *t* i odpowiedz *b* na pojawiające się pytanie.

Musimy teraz sprawić, żeby nowa partycja była bootowalna. Wpisz *a*, a następnie wprowadź numer partycji – *1*. Teraz wpisz *w* w celu zapamiętania zmian.

PLIKI

Na początku utwórz na nowej partycji system plików:

```
# mkfs.vfat /dev/sda1
```

Teraz zdefiniuj punkt montowania tworzonego systemu plików:

```
# mount /dev/sda1 /mnt/usb
```

Skopiuj pliki hakin9 live do przygotowanej lokalizacji:

```
# cp -a /mnt/cdrom/* /mnt/usb/
```

Niektóre struktury plików powinny zostać usunięte:

```
# cd /mnt/usb/
# rm boot/vmlinuz
# rm boot/initrd.gz
```

W */mnt/usb* powinien znajdować się plik *syslinux.cfg*.

Po tej operacji wykonaj następujące polecenia:

```
# umount /dev/usb/
# syslinux /dev/sda1
```

W przypadku problemów wydaj polecenie:

```
# syslinux-nomtools /dev/sda1
```

Zrestartuj teraz maszynę i w BIOSie ustaw bootowanie z USB-HDD

Gotowe, utworzyłeś w pełni funkcjonalny system na swoim pendrivie. Pamiętaj, że bootowanie z USB jest obsługiwane jedynie przez nowe płyty główne.

Obecnie cała ta operacja jest możliwa tylko z pendrive'ami, które posiadają sektory o rozmiarze 512 bajtów.

JAK ZACZĄĆ

Aby zacząć pracę z *hakin9.live*, wystarczy uruchomić komputer z CD. Po uruchomieniu systemu możemy zalogować się jako użytkownik *hakin9* bez podawania hasła.

ADVANCED ARCHIVE PASSWORD RECOVERY (ARCHPR)

Program do odtwarzania zapomnianych haseł do archiwów ZIP (PKZIP, WinZIP), ARJ/WinARJ, RAR/WinRAR i ACE/WinACE. Program łączy w sobie wszystkie możliwości Advanced ZIP Password Recovery, Advanced ARJ Password Recovery, Advanced ACE Password Recovery i Advanced RAR Password Recovery. Oprócz tego, w ARCHPR możliwe jest gwarantowane odtwarzanie zawartości WinZIP archiwów, jeśli zawierają one więcej niż pięć plików. Podstawowe cechy ARCHPR wymieniono poniżej:

- gwarantowane odtwarzanie zawartości chronionych hasłem archiwów WinZIP niezależnie od trudności ustawionego hasła (operacja odtwarzania zajmuje około 1 godziny),
- program ma interfejs angielski,
- najszybsze na świecie przeszukiwanie haseł do archiwów ZIP, ARJ i RAR, optymalizowane pod procesory Pentium, Pentium II, Pentium III, Pentium IV i AMD Athlon, szybkość przeszukiwania haseł do archiwum ZIP wynosi około 15 milionów haseł na sekundę na procesorze P-III 1GHz,
- obsługiwanie archiwów PKZIP 4.0 (z podpisem cyfrowym i algorytmem kompresji deflate64),
- w przypadku archiwów ZIP i ARJ możliwy jest atak według znanej zawartości (ang. *known plaintext attack*). Jeśli znana jest zawartość chociaż jednego pliku w archiwum ZIP, odtworzenie hasła następuje w ciągu kilku godzin niezależnie od jego trudności i długości,
- szybkość pracy programu nie zależy od ilości plików w archiwum,
- obsługiwane są archiwa samorozpakowujące się (SFX),
- program posiada mnóstwo ustawień – mogą Państwo zadawać dowolną długość hasła, zestaw symboli i wiele innych opcji,
- możliwość zadania własnego zestawu symboli do przeszukiwania haseł (obsługiwane jest również rosyjskie przeszukiwanie symboli),
- możliwe jest przeszukiwanie haseł według słownika,
- dostępne jest przeszukiwanie haseł według maski,
- maksymalna długość hasła nie jest ograniczona,
- pracę programu można przerwać w dowolnej chwili i później kontynuować wykonanie,
- program może pracować w trybie tła, nie zajmując czasu procesora, kiedy ten jest potrzebny do wykonania innych zadań.

Rysunek 1. *Archpr 4.0 Professional Edition*

ADVANCED RAR PASSWORD RECOVERY

Program do odtwarzania utraconych haseł do archiwów utworzonych przez archiwatory RAR i WinRAR. W tych archiwatorach zastosowany został bardzo odporny algorytm szyfrowania. Z tego powodu hasła do archiwum nie można wyliczyć z jakichkolwiek danych. Jednak hasło można dobrać przez bezpośrednie przebieranie lub za pomocą ataku według słownika. W ARPR wbudowany został mocny moduł przebierania, zoptymalizowany pod procesory Pentium III i Pentium IV.

Podstawowe cechy możliwości ARPR:

- program posiada angielski i rosyjski interfejs,
- obsługiwanie archiwów zawierających jeden plik,
- obsługiwanie wszelkich metod kompresji dostępnych w RAR,
- obsługiwane są archiwa samorozpakowujące się (SFX),
- program posiada mnóstwo ustawień – mogą Państwo zadać dowolną ilość hasła, wybór symboli i wiele innych opcji,
- możliwość zadawania własnego zestawienia symboli do przebierania haseł (obsługiwane jest również rosyjskie zestawienie symboli),
- możliwość przebierania haseł według słownika,
- możliwość przebierania haseł według maski,
- pracę programu można przerwać w dowolnej chwili i później kontynuować wykonanie,
- program może działać w trybie tła, nie zajmując czasu procesora, kiedy ten potrzebny jest do wykonania innych zadań.

ADVANCED ZIP PASSWORD RECOVERY (AZPR)

Program do odtwarzania zapomnianych haseł do archiwów utworzonych przez archiwatory ZIP i WinZIP. Program umożliwia dostosowanie haseł do archiwów przez bezpośrednie przesz ukiwanie lub atak według słownika. Do AZPR został wbudowany najszybszy na świecie moduł przeszukiwania, zoptymalizowany pod procesory Pentium III i Pentium IV. Podstawowe możliwości AZPR:

- program posiada angielski interfejs,
- obsługa archiwów zawierających dowolną ilość plików,
- obsługa wszystkich sposobów kompresji dostępnych w ZIP,
- obsługiwane są archiwa samorozpakowujące się (SFX),
- program posiada mnóstwo ustawień – mogą Państwo zadawać dowolną długość hasła, zestaw symboli i wiele innych opcji,
- możliwość zadania własnego zestawu symboli do przeszukiwania haseł (obsługiwane jest również rosyjskie przeszukiwanie symboli),
- możliwe jest przeszukiwanie haseł według słownika,
- możliwe jest przeszukiwanie haseł według maski,
- pracę programu można przerwać w dowolnej chwili i później kontynuować wykonanie,
- program może pracować w trybie tła, nie zajmując czasu procesora, kiedy ten jest potrzebny do wykonania innych zadań.

COMDOM ANTISPAM FOR LINUX

Comdom AntiSpam zapewnia, dzięki poszczególnym modułom składającym się na aplikację, skuteczną ochronę przed spamem. Podstawowe moduły to:

- brama pocztowa,
- kontrola transmisji,
- wspomaganie autoryzacji,
- filtr Bayesa,
- czarne, białe, szare listy,
- konfiguracja zależna od domeny/email,
- DKIM,
- SPF.

Comdom Antispam jest dostępny dla każdego bez żadnej rejestracji, ani innych procedur! Wystarczy jedynie, że zaakceptujesz umowę licencyjną i możesz w pełni korzystać z tego rozwiązania. Wersja ta jest ograniczona do 100 adresów email (nie do 100 użytkowników!). Adresy muszą być jasno wpisane w pliku konfiguracyjnym po instalacji programu. Tylko wtedy wiadomości będą akceptowane, w przeciwnym wypadku e-maile zostaną odrzucone. Wersja ta nie posiada żadnej bazy spamu, jednak umożliwia zbudowanie własnej. Aby pobrać już przygotowane bazy, należy się zarejestrować.
WWW: www.ti.com.pl, www.comdomantispam.pl, www.comdomsoft.com.
E-mail: comdom@ti.com.pl

Rysunek 2. *Comdoms*

FASTREPORT SERVER

FastReport Server to prosty w ustawieniach, stabilny i posiadający duże możliwości serwer raportów. Przeznaczony jest do tworzenia załączników na podstawie zawartości sieci Web, może być zintegrowany z różnymi projektami działającymi w architekturze klient-serwer.
Możliwości serwera:

- całkowicie autonomiczny serwer HTTP,
- wysoka wydajność pracy,
- minimalne wymagania w odniesieniu do zasobów systemowych,
- prostota administrowania,
- dostęp do baz danych zrealizowany jest za pomocą mechanizmu Microsoft AD,
- dostarczany wraz z projektem sprawozdań FastReport i narzędziem konfigurowania serwera.
- wykorzystanie w charakterze klienta dowolnej przeglądarki internetowej, wielostronicowy podgląd za pomocą nawigatora stron,
- praca w Windows,
- obsługiwane wyjściowe formaty: HTML, PDF, RTF, XML, Jpeg, Bmp, Gif, Text, CSV, FastReport,
- wbudowane środki identyfikacji użytkowników w celu ograniczenia dostępu do sprawozdań,
- ograniczenie dostępu do serwera dla zdefiniowanych adresów IP,
- zastosowanie dziennika kontroli dostępu oraz błędów,
- analiza dzienników w celu otrzymania statystycznej informacji dotyczącej wykorzystania serwera,
- kontrola obciążenia serwera,
- wsparcie dla formularzy webowych do prowadzenia dialogu z użytkownikiem,
- wysyłanie sprawozdań w wyspecyfikowanym formacie pocztą elektroniczną na żądanie lub we wcześniej ustalonym terminie, wsparcie dla realizacji zaległych zapytań z niskim priorytetem przy dużym obciążeniu serwera,
- ochrona integralności przekazywanej informacji przy pomocy metod podpisu cyfrowego (Message Integrity Checksum),
- przechowywanie i dostarczanie na żądanie najczęściej wymaganych informacji,
- wsparcie technologii SSI (Server Side Include) dla statycznych stron HTML,
- współpraca z innymi serwerami HTTP przy pomocy tuneli CGI,
- integracja z innymi aplikacjami w architekturze klient-serwer przy wykorzystaniu bibliotek z pakietu FastReport Studio.

Wymagania:
Platforma: Microsoft Windows XP/2003,
RAM: 128MB – minimum, 512MB – optymalnie,
HDD: 512MB – minimum,
CPU: 500MHz – minimum, 2GHz – optymalnie.

PANDA INTERNET SECURITY 2008

Spoglądając na ilość powstających zagrożeń, można już teraz śmiało stwierdzić, iż rok 2008 będzie pod tym względem rekordowy. Nawet 2000 i więcej próbek przechwytywanych każdego dnia przez laboratoria antywirusowe może przyprawić o ból głowy. Dobre wieści są takie, iż powstają rozwiązania, które zadbają w pełni o nasze bezpieczeństwo.

Doskonałym przykładem jest Panda Internet Security 2008 – niezwykle rozbudowany pakiet, który nie tylko chroni przed wszelkimi zagrożeniami, ale też daje możliwość tworzenia kopii zapasowych czy filtrowania treści w Internecie. Warto jednak rozpocząć od tego, co oprogramowanie antywirusowe posiadać musi bezapelacyjnie. Jest to ochrona przed znanymi i nieznanymi zagrożeniami. Panda Internet Security zabezpiecza zarówno przed wirusami, robakami, końmi trojańskimi, ale także przed programami szpiegującymi czy rootkitami. Aby użytkownik nie musiał martwić się o ciągłe aktualizacje (które notabene przeprowadzane są automatycznie nawet kilka razy dziennie), program wyposażony jest w drugą linię ochrony w postaci technologii TruPrevent™. Moduł ten, uznany przez niezależną organizację AV-Test za najskuteczniejszy sposób zapobiegania infekcjom jeszcze nie wykrytym przez laboratoria, zapewni nam maksimum bezpieczeństwa w każdej chwili.

Abyśmy mogli dodatkowo przeprowadzić inspekcję systemu pod kątem złośliwych kodów, których tradycyjna ochrona nie jest w stanie wykryć, Panda Security udostępniła użytkownikom swoich programów możliwość przeprowadzenia skanowania z poziomu specjalnie przygotowanej strony internetowej. To innowacyjne rozwiązanie, zawarte w pakiecie Panda Internet Security 2008, pozwala skorzystać z zewnętrznej bazy zagrożeń zawierającej niemal 3 miliony sygnatur w celu dokonania tzw. *dogłębnego*

skanowania systemu on-line. Zewnętrzny skaner zarówno wyszuka, jak i usunie wszelkie *wykwintne* zagrożenia, pozostawiając nasz system wolny od najbardziej wyszukanych, złośliwych kodów.

Program Panda Internet Security wyposażony jest także w firewall, co w tego typu aplikacjach jest już standardem. Warto jednak odnotować fakt, iż w przypadku programu Panda zapora jest dwukierunkowa. Dzięki temu chronimy się przed włamywaczami, ale też zapobiegamy sytuacji, w której to nasz komputer miałby się stać *nieświadomym* atakującym, pozostającym na usługach hakerów. Wspomniany firewall jest w pełni konfigurowalny.

Ochrona tożsamości oraz AntiPhishing to kolejne moduły odpowiedzialne za nasze bezpieczeństwo w Internecie. Można śmiało stwierdzić, iż ochrona ta jest bardzo dobrze przemyślana. Odebrana wiadomość typu phishing zostanie z dużym prawdopodobieństwem okraszona przez program Panda komentarzem, iż może to być próba pozyskania naszych poufnych danych. W sytuacji, gdy komentarz ten zostanie przez nas zlekceważony, program i tak nie pozwoli przesłać wszelkich wrażliwych danych, które wcześniej powierzyliśmy mu w opiekę. Panda Internet Security stać może się zatem obrońcą takich informacji jak numer karty kredytowej, hasła do konta bankowego czy też innego, ważnego dla nas ciągu znaków.

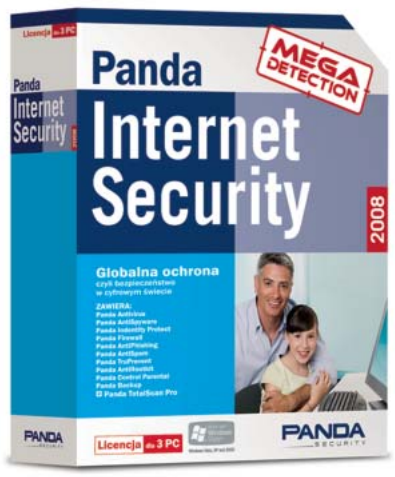
Bezpieczeństwo w Internecie to także pewność, iż nasze pociechy nie mają dostępu do określonych stron WWW. Tutaj Panda także idzie krok dalej, nie uznając kompromisów. Skoro bowiem możemy chronić nasze pociechy, czemu nie zadbać o swój portfel i np. zabronić naszej drugiej połowie dostępu do serwisów aukcyjnych? Panda pozwala ustalić zasady filtrowania stron w oparciu o niemal 60 kategorii takich jak erotyka, religia, polityka, broń itd. Abyśmy dodatkowo nie byli atakowani niezliczoną ilością wiadomości zachęcających nas do zakupów w niepewnych źródłach, program wyposażono w zaawansowaną ochronę przed spamem.

Warto przyjrzeć się dwóm dodatkowym modułom, które wyróżniają Panda Internet Security 2008. Jest to Backup oraz Tuneup. Pierwszy moduł skupia się na ochronie naszych danych przed fizyczną utratą. Możemy zatem przeprowadzać archiwizację najważniejszych dokumentów. Archiwizacja ta odbywać może się automatycznie, na wybrany przez nas nośnik. Firma Panda Security udostępnia także bezpłatnie 1GB wolnego miejsca na zaufanym serwerze, gdzie zamieścić możemy dane bez obawy o ich utratę. Moduł Tuneup skupia się z kolei na optymalizacji działania systemu poprzez takie operacje, jak usuwanie zbędnych plików czy też defragmentacja wskazanych dysków.

Produkt Panda Internet Security chroni zatem globalnie i korzystając z jego zaawansowanych funkcji możemy poczuć się w pełni bezpiecznie. Wymagania systemowe programu są następujące:

Procesor: Pentium 300 MHz lub szybszy, RAM: 128 MB (rekomendowane 256 MB), Dysk twardy: 270 MB wolnego miejsca,

System operacyjny: Windows Vista 32 oraz 64 bit, Windows XP 32 oraz 64 bit, Windows 2000, Przeglądarka Internet Explorer 6.0.

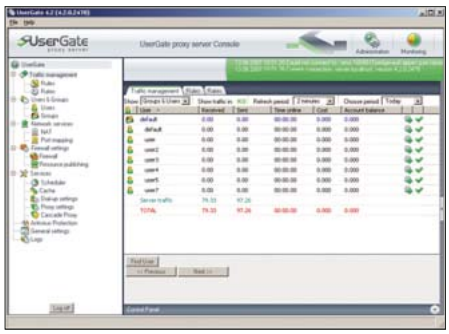


Rysunek 3. *Internet Security*

USERGATE

UserGate to kompletne rozwiązanie pozwalające na współdzielenie dostępu do Internetu. Pozwala ono na podłączenie sieci lokalnej do Internetu przy użyciu tylko jednego, zewnętrznego adresu IP. UserGate zapewnia scentralizowane zarządzanie połączeniami z Internetem, magazynuje źródła w cache, oblicza obciążenie łącza ruchem, zawiera wbudowany system naliczania kosztów oraz moduł statystyczny. Główne cechy programu:

- podłączenie do sieci lokalnej bez użycia sprzętowego routera,
- ekonomizacja obciążenia łącza,
- filtrowanie zawartości,
- zarządzanie dostępem użytkowników do Internetu,
- zabezpieczenie typu firewall,
- antywirus i antyspyware.



Rysunek 4. *Usergate42*

PODZIĘKOWANIA

Serdeczne podziękowania dla firm Softkey Poland Sp. z o. o., TTS Company oraz Panda za udostępnienie programów na płytę.

Softkey

HAKIN9 LIVE

Żeby uruchomić swój komputer z płyty *hakin9.live*, ustaw swój BIOS na bootowanie z napędu CD-ROM. Po dokonanych zmianach uruchom ponownie komputer. Uruchomi się dystrybucja *hakin.live*, na której możesz przećwiczyć techniki prezentowane w tutorialach. Upewnij się, ze sprawdziłeś desktopowe foldery – zawierają wiele dodatkowych materiałów. Zawartość CD można również przejrzeć w systemie Windows.

Panda Antivirus + Firewall 2008

Producent
Panda Security
System
Windows Vista, Windows XP, Windows 2000
Typ
Antywirus
Strona producenta
www.pandasoftware.com.pl
Recenzent
Paweł Lisowski

OCENA
★★★★★

Przez cały okres użytkowania opisywanego oprogramowania komputer nie został w żaden sposób zainfekowany czymkolwiek, nie został także dokonany żaden udany atak. Proces aktualizacji odbywa się płynnie i nie przysparza żadnych problemów. Warto wspomnieć o tym, że aktualizacje są dokonywane codziennie. Zwiększa to znacznie bezpieczeństwo naszego komputera. Panda skanuje również w czasie rzeczywistym pocztę. Ponadto oprogramowanie posiada system zwany *Smart Clean*, który pozwala naprawić uszkodzenia wywołane przez niepożądany złośliwy kod. Pakiet składa się z wielu modułów – obecnie jest to standard, od którego Panda nie odchodzi. Pierwszym z modułów jest oczywiście antywirus, wykrywający skutecznie wirusy, trojany i robaki.

Panda dysponuje również technologią proaktywną, która pozwala zabezpieczyć komputer przed jeszcze nie rozpoznanymi zagrożeniami. Moduł *antyspyware* chroni przed programami szpiegującymi, a także przed denerwującymi *pop–upami*, reklamami itd. Komponent *antyphishing* pomaga nam uchronić się przed fałszywymi stronami WWW. Unikamy dzięki temu kradzieży numerów kart kredytowych czy kont bankowych, haseł i innych cennych dla nas informacji. Dlatego system ten jest bardzo istotnym elementem pakietu. Wreszcie moduł *antyrootkit* wykrywa programy próbujące ukryć inne niebezpieczne aplikacje.

Panda posiada również system całkowitego, dogłębnego skanowania, który nazwano *Total Scan Pro*. Wykrywa on różnego rodzaju złośliwe oprogramowanie. Panda została również wyposażona w system filtracji stron WWW, który blokuje dostęp do podejrzanych i niebezpiecznych stron, chroniąc nas tym samym przed pobraniem niechcianego oprogramowania szpiegującego. Można również podzielić się własnymi spostrzeżeniami, wysyłając informacje do twórców programu dzięki skorzystaniu z formularza *Uwagi użytkowników*. Możliwości konfiguracji są szerokie, a interfejs użytkownika czytelny, przejrzysty i ładny. Wydaje mi się, że obecna Panda jest o wiele lepsza od poprzednich wersji, które nie cieszyły się zbyt dobrą opinią. Ciekawą opcją jest możliwość identyfikacji wykrytych luk w zabezpieczeniach – tyle tylko, że podawane przez program symbole raczej nic nie mówią. Na szczęście problem można rozwiązać aktualizując oprogramowanie.

Co niezwykle istotne, Panda nie obciąża mocno systemu. Wymagania są naprawdę niewysokie. Według producenta wystarcza procesor 150 MHz, 64 MB pamięci RAM i ok. 170 MB wolnego miejsca na dysku. Z moich testów wynika, że zapewnienia producenta są prawdą – Panda naprawdę nie obciąża systemu, nie wpływając w zauważalny sposób na wydajność systemu. Jest to ogromnym plusem na tle konkurencji.

Krótko mówiąc, Panda to bardzo dobry antywirus, nie sprawiający problemów, łatwo konfigurowalny, z przyjemnym interfejsem. Może trochę liczb: liczba zablokowanych w czasie testów ataków to 118, połączenia odrzucone: 15, wykryte luki: 60, a całkowita liczba zablokowanych zagrożeń to 31602. Widać zatem, że Panda działa wprost doskonale. Potwierdza to brak na komputerze jakichkolwiek trojanów, wirusów, robaków, nie ma również irytujących reklam *pop–up*. Można dokładnie przeanalizować, jakie programy chciały zainfekować nasz komputer – wystarczy przejrzeć raporty skanowania. Uzyskujemy tam również informację o adresach IP, z których próbowano się do nas podłączyć.

Jeśli chodzi o pomoc programu, to można ją ocenić na 5. Wszystko jest opisane w sposób łatwy do zrozumienia. Zagadnienia są ułożone w sposób sprzyjający płynnej nawigacji i dobrze skategoryzowane. Dla osób niezaawansowanych korzystanie z pomocy na pewno okaże się bardzo przydatne – zwłaszcza, że w zasadzie nie ma możliwości niezrozumienia określonego zagadnienia.

Na dobrą sprawę – minusów podczas testów nie zauważyłem. Ważne jest przede wszystkim to, aby antywirus w należyty sposób chronił komputer.

Norton Ghost 12.0

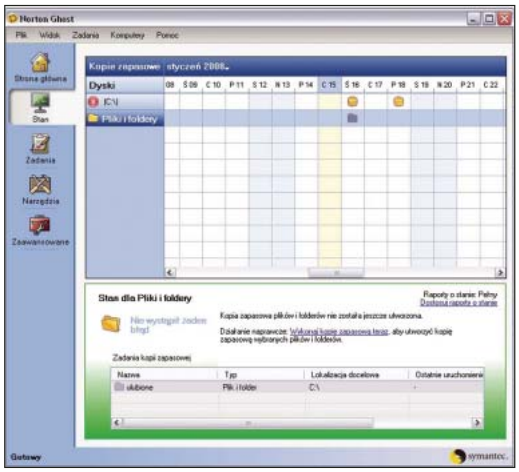
Do dzisiaj używałem Norton Ghost 2003. Zawsze, gdy musiałem sklonować dysk, wrzucałem bootowalną płytę do napędu i w DOS–owych okienkach wybierałem kolejne opcje, aby na koniec dostać dwa identyczne dyski i zawsze się zastanawiałem, czy dostanę dwa puste dyski, czy dwa pełne.

Trzeba przyznać, że 5 lat to duzo czasu. Widać to idealnie, gdy po NG2003 włącza się NG 12.0. Na samym początku postanowiłem sprawdzić swoje ulubione opcje, czyli to, co oferuje program przed odpaleniem Windows. Przede wszystkim mamy do dyspozycji opcję przywracania systemu. Funkcja nieoceniona, zwłaszcza dla ludzi, którzy lubią poszperać w rejestrach i ukrytych plikach. Tutaj jest jedno ale. Wcześniej taki punkt przywracania (lub obraz dysku) musi być zapisany bądź na dysku twardym, bądź na płycie, czy wreszcie na jakimkolwiek innym nośniku podłączonym przez USB lub FireWire.

Oprócz przywracania danych wersja bootowalna Norton Ghost posiada wiele innych funkcji, takich jak: skaner antywirusowy (z możliwością aktualizacji sygnatur wirusów z pliku), sprawdzanie błędów na dyskach twardych, eksploracja komputera (obejmująca praktycznie wszystkie nośniki doń podłączone), dostęp do linii komend, a także proste, acz przydatne operacje na partycjach (np. zmiana aktywnej partycji).

To wszystko czyni Noton Ghost 12.0 narzędziem przydatnym już na etapie *boot–CD*.

Zaraz po zainstalowaniu programu, NG proponuje stworzenie obrazu całego dysku (czyli punktu przywracania) bądź tylko poszczególnych katalogów. Od tego momentu Norton Ghost może przejąć kontrolę nad naszymi *backupami*. Jesteśmy w stanie – przy pomocy prostego kreatora – zdecydować, czy *backup* Moich Dokumentów ma się robić we środę o 22, czy może lepiej w sobotę o 10:00. I w ten sposób można ustalić terminy tworzenia kopii zapasowej każdego folderu na naszych dyskach. Każde zadanie jest reprezentowane graficznie na czymś w rodzaju kalendarza, co w dużym stopniu ułatwia zarządzanie *backupami*. Każdy proces *backupu* można w

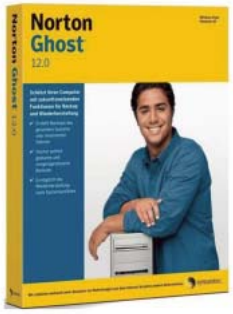


dowolnej chwili zarówno włączyć, jak i wyłączyć, czy też skontrolować wydajność systemu podczas aktualnie realizowanego zadania bądź obejrzeć dotyczący go raport.

W wersji instalacyjnej pojawiła się opcja klonowania dysku. Trochę szkoda, że pominięto ją, gdy *boot'ujemy* NG z płyty. Tutaj jakichś oszałamiających opcji nie ma. Ot – kopia *bit–to–bit* z dysku na dysk. Może mało odkrywcze, ale – jak wspomniałem na początku – niesamowicie przydatne. I właściwie w tym zakresie ciężko dodać coś nowego.

Z ciekawostek, warto zaznaczyć obecność funkcji konwersji na dysk wirtualny. Polega ona na przekonwertowaniu utworzonego przez siebie punktu przywracania na format dysku wirtualnego (*VMDK* lub *VHD*).

Powyżej wymieniona funkcjonalność, w połączeniu z możliwościami, których nie wymieniłem w tym krótkim tekście, sprawiły, że odstawiam Norton Ghost 2003 i zaczynam korzystać z NG 12.0. Przede wszystkim świetny tryb pracy z bootowalnej płyty, jak i różnorodność opcji przy ustawianiu *backupów*, pozwalają mi spokojnie bawić się systemem – bez obawy, że po którymś z kolei restarcie okaże się, iż straciłem dostęp do danych na swoich dyskach twardych. Oczywiście wszystko to dostaniemy dopiero, gdy skonfigurujemy sobie odpowiednio Norton Ghosta, ale z oferowanymi przez niego kreatorami – będzie to dziecinnie proste.



Producent
Symantec
System
Windows XP i Windows Vista
Typ
Oprogramowanie do ochrony danych
Strona dystrybutora
www.softpoint.com.pl
Recenzent
Bartek Zalewski

OCENA

★★★★★

SŁAWOMIR ORŁOWSKI

Automatyczna generacja ciągów

Stopień trudności

Niestandardowe rozwiązania dla standardowych problemów potrafią znacznie uprościć i przyspieszyć pracę. W tym artykule przedstawiam ciekawą metodę rozwiązującą problem generacji wszystkich możliwych ciągów znaków z danego zbioru.

Z ARTYKUŁU DOWIESZ SIĘ

podstawy języka Java.

CO POWINIENEŚ WIEDZIEĆ

rekurencyjne wywołanie funkcji,

jak używać klasy BigInteger do reprezentacji dużych liczb,

jak generować dowolne permutacje dla pewnego alfabetu.

Istnieje wiele programów, które pomagają złamać hasło lub sprawdzają, czy hasła użytkowników zarejestrowanych w jakimś systemie są wystarczająco skomplikowane. Sami użytkownicy, aby zapamiętać hasło, zwykle używają imienia swojego partnera, zwierzęcia, przezwiska szkolnego itd. Dzięki temu znacznie ułatwiają potencjalnemu włamywaczowi dostęp do swoich danych. Co jednak zrobić, jeśli hasło jest na tyle mocne, że standardowe słowniki nie wystarczą? Możemy użyć naszego osobistego uroku i oczarować ofiarę – tak, aby zdradziła nam hasło. Jednym słowem zastosować bardzo obecnie modne socjotechniki. Gdy to nie zadziała, pozostaje nam tylko tzw. metoda brute force, czyli siłowa. Nie chodzi tu oczywiście o próbę zastraszenia, tylko o automatyczne generowanie wszystkich możliwych haseł. Musimy się więc zmierzyć z problemem generacji możliwych ciągów dla danego alfabetu (zbioru znaków). Dla przypomnienia – liczba wszystkich kombinacji dla zbioru n-elementowego wynosi `n!`. Oznacza to, że dla `n=5` liczba ta wyniesie `1*2*3*4*5 = 120`. Jak szybki jest wzrost tej funkcji, można się przekonać licząc np. `10!`, `12!` czy `64!`. W przypadku próby znalezienia odpowiedniego hasła możemy nałożyć sobie więzy, które znacznie skrócą czas obliczeń. Więzy te to długość hasła. Nasze zadanie skraca się zatem do problemu `NK`, gdzie `N` liczbą znaków w alfabecie, natomiast przez `K` oznaczamy długość hasła. Dramatycznie zmniejsza to przestrzeń naszych poszukiwań.

Rozważmy alfabet o długości 35 znaków, który zwiera wszystkie litery i cyfry. Liczba możliwych kombinacji wyniesie:

```
35! = 1,033314796638614492966651337523e+40
```

Natomiast jeżeli szukamy słowa, powiedzmy, o długości 8 znaków, to:

```
358 = 2251875390625
```

Mogę śmiało zaryzykować stwierdzenie, że hasła o długości 1 lub 2 zdarzają się dosyć rzadko. Realnie używane hasła zwykle też nie są zbyt długie.

Tradycyjne podejście programistyczne nakazuje użycie rekurencyjnego wywołania funkcji. Jest to jakiś sposób. Należy jednak pamiętać, że przy rekurencji nie jesteśmy w stanie w prosty sposób przewidzieć zużycia pamięci. Często zdarza się również, że programista, który napisał funkcję rekurencyjną, sam nie jest pewny, czy działa ona poprawnie. Listing 1. przedstawia przykładową metodę w języku Java, która rekurencyjnie generuje wszystkie możliwe ciągi dla danego alfabetu. Należy jeszcze zadeklarować pola `sb` oraz `alphabet`:

```
private static StringBuilder sb = new
      StringBuilder(len);
private static String alphabet = "0123456789
      abcdefghijklmnopqrstuvwxyz";
```

Zmienna `len` będzie określała długość generowanego ciągu.

W tym artykule chciałbym zaproponować Czytelnikowi nieco inne podejście do tego problemu. Sama idea tego rozwiązania jest prosta. Każdy z ciągów utworzony z liter danego alfabetu może być reprezentowany przez pewną niepowtarzalną liczbę. Liczba ta jednoznacznie reprezentuje daną kombinację. Dzięki temu w kodzie może być użyty zwykły mechanizm indeksowania, co znacznie upraszcza i przyspiesza całą sprawę. Przyjrzyjmy się bliżej, na jakiej zasadzie to działa. W systemie dziesiętnym dowolną liczbę możemy zapisać jako sumę cyfr z tego systemu pomnożonych przez kolejne potęgi dziesiątki. Liczba 1978 będzie zapisana jako:

```
1978 = 1*103 + 9*102 + 7*101 + 8*100
```

Jest to wiedza serwowana nam już chyba w szkole podstawowej. W podobny sposób możemy pomyśleć o ciągach z danego alfabetu. Niech nasz alfabet składa się z sześciu znaków:

```
? = {a,b,c,d,e,f}
```

Litera a ma pozycję numer 1, `b` numer 2 itd. My będziemy szukać wszystkich ciągów o długości 4. Wiemy już, że takich ciągów będzie 64, czyli 1296. Spróbujmy teraz przedstawić ciąg cafe za pomocą naszej notacji:

```
cafe = 3*63 + 1*62 + 6*61 + 5*60 = 725
```

Znak `c` jest trzecim znakiem w alfabecie, długość alfabetu wynosi 6. Mnożymy więc 3 i 6 podniesione do trzeciej potęgi, ponieważ licząc od prawej i indeksując począwszy od zera, `c` zajmuje trzecią pozycję w ciągu cafe. Możemy również uznać, że litera a ma pozycję 0, wówczas:

```
cafe = 2*63 + 0*62 + 5*61 + 4*60 = 466
```

Należy tylko pamiętać, że odnosi się to jedynie do ciągów o tej samej długości. W przeciwnym wypadku mielibyśmy kłopot z sekwencjami typu *a*, *aa*, *aaa* itd., ponieważ za każdym razem ich reprezentacja wynosiłaby 0. Nie byłoby to więc wzajemnie jednoznaczne. Proszę zwrócić uwagę na fakt, że znając jakąś permutację możemy powiedzieć, jaka będzie następna. Co więcej, możemy wskazać k-tą permutację dla danych warunków początkowych (alfabet, długość szukanego ciągu). Jest to niewątpliwą zaletą tego rozwiązania w porównaniu do rekurencyjnego wywoływania funkcji. Jeśli z jakiś powodów nagle przerwiemy działanie metody rekurencyjnej z Listingu 1. to aby wygenerować wszystkie ciągi, jesteśmy zmuszeni wywoływać funkcję jeszcze raz

Listing 1. *Rekurencyjne generowanie ciągów o danej długości*

```
private static void Brute(int len) {
      if (len == 0) {
         System.out.println(sb.toString());
      }
         else {
         for (int i = 0; i < length; i++) {
            sb.setCharAt(len-1, alphabet.charAt(i));
            Brute(len-1);
         }
      }
   }
```

Listing 2. *Metoda main*

```
public static void main(String[] args) {
      s = new StringBuilder();
      bia = new BigInteger[2];
      alphabet = "0123456789abcdefghijklmnopqrstuvwxyz";
      length = alphabet.length();
      int min = 2;
      int max = 3;
      String message = "passbrute.exe [min] [max]\nwhere min is a minimum length of
                       the password and max is a maximum length of the password. For
                       example: passbrute.exe 4 5";
      /*
      if(args.length !=2) {
         System.out.println(message);
         return;
      }
      if (!Character.isDigit((char)args[0].charAt(0)) || !Character.isDigit((char)args
                       [1].charAt(0))) {
         System.out.println(message);
         return;
      }
        **/

      for (int i = min; i <= max; i++) {
         BigInteger all = BigInteger.valueOf(length);
         all = all.pow(i);
         BigInteger number = BigInteger.ZERO;
         while (number.compareTo(all) < 0) {
             System.out.println(Count(number, i));
             number = number.add(BigInteger.ONE);
         }
      }
   }
```

Listing 3. *Metoda Count*

```
   private static String Count(BigInteger base, int stlen) {
      s.setLength(0);
      for (int i = stlen - 1; i >= 0; i--) {
         bia = base.divideAndRemainder(BigInteger.valueOf(length));
         int step = base.remainder(BigInteger.valueOf(length)).intValue();
         s.append(alphabet.charAt(bia[1].intValue()));
         base = bia[0];
      }
      return s.toString();
   }
```

od nowa. W przypadku proponowanej metody możemy wznowić poszukiwania po przerwaniu generacji (np. na liczbie 466), ponieważ znamy liczbę reprezentującą następny ciąg. Jest to liczba o jeden większa od tej, na której przerwaliśmy poszukiwania (467).

Mój tekst jest inspirowany artykułem pt. *Using Permutations in .NET for Improved Systems Security*, który napisał James McCaffrey dla MSDN. Implementacja tej metody jest również przedstawiona w artykule pochodzącym z witryny *Code Project*. Linki do nich znajdują się w ramce *W Sieci*. Zachęcam do przestudiowania tych tekstów.

Przejdźmy teraz do napisania programu. Będzie to aplikacja konsolowa. Językiem programowania będzie Java. Nasz program będzie miał możliwość określenia długości generowanych ciągów poprzez podanie wartości minimalnej i maksymalnej. Rozpoczniemy od definicji statycznych pól naszej klasy:

```
private static StringBuilder s;
private static String alphabet;
private static long length;
private static BigInteger[] bia;
```

Pole `alphabet` przechowywać będzie alfabet, jaki użyjemy w trakcie generowania ciągów. Jego długość będzie przechowywana w zmiennej `length`. Referencja klasy `StringBulider` posłuży nam do konstrukcji ciągu z danej reprezentacji liczbowej. Jest to znacznie lepsze rozwiązanie niż użycie klasy `String`, ponieważ zużywa znacznie mniej pamięci i jest prawdopodobnie szybsze. Innym rozwiązaniem może być użycie tablicy znaków. Jednak moim zdaniem wygodniejsze jest użycie właśnie klasy `StringBuilder`. Liczby reprezentujące ciągi mogą być znacznych rozmiarów. Dlatego zdecydowałem się wykorzystać klasę `BigInteger`. Jest to bardzo wygodna klasa do reprezentacji dużych liczb, dla których zakresy double lub long są za małe. Aby mieć dostęp do tej klasy, musimy zaimportować odpowiedni pakiet:

```
import java.math.BigInteger;
```

Na pierwszy rzut oka dziwić może fakt deklaracji tablicy `bia`. Do czego może się przydać? Otóż klasa `BigInteger` posiada metodę `divideAndRemainder`, która zwraca wynik zwykłego dzielenia i dzielenia modulo jako tablicę dwuelementową. Metoda ta przyda nam się nieco dalej do zamiany liczby na odpowiadający jej ciąg. Inicjalizację wszystkich pól przeprowadzimy w metodzie main, której kod przedstawia Listing 2. Jest to zabieg raczej estetyczny, ponieważ moglibyśmy je inicjalizować również przy definiowaniu.

Na początek powołujemy do życia pola `s` i `bia`. W kolejnym kroku określamy alfabet oraz jego długość. Dalej definiujemy zmienne min i max, które będą przechowywały zakres długości generowanych ciągów. Na początku niech będzie to 2 i 3. Sam zakres będzie mógł wprowadzać użytkownik poprzez wywołanie programu z parametrami `min` i `max`. Do tego służą właśnie dwie kolejne instrukcje warunkowe, które sprawdzają poprawność argumentów. Na potrzeby testów można je zakomentować, tak, jak jest to zrobione na Listingu 2. Najważniejszy fragment kodu to dwie zagnieżdżone pętle. Zewnętrzna pętla `for` (indeksowana przez `i`) wskazywać będzie aktualną długość szukanego ciągu. Druga pętla będzie generowała liczby jednoznacznie reprezentujące ciągi. Ich sposób reprezentacji przedstawiony był we wstępie. Zakres pętli zawiera się pomiędzy zerem a maksymalną liczbą ciągów dla danej długości alfabetu i danej długości ciągu, czyli `length^i`. Użyłem tutaj typu `BigInteger`, ponieważ te liczby mogą być znacznych rozmiarów. Z racji tego, że nie możemy przeciążać operatorów, zmuszeni jesteśmy do użycia metod klasy `BigInteger`, które realizują odpowiednie działania. Metoda `compareTo` porównuje ze sobą dwie liczby `BigInteger` (`this` i argument wywołania metody). Zwraca -1, jeśli liczba `this` jest mniejsza od argumentu, 0 jeśli są równe `i` 1 – jeśli `this` jest większy od argumentu. Ze względu na to wygodnie jest użyć pętli while, której zmienna iteracyjna będzie zwiększana za pomocą metody `add`. W tej pętli wywoływana jest metoda `Count`, która na podstawie długości ciągu `i` liczby będzie zwracała reprezentujący ją ciąg. Jak widać, ciągi te wypisywane są na konsolę. Należy pamiętać, że w przypadku długich obliczeń potrafi to znacznie spowolnić proces generacji ciągów. Ostatnim krokiem w pisaniu tego programu będzie zdefiniowanie wspomnianej wcześniej metody Count. Jej kod przedstawia Listing 3.

Argument base odpowiada za liczbę, którą chcemy zamienić na ciąg, `a` argument `stlen` określa długość tego ciągu. Na początku metoda `setLength`

W Sieci

- *http://msdn2.microsoft.com/en-us/library/aa302371.aspx* – artykuł J. McCaffrey, Using Permutations in .NET for Improved Systems Security, 2003 MSDN,
- *http://www.codeproject.com/KB/security/Hacking_BruteForce.aspx* – artykuł F. Waeytens, A small and elegant bruteforcing class, 2006 Code Project,
- *http://java.sun.com/j2se/1.4.2/docs/api/java/math/BigInteger.html* – specyfikacja klasy BigInteger,
- *http://pl.wikipedia.org/wiki/Atak_brute_force* – opis ataku brute force wraz z przykładową implementacją w języku C.

Terminologia

- *Atak brute force* – jest to tzw. atak siłowy. Polega on na sprawdzeniu wszystkich możliwych kombinacji ciągów w celu poszukiwania hasła lub klucza szyfrującego. Jest to atak najbardziej prymitywny, niemniej jednak potrafi być skuteczny. W teorii zawsze gwarantuje sukces, jednak przy haśle składającym się z większej liczby znaków jego czas wykonywania może być bardzo długi.
- *Atak słownikowy* – jest zbliżony do ataku brute force. Polega on również na sprawdzaniu pewnych ciągów. Jednak ich lista jest ograniczona do pewnego podzbioru – słownika. Istnieje wiele sposobów tworzenia słownika. Ogranicza on czas potrzebny do odgadnięcia hasła, jednak nie gwarantuje sukcesu.

resetuje nam referencję klasy `StringBuilder`, ustawiając jej długość na zero. Zamiast tego moglibyśmy tutaj użyć operatora `new`, który tworzyłby nowy egzemplarz, jednak takie rozwiązanie zmniejszyłoby wydajność tej metody. Budowanie nowego egzemplarza klasy trwa dosyć długo – właśnie dlatego referencja `s` jest powoływana do życia w metodzie `main`, a nie tutaj. W pętli `for` będziemy wyłuskiwać kolejne znaki tworzące ciąg. Znaki te dodawane są do pola s za pomocą metody `append`. Aby obliczyć dla danej pozycji, jaki znak z alfabetu ona reprezentuje, musimy zastosować dzielenie modulo zmiennej `base` przez długość alfabetu. Następnie powinniśmy zmniejszyć zmienną `base` poprzez zwykłe podzielenie jej przez długość alfabetu. W Javie można to robić za pomocą jednej metody `divideAndRemainder`. Zwraca ona dwuelementową tablicę, w której znajduje się wynik standardowego dzielenia (*indeks 0*) oraz dzielenia modulo (*indeks 1*). Po zakończeniu pętli zwracamy uzyskany ciąg. I to wszystko. Program jest już gotowy. Zachęcam do jego testów.

Jedną z podstawowych metod testowania szybkości algorytmów, jest ich czas wykonywania. Przed wywołaniem metody realizującej badany algorytm można umieścić linijkę:

```
long start = System.currentTimeMillis();
Po wywołaniu metody umieszczamy w kodzie wpis:
long stop = System.currentTimeMillis();
```

Teraz wartość *stop-start* odzwierciedla nam czas działania programu. Celowo użyłem tu słowa *programu*, ponieważ taka metoda mierzy czas działania programu (lub jego fragmentu) w systemie operacyjnym. A system operacyjny może w trakcie jego działania przełączać się pomiędzy pamięcią fizyczną a plikiem wymiany na dysku, może przełączać zadania, zmieniać priorytet, obsługiwać jakieś zdarzenie itd. Jest to więc metoda mało wiarygodna. Dodatkowo głównym czynnikiem, które będzie generować opóźnienia w programie, są operacje *wejścia/wyjścia*, czyli np. wypisywanie ciągów do pliku czy na ekran. W związku z tym programy wykorzystujące czy to rekurencję, czy to zaproponowaną przeze mnie metodę będą działały z podobną prędkością właśnie ze względu na operacje *we/wy*.

Pisząc ten artykuł wyszedłem z założenia, że odrobina matematyki programistom na pewno nie zaszkodzi. Przedstawione rozwiązanie odbiega od standardowych metod, które stosuje się zwykle w takich przypadkach. Być może nie jest optymalne i można je jeszcze przyspieszyć. Jednak sama idea, oprócz tego, że nietypowa – jest prosta, a to przecież znacznie ułatwia implementację. Wiedza tu przedstawiona może posłużyć nie tylko do prób uzyskania hasła, ale przede wszystkim w testach bezpieczeństwa. Łatwo możemy zbudować z tego kodu klasę, której będziemy używać w innych aplikacjach.

Sławomir Orłowski
Z wykształcenia fizyk. Obecnie jest doktorantem na Wydziale Fizyki, Astronomii i Informatyki Stosowanej Uniwersytetu Mikołaja Kopernika w Toruniu. Zajmuje się symulacjami komputerowymi układów biologicznych (dynamika molekulama) oraz bioinformatyką. Programowanie jest nieodzowną częścią jego pracy naukowej i dydaktycznej. Ma doświadczenie w programowaniu w językach C, C++, Delphi, Fortran, Java, C# i Tcl. Współzałożyciel i koordynator grupy .NET WFAiIS. Jest autorem artykułów i książek informatycznych. Strona domowa: *http://www.fizyka.umk.pl/~bigman*.
Kontakt z autorem: *bigman@fizyka.umk.pl*

PRZEMYSŁAW ŻARNECKI

Hakowanie pakietów biurowych

Stopień trudności

Z pakietu biurowego korzysta w zasadzie każdy. Nie wszyscy jednak zdają sobie sprawę, że nieodpowiedzialne użytkowanie może skończyć się wręcz tragicznie. Pakiety posiadają liczne luki, dzięki którym intruz może wręcz przejąć kontrolę nad komputerem, z drugiej – sami użytkownicy go w tym często wspomagają.

Z ARTYKUŁU DOWIESZ SIĘ

jak ważna jest problematyka bezpieczeństwa pakietów biurowych,

jakie potencjalne zagrożenia niosą za sobą dziury w używanym przez ciebie programie,

jakie zachowania użytkownika sprzyjają wykorzystaniu przez intruza luki w pakiecie biurowym,

jakie są podstawowe mechanizmy przeciwdziałania zagrożeniu.

CO POWINIENEŚ WIEDZIEĆ

w zasadzie wszystkie programy biurowe posiadają mniejszą lub większą liczbę luk, które mogą umożliwić nawet pełne przejęcie kontroli nad twoim komputerem,

poziom bezpieczeństwa komputera zależy nie tylko od stosowanych zabezpieczeń, lecz również od samego użytkownika,

otwieranie zupełnie nieznanych plików może skończyć się wręcz tragicznie,

zawsze korzystaj z aktualizacji krytycznych,

wreszcie – nie daj się omamić mitowi bezpiecznego programu czy systemu, o bezpieczeństwie decydujesz przede wszystkim ty.

Bezpieczny pakiet biurowy jest często gwarantem stabilnej pracy komputera domowego czy firmowego. Bardzo często użytkownicy skupiają się tak bardzo na kwestiach bezpieczeństwa swojego systemu komputerowego, że zapominają o tym, iż jego elementem są również używane na co dzień programy, chociażby biurowe. Dotyczy to wszystkich systemów operacyjnych. Nie jest to bynajmniej kwestia związana z wirusami, od których wolny jest np. taki Linux. Czy mając ten system można czuć się całkowicie bezpiecznie? Wiele osób tak ma i często spotyka je nieprzyjemna niespodzianka. Nie ma bowiem oprogramowania idealnego. Nawet w Linuksie może zdarzyć się program, który przyczyni się do włamania do systemu. Wystarczy, że umożliwi manipulację lub nawet skasowanie danych, nad którymi użytkownik pracuje.

Z każdego programu, niezależnie od systemu, należy korzystać umiejętnie. W artykule podam kilka przykładów dziur – luk w zabezpieczeniach popularnych pakietów biurowych, a więc przede wszystkim MS Office i OpenOffice. Skupię się głównie na najnowszych wersjach pakietów. Omawianie każdej pojedynczej dziury można by uznać za swoiste polowanie na czarownice. W artykule chciałbym raczej zwrócić uwagę na to, jakiego rodzaju luki pojawiają się w pakietach biurowych, jak również – kiedy błędy te stają się rzeczywistym problemem. Artykuł jest próbą zwrócenia uwagi na to, gdzie czyhają pewne niebezpieczeństwa. Mam nadzieję, że trafi on przede wszystkim do początkujących i średniozaawansowanych użytkowników.

Kolejność prezentowanych informacji nie ma najmniejszego znaczenia. Zapraszam do lektury.

MS Office 2007

Najnowszy, wręcz flagowy, program Microsoftu został jako pierwszy w historii poddany skomplikowanym procedurom bezpieczeństwa, które miały zagwarantować, że oprogramowanie jest bezpieczne. Złośliwi zwracają uwagę, że nie tyle zostało ono pierwszy raz poddane testom, co pierwszy raz je przeszło – w związku z czym Microsoft robi szum. Jakby nie było, w swoich komunikatach gigant z Redmond wskazuje, iż postawił na bezpieczeństwo pakietu. Jaka jest rzeczywistość?

Pierwszą poważną lukę odkryto w lutym 2007 roku, cztery tygodnie po wydaniu konsumenckiej wersji pakietu. Firma eEye Digital Security poinformowała MS, że pierwsza krytyczna dziura dotyczy MS Publisher. Okazało się, że zastosowany w nim format plików umożliwia osobom postronnym zdalne uruchomienie niebezpiecznego kodu. W ostateczności może to zostać wykorzystane do przejęcia kontroli nad systemem. Dzieje się tak, jeżeli osoba uruchamiająca plik zalogowana jest jako administrator. Generalnie atakujący otrzymuje takie uprawnienia, jakie posiada zalogowany użytkownik. Sama luka wykorzystuje błąd przy czyszczeniu pamięci w trakcie zapisywania danych z dysku do pamięci. Microsoft uznał tę informację i opracował odpowiednią łatę. Jako datę publikacji

informacji ze strony MS można podać 10 lipca. W okresie pomiędzy zgłoszeniem a prezentacją uaktualnienia jedyną ochroną dla użytkownika był zdrowy rozsądek i nie włączanie plików nieznanego pochodzenia.

W czasie, gdy wykryto tę lukę, pojawiła się jeszcze jedna informacja, która spowodowała wzburzenie wśród wielu użytkowników pakietu. Mianowicie, Office 2007 bez wiedzy użytkownika wysyła informacje o komputerze – co więcej, Microsoft nawet nie próbował tego dementować. Przedstawiciele korporacji tłumaczyli, że – przede wszystkim – nie są to bynajmniej dane osobowe. Mają one być związane rzekomo z procesem aktualizacji pakietu. Microsoft w ten sposób ma się dowiadywać, jaki procent osób dokonuje w ogóle aktualizacji, jak również – ile aktualizacji kończy się powodzeniem, ile zaś nie. Podobno dzięki tym danym może również ustalić, jakie były tego przyczyny.

Luki o podobnym charakterze występowały i nadal występują w starszych wersjach pakietu. Jeszcze w lutym 2008 ukazały się aktualizacje do luk bezpieczeństwa dla MS Office 2000 oraz 2003. Dziura związana była po raz kolejny z możliwością uruchomienia niebezpiecznego kodu, co umożliwiało przejęcie pełnej kontroli nad systemem. Niebezpieczeństwo dotyczyło zarówno użytkowników Windowsa, jak również Mac OS.

Pisanie o dziurach w jakimkolwiek programie nie powinno mieć bynajmniej formy ataku. Nie ma programów idealnych. Microsoft z wielu przyczyn jest ulubionym celem ataku zarówno hakerów, jak również samych mediów. Wieloletnie zaniedbania bezpieczeństwa mają z pewnością na to wpływ. Informacje o większości dziur są zazwyczaj lakoniczne i zawierają podstawowe rady, jak się przed nimi uchronić. Najwłaściwszą metodą jest instalacja odpowiednich uaktualnień. Jeżeli takowych jeszcze nie ma – a wskazany przykład pokazał, że taki stan rzeczy często trwa o wiele za długo – to po prostu trzeba uważać i uważnie wystrzegać się nieznanych plików.

Informacje na temat poszczególnych dziur pojawiają się zazwyczaj w Internecie w formie różnego rodzaju raportów. Niektóre z nich należą bezpośrednio do MS – z tym, że pojawiają się one zazwyczaj z chwilą ukazania się poprawki. Wiele osób krytykuje tę praktykę, ma ona jednak także swoich zwolenników. Zdaniem tych drugich przedwczesne ujawnienie informacji, zwłaszcza zbyt szczegółowych, jeszcze bardziej przyczynia się do zwiększenia niebezpieczeństwa, ponieważ informacja może dotrzeć (czytaj *na pewno dotrze*) do osób, które ją wykorzystają w złych celach. Ma to jakiś sens. Istnieje nawet określenie na ataki, które następują po tym, jak ogłosi się przedwcześnie podatność jakiegoś miejsca na atak. Taki atak to *zero-day exploit*.

W każdym razie wcześniej trzeba liczyć głównie na doniesienia prasowe. W oparciu o nie i o informacje Microsoftu można wskazać na pewne klasy najważniejszych dziur (niektóre są załatane, inne ciągle się pojawiają).

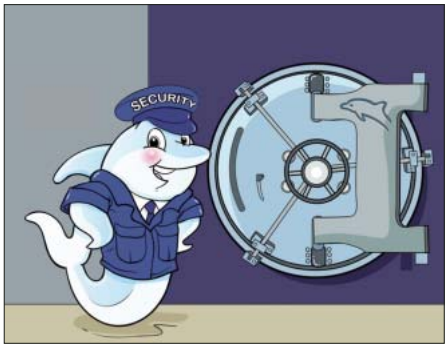
Większość z dziur związana jest z możliwością uruchomienia niepożądanego kodu, a nawet zdalnego przejęcia kontroli nad systemem operacyjnym. Różnią się one między sobą przede wszystkim co do sposobu, w jaki to powodują. Jedna z wykrytych na wiosnę poprzedniego roku dziur doprowadzała do przejęcia kontroli nad systemem poprzez przepełnienie stosu. Taki atak polega na tym, że do programu wysyłana jest zbyt duża liczba danych. Dziura w takiej sytuacji objawia się tym, że zamiast na przykład zakończyć działanie, poinformować o problemie, lub uniemożliwić niebezpieczną akcję, program wykonuje czynność, której użytkownik zupełnie się nie spodziewa – przyznaje danemu plikowi zbyt duże uprawnienia. Można to porównać do sytuacji życiowej, w której bombardowana zbyt dużą ilością informacji jednostka ma spory problem z ich selekcją i podjęciem decyzji. W takich przypadkach niekoniecznie podejmuje się te najważniejsze (patrz: wizyta w instytucjach pożyczających pieniądze i szczegółowe warunki umowy – przysłowiowe zero prowizji).

Kwestia uprawnień jest w gruncie rzeczy sprawą naprawdę istotną. Część dziur nie miałaby zapewne zbyt wielkiego znaczenia, gdyby nie fakt, że większość użytkowników systemów Microsoftu pracuje zazwyczaj na kontach administratora, a nie zwykłego użytkownika. Weźmy pewną lukę, która pojawiła się w Excelu. W Sieci pojawiły się specjalnie spreparowane pliki z niebezpiecznymi elementami, które mogą pozwolić na włamanie się do komputera. Z tym, że owo działanie nie jest możliwe, jeżeli próba uruchomienia kodu nastąpiła podczas pracy na koncie zwykłego użytkownika.

Kolejna dziura związana jest z sytuacją, która spotyka nas bardzo często. Ot, dostajemy jakąś rzekomo pilną wiadomość z załączonym plikiem MS Office. W takim pliku może znajdować się rysunek. Dziura ujawnia się podczas analizy obiektu graficznego. Specjalnie przygotowana grafika może posłużyć jako sposób do włamania się do systemu. Pliki z takimi obiektami znajdują się często w różnego rodzaju wiadomościach, łańcuszkach, rzekomych reklamach, słowem – w spamie. Po co to otwieramy? Spreparowany plik powoduje uszkodzenie pamięci, w konsekwencji atakujący może uzyskać wszelkie

W Sieci

- *www.cert.pl*,
- *www.dobreprogramy.pl*,
- *www.idg.pl*,
- *www.centrumxp.pl*,
- *www.securitywortal.pl*,
- *http://www.microsoft.com/poland/technet/security/default.mspx*,
- i wiele, wiele innych.



Rysunek 1. *Bezpieczeństwo danych to podstawa*

prawa administratora. Co to oznacza? Może instalować, kopiować, kasować programy, tworzyć i zarządzać kontami użytkowników. Pechowy użytkownik może pewnego razu nie zalogować się do własnego komputera, bowiem ktoś mu zmieni hasło.

Kolejna luka może mieć również niezbyt przyjemne konsekwencje, przede wszystkim jeżeli użytkownik zadziała nieodpowiedzialnie. Okazało się, że Microsoft Office SharePoint Server 2007 zawiera lukę związaną z wykorzystaniem skryptów. Mianowicie nie sprawdza dostatecznie, czy uruchamianie załączniki nie zawierają jakiegoś niepożądanego skryptu. Taki skrypt podnosi poziom uprawnień niepożądanego gościa – co prawda tylko w ramach programu. Niemniej za jego pomocą atakujący może poznać zawartość pamięci podręcznej, a w dalszej kolejności dowiedzieć się wiele o stacji roboczej. Nie stanie się tak, jeżeli wpierw nie nakłoni kogoś do kliknięcia spreparowanego linku!

Moglibyśmy omówić jeszcze przynajmniej kilka takich dziur. Większość z nich jest prędzej czy później łatana przez Microsoft. Trudno powiedzieć, jaki procent luk jest wykryty przez samą korporację, ile zaś wychwytują liczne podmioty zajmujące się bezpieczeństwem systemów komputerowych. Sam spis luk zresztą nie jest aż tak ważny. Coś, co w trakcie pisania tekstu jest poważną luką, tuż po opublikowaniu może być już dawno załatane, za to w międzyczasie pojawią się nowe podatności.

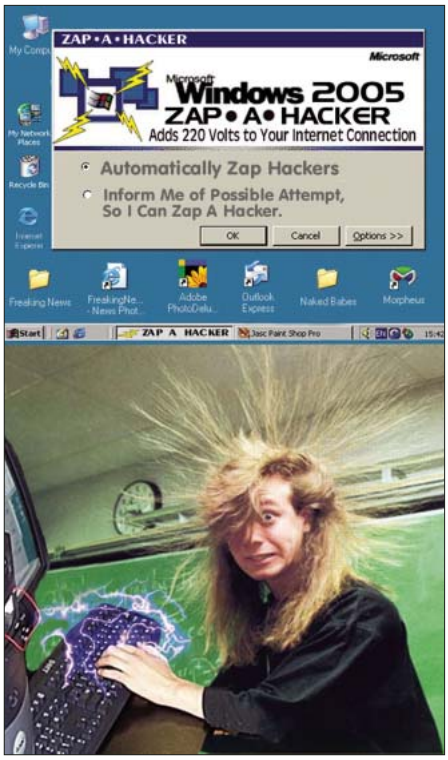
Ważny jest z pewnością fakt, że nowy produkt Microsoftu nie jest taki bezpieczny, jak to się jego producentowi wydaje. Owszem, jak na razie dziur jest o wiele mniej, niż w poprzednich wersjach. Wynika to jednak z tego, że pakiet jest o wiele krócej w sprzedaży – zresztą cały czas są jeszcze znajdowane dziury w poprzednich wersjach Office. Zastanawiające jest również, że nie odbiegają one w swoich konsekwencjach od tego, co mamy w najnowszym pakiecie. Niestety, wiele z tych luk ujawnia się dopiero przy nieodpowiedzialnym zachowaniu użytkownika. Nawet najbezpieczniejszy program nam nie pomoże, jeżeli nie zaprzestaniemy uruchamiania wielu kompletnie nieznanych nam linków czy załączników do wiadomości mail. Osobiście przestrzegam przed wieloma, często szczytnymi w swoich założeniach, łańcuszkami.

Generalnie zaleca się regularne aktualizowanie produktu. Trudno wymagać od użytkownika, żeby czytał każdą informację o dziurze. Większość informacji do niczego mu się nie przyda. Bezpieczny nie będzie, dopóki nie pobierze aktualizacji. W międzyczasie jedynym rozwiązaniem jest niestety rozwaga.

OpenOffice

W przypadku darmowego pakietu OpenOffice sprawa nie jest tak oczywista – przynajmniej tak na pierwszy rzut oka. Zdaniem wielu specjalistów, oprogramowaniu brakuje przede wszystkim systematycznych testów pod kątem bezpieczeństwa, co wynika chociażby z jego mniejszej popularności. Generalnie panuje powszechna opinia, że pod względem bezpieczeństwa OpenOffice przewyższa on pakiet Microsoftu. Nie jest ona bynajmniej bezpodstawna, jednakże również i tutaj trafiają się różnego rodzaju luki. Szczerze mówiąc, nie różnią się one zbytnio w skutkach

Rysunek 2. *Mimo mniej lub bardziej szczerych intencji twórców oprogramowania, komputer nie jest wystarczająco zabezpieczony*



Rysunek 3. *Wiele negatywnych konsekwencji wynika z nieodpowiedzialnego działania użytkownika*

i mechanizmach działania (poza typowymi szczegółami technicznymi) od tego, co można spotkać w MS Office.

Najpierw posłużę się przykładem jeszcze z 2006 r., w pewnym sensie aktualnym. W owym czasie francuskie ministerstwo obrony rozważało migrację z MS Office na OpenOffice. Zleciło w tym celu odpowiednią analizę. Wykazała ona rzekomo (szczegółów nie znamy, bowiem była tajna, ale – jak to zwykle bywa – są pewne przecieki), że w niektórych zastosowaniach OpenOffice jest nawet mniej bezpieczny od MS Office. Sprawa miała dużo wspólnego z makrami. Generalnie pakiet nie chroni wystarczająco przed niepożądanym kodem. Ustawienie wysokiego poziomu bezpieczeństwa nie zawsze zabezpieczy nas przed intruzem. Zawsze istnieje coś takiego, jak bezpieczna lokalizacja, którą jest zazwyczaj odpowiedni katalog w miejscu instalacji. Z niej każde makro zostanie uruchomione zawsze. Kolejnym poziomem ochrony są co prawda różnego rodzaju podpisy i certyfikaty, ale co w sytuacji, jeżeli zostaniemy namówieni do zaakceptowania fałszywego? Od roku 2006 sporo się w tej materii poprawiło, jednakże ciągle jest to słabsza strona OpenOffice.

Moim zdaniem widzimy to samo, co w sytuacji najlepszych drzwi antywłamaniowych. Jeżeli wpuścimy złodzieja do domu, to jest to tylko i wyłącznie nasza wina. Tak jest również z bezpieczeństwem pakietów, o czym wspominałem już przy Microsofcie.

Kilka rzeczywistych dziur również jest związanych z uruchomieniem niebezpiecznego kodu. Jeżeli użytkownik kliknie w całkowicie nieznany odnośnik i uruchomi nieznany plik, musi liczyć się z konsekwencjami. Na przykład deweloperzy Debiana (jednej z najpopularniejszych dystrybucji Linuksa) odkryli dwie takie dziury. Jedna z nich doprowadzała do zawieszenia komputera. Druga pozwalała na uruchamianie poleceń w shellu (tyczy się Linuksa). Oczywiście najpierw trzeba zostać namówionym do otwarcia niebezpiecznej zawartości. Niemniej dziury są – a raczej były – poważne.

Również w OpenOffice wykryto kilka dziur, które umożliwiają przejęcie kontroli nad komputerem po kliknięciu odnośnika.

Czy OpenOffice jest więc bardziej bezpieczny od MS Office? Analiza dodatkowych danych pozwala stwierdzić, że mimo wszystko tak. Jakie to dane? Dobrym źródłem są na przykład deweloperzy Debiana, którzy non stop pracują nad rozwojem swojej dystrybucji, a więc również i stosowanego w niej Otwartego Oprogramowania. Debian znany jest z bardzo kompleksowego i przede wszystkim odpowiedzialnego podejścia do kwestii bezpieczeństwa. Za bezpieczeństwo odpowiada w zespole znaczna liczba osób. Dla nieobeznanych z tą tematyką – nowe wersje tej dystrybucji powstają stosunkowo rzadko, przynajmniej jak na warunki Linuksowe. Jednak to, co już zostanie wypuszczone, jest maksymalnie przetestowane na wszelkie sposoby. W przerwach pomiędzy poszczególnymi wersjami deweloperzy mają czas – zresztą taki jest ich cel – na testowanie aplikacji pod każdym kątem. Wszelkie aktualizacje dystrybucji to nie tyle nowe wersje programów (takie ukazują się rzadko, bowiem postawiono w niej na stabilność i sprawdzone rozwiązania), lecz właśnie aktualizacje krytyczne.

Z analizy pracy zespołu można wynieść informacje, że luk dla OpenOffice nie ma zbyt dużo, zaś łatanie istniejących odbywa się stosunkowo szybko. Członkowie społeczności Debiana udostępniają wyniki swoje pracy, stąd poprawkę otrzymują wszyscy zainteresowani. Nad rozwojem OpenOffice czuwają również inni deweloperzy, jednak to był najbardziej spektakularny przykład.

Podsumowanie

Gdyby podjąć próbę przeanalizowania mniej popularnych pakietów biurowych, prawdopodobnie uzyskano by podobne informacje. Nie można również przypominać, że im pakiet bardziej popularny, tym więcej ma użytkowników – w tym takich, którzy szukają wszelkiego rodzaju sposobów na uprzykrzenie życia innym.

Mimo rzekomych starań, najnowszy pakiet Microsoftu posiada sporo dziur. Jest to spory problem. Jeszcze większym jest jednak zróżnicowany, często zbyt długi czas reakcji. Pierwsza dziura została załatana po mniej więcej dwóch miesiącach. To zdecydowanie za dużo.

W przypadku OpenOffice liczna dziur jest o wiele mniejsza. Nie są one oczywiście przez to mniej groźne. Plusem jest, że ich likwidacja trwa o wiele krócej – często jest to kwestia nawet kilku dni. Poza tym stosunkowo restrykcyjny cykl włączania OpenOffice do wielu popularnych dystrybucji Linuksa sprzyja wczesnemu wykrywaniu i likwidowaniu dziur.

Moim zdaniem spór o to, który pakiet cechuje się wyższym poziomem bezpieczeństwa, jest jednak jałowy. Przepraszam za ten kolokwializm. Niemniej sporo zależy tak naprawdę od samego użytkownika. Dla przykładu w Linuksie nie pracuje się dzień w dzień na koncie administratora, przez co pewne zagrożenia są znacznie zredukowane. W Windowsie, zwłaszcza na komputerach domowych, praca na koncie z podwyższonymi uprawnieniami jest nagminna. Czy jest to winą tylko użytkownika? W dużym stopniu tak. Tylko, że prawie każdy Linux wymaga utworzenia przynajmniej jednego konta zwykłego, a niektóre blokują korzystanie z konta administratora do standardowej pracy. Microsoft nie zwraca na to specjalnej uwagi (wystarczyłoby parę plansz w czasie instalacji) – chyba, że w biuletynach bezpieczeństwa, w mało czytelnych sekcjach, do których poza specjalistami prawie nikt nie dociera.

Po lekturze tego krótkiego artykułu chciałbym, abyście zwrócili uwagę na fakt, że każdy pakiet ma mniej lub więcej dziur. Jednakże ostatecznie poziom bezpieczeństwa zależy również i od Was. Koniecznie pobierajcie wszelkie uaktualnienia, zwłaszcza krytyczne. Jeżeli temat interesuje Was jeszcze, zapraszam na liczne portale informacyjne. Kilka adresów znajdziecie w ramce *W Sieci*.

Przemysław Żarnecki
Autor para się dziennikarstwem, próbuje swoich sił w różnych przedsięwzięciach. Nie byłby specjalnie zadowolony, gdyby zniknęła mu część danych, bo kliknął link. Bezpieczeństwo danych jest dla niego priorytetem i nigdy nie bierze udziału w łańcuszkach.
Kontakt z autorem: *p.zarnecki@plusnet.pl*

PIOTR ŁASKAWIEC

Peach 2.0 – rozbudowany fuzzing

Stopień trudności

Testowanie aplikacji pod kątem ewentualnych luk na pewno nie jest czynnością łatwą. W celu zwiększenia komfortu pracy i szybkości wykonywanych działań warto posłużyć się odpowiednimi, profesjonalnymi narzędziami. W niniejszym artykule poznamy jedno z nich – Peach 2.0.

Z ARTYKUŁU DOWIESZ SIĘ

jak używać Peach 2.0,

jak wygląda budowa Peach 2.0,

jak generować pseudolosowe dane z wykorzystaniem Peach,

jak konstruować w pełni funkcjonalne fuzzery,

jak zminimalizować ryzyko wystąpienia ewentualnych błędów.

CO POWINIENEŚ WIEDZIEĆ

powinieneś wiedzieć, jak działają fuzzery,

powinieneś znać podstawy Pythona,

powinieneś znać podstawy języka XML.

Jakiś czas temu na łamach hakin9 ukazał się mój artykuł dotyczący procesu *fuzzingu* z zastosowaniem języka Python. Opisałem tam najpopularniejsze *fuzzery* i *frameworki* napisane w tym języku oraz przedstawiłem w zarysie teorię testowania aplikacji. W tym artykule postaram się zagłębić w tę jakże ciekawą tematykę i bliżej opisać jeden z najnowszych (a także najlepszych) obecnie dostępnych *frameworków* wyspecjalizowanych w tworzeniu własnych programów testowych – Peach 2.0.

We wcześniejszym, wspomnianym już artykule, opisałem poprzednią wersję Peach. Już wtedy był to świetny produkt, który swoimi możliwościami przewyższał konkurencję. Cechował się m. in. krótkim czasem tworzenia wyjściowej aplikacji, rozszerzalnością oraz możliwością wielokrotnego wykorzystania raz stworzonego kodu. Co więcej, jego główną zaletą była możliwość testowania praktycznie dowolnego protokołu. Niestety, Peach 1.0 nie był idealny – jego nauka przysparzała początkującym programistom wielu kłopotów, a logika działania nie była dla wszystkich jasna. Wcześniejsza wersja nie posiadała także usystematyzowanego wzorca tworzenia aplikacji, który wprowadzałby jednolity styl pisania *fuzzerów* na podstawie szablonów. Najnowsza, stabilna wersja naprawia błędy poprzednika i oferuje kolejne, niespotykane dotąd możliwości. Mam nadzieję, że po przeczytaniu tego tekstu korzystanie z Peach 2.0 stanie się łatwiejsze i sprawi, że radość ze znalezionych błędów w aplikacjach będzie dużo częstszym doznaniem niż do tej pory.

Ewolucja Peach

Rozwój *frameworku* Peach postępuje niewątpliwie w dobrym kierunku. W kolejnej wersji, oznaczonej numerem 2.0, znalazło się wiele innowacyjnych rozwiązań, które wydatnie ułatwiają pracę. Znajomość składowych Peach i ich funkcji pozwoli na lepsze zrozumienie pozostałej części tekstu i pełne wykorzystanie możliwości całego pakietu.

Po pierwsze, zaszły naturalne dla kontynuacji projektu zmiany, czyli zachowano zalety poprzedniej wersji oraz poprawiono i zoptymalizowano wszystkie wymagające tego elementy. W związku z tym tworzenie pełnowartościowych aplikacji jest jeszcze prostsze (dzięki zmianie składni) i szybsze, a wszystkie aspekty związane z rozszerzalnością i ponownym wykorzystaniem kodu pozostały nietknięte (były już wcześniej wystarczająco dopracowane).

Po drugie, dodano wiele ciekawych elementów zwiększających funkcjonalność *frameworku*. Nowa wersja Peach nie wymaga od swoich użytkowników znajomości Pythona (choć jest to zalecane). Co więcej, możemy stworzyć w pełni działający *fuzzer* bez napisania choćby jednej linijki kodu. To wszystko jest zasługą tzw. warstwy definicji danych (ang. *data definition layer* – DDL), której próżno szukać w poprzedniej wersji. Składa się ona z pełnego schematu XML oraz zależności

pomiędzy poszczególnymi typami danych. Co nam to daje? Możliwość ponownego wykorzystania raz stworzonych definicji danych, podział na wyspecjalizowane elementy wykonujące określony zakres działań oraz rozgraniczenie definicji danych i procesu generowania pseudolosowych wartości. Oprócz tego Peach 2.0 dostarczany jest z szeregiem *API* (do użycia m. in. w Python, .NET i Java) i narzędziem *Peach Builder* (służącym do stworzenia w pełni działającego *fuzzera* za pomocą kilku kliknięć myszką). Wiemy już zatem, jak kształtuje się ogólna idea programu, ale nie potrafimy jeszcze wykorzystywać jego możliwości. Aby zacząć pisać własne *fuzzery*, należy poznać wewnętrzną budowę Peach, a w szczególności jego DDL.

W głąb DDL

Budowa fuzzerów za pomocą Peach przypomina układanie klocków w jedną, świetnie działającą całość. Do naszej dyspozycji oddanych jest kilka składowych, które możemy dowolnie komponować w zależności od potrzeb. W Ramce *Składowe DDL* przedstawione są wszystkie elementy warstwy definicji danych (DDL). W celu lepszego zrozumienia praw rządzących budową *fuzzerów* warto opisać własności poszczególnych składowych.

Przestrzenie nazw

Jeżeli planujemy test programu wykorzystującego popularny protokół, prawdopodobnie nie będziemy musieli implementować własnych mechanizmów *fuzzingu*. Jest wielce prawdopodobne, że ktoś inny stworzył odpowiedni program wcześniej i możemy go wykorzystać do własnych celów. Przestrzenie nazw, połączone z importowaniem odpowiednich schematów XML, pozwalają tworzyć nam programy na podstawie innych aplikacji. Przypuśćmy, że ktoś napisał za pomocą Peach schemat XML odpowiedzialny za testowanie protokołu HTTP i udostępnił odpowiedni plik w sieci. Możemy wtedy w naszym schemacie dodać następującą linijkę:

```
<Include ns="httpfuzz" src="file:http_
                    fuzzer.xml" />
```

Powyższy kod importuje schemat XML i przypisuje mu przestrzeń nazw *httpfuzz* (za pomocą parametru *ns*). Co więcej, do schematów XML możemy odwoływać się zdalnie (np. poprzez adres internetowy), bez konieczności pobierania ich na dysk.

Typy danych

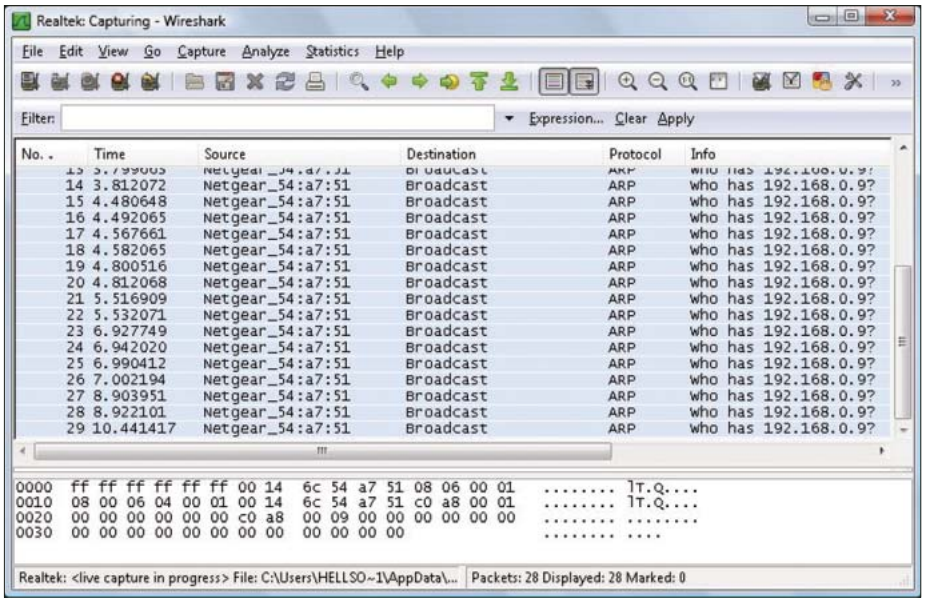
Prawdopodobnie najważniejsza część DDL. Typy danych służą do generowania odpowiednio dostosowanego wyjścia w postaci zmiennych pseudolosowych. Możemy je podzielić na liczby (*Numbers*), ciągi znakowe (*Strings*), flagi (*Flags* – odpowiedzialne za ustawianie parametrów), dane nieznanego typu (*Blob*), sekwencje (*Sequences* – uporządkowany zbiór elementów) i bloki (*Blocks*).

Właściwości powyższych typów powinny być oczywiste. Jedynym elementem, który zostanie opisany bardziej szczegółowo, będzie blok. Jest to nic innego, jak zbiór elementów zróżnicowanego typu. Może on zawierać ciągi znakowe, liczby i inne bloki. Tworzymy go, aby posługiwać się i dokonywać operacji na jednym elemencie (grupującym inne elementy), a nie na wielu pojedynczych obiektach – co znacząco utrudniłoby pracę. Bloki mają duże znaczenie w korelacji z szablonami.

Szablony

Szablony opisują format danych i zależności pomiędzy poszczególnymi elementami. Szablony, podobnie jak bloki, mogą składać się z elementów różnych typów: innych szablonów, bloków, liczb, ciągów znakowych itd. Pozwalają np. na modelowanie testowanych protokołów. Przypuśćmy, że naszym zadaniem jest sprawdzenie programu wykorzystującego protokół *ICMP*. Na Listingu 1. przedstawiono użycie szablonu do modelowania ramki *ICMP*. Jak wiemy, ramka protokołu *ICMP* składa się z pól: *Typu*, *Kodu*, *Sumy Kontrolnej* i *Danych* (które są dołączane opcjonalnie). Naszym zadaniem jest stworzenie abstrakcyjnego modelu, który będzie wykorzystywany do przekazywania

Rysunek 1. *Visual Studio 2005 w akcji*



Rysunek 2. *Wireshark*

pseudolosowych danych. Cały model napisany jest w XML. Zaczynamy od deklaracji szablonu poprzez znacznik `<Template>`. Następnie deklarujemy trzy liczby będące logicznymi odpowiednikami pól: *Typ*, *Kod* i *Suma Kontrolna*. Warto zwrócić uwagę na liczbę reprezentującą Sumę Kontrolną i znacznik `<Relation>`, umiejscowiony zaraz po jej deklaracji. Znacznik ten reprezentuje poruszoną wcześniej kwestię zależności pomiędzy danymi. Pozwala on określić relacje panujące pomiędzy dwoma typami danych. W naszym przypadku informuje on, że numer określający sumę kontrolną jest automatycznie obliczany przez Peach 2.0 na podstawie parametru type. Oczywiście istnieje szereg innych relacji, takich jak rozmiar (zamiast `type="checksum"` wstawiamy `type="size"`) czy liczba elementów (wstawiamy `type="count"`).

Testy (Tests)

W testach określamy szablony, które mają zostać wykonane. Definiujemy także docelowe miejsce testów. Miejsce testów określa cel, do którego będą przekazywane wygenerowane dane. Może to być ekran lub np. określony port konkretnego komputera w sieci. Na Listingu 2. pokazany jest przykład testu, w którym określono *wydawcę* (poprzez parametr *Publisher*), natomiast na Listingu 3. – przykład testu, dla którego celem jest komputer o wskazanym adresie IP i porcie. Warto zwrócić uwagę na sposób przekazywania parametrów do wydawcy (parametr *Param*). W obu testach użyłem niezdefiniowanego, abstrakcyjnego szablonu *MyTemplate*.

Listing 1. *Użycie szablonu do modelowania ramki ICMP*

```
<Template name="ICMP">
  <Number name="Typ" size="8" endian="network" />
  <Number name="Kod" size="8" endian="network" />
  <Number name="SumaKontrolna" size="16" endian="network">
      <Relation type="checksum" of=" ICMP " />
  </Number>
  <Blob name="Dane" />
</Template>
```

Listing 2. *Test z wywołaniem na ekran*

```
<Test name="MyTest" description="MyTest">
      <Template ref="MyTemplate" />
      <Publisher class="stdout.Stdout" />
</Test>
```

Listing 3. *Test, którego celem jest inny komputer*

```
<Test name="MyTest" description="MyTest">
      <Template ref="MyTemplate" />
      <Publisher class="tcp.Tcp">
         <Param name="host" value="192.168.0.4" />
         <Param name="port" value="456" />
      </Publisher>
</Test>
```

Listing 4. *Przykład uruchomienia*

```
<Run name="MyRun" description="MyRun">
<!–Lista testów do wykonania -->
    <Test ref="MyTest" />
</Run>
```

Uruchomienia (Runs)

Opisują one poszczególne uruchomienia *fuzzera*. Jest to jeden z najprostszych elementów struktury Peach. W ich ciele podaje się w większości wypadków tylko testy, które mają zostać uruchomione podczas startu programu. Przykład uruchomienia pokazuje Listing 4.

Możemy także dodać możliwość logowania każdego uruchomienia *fuzzera* poprzez dodanie do sekcji *Run* wpisu:

```
<Logger class="logger.Filesystem">
  <Param name="path" value="C:\log" />
 </Logger>
```

Inne elementy warstwy definicji danych

W czasie korzystania z Peach możemy natknąć się na inne, mniej lub bardziej przydatne, elementy warstwy definicji danych. Jednym z najciekawszych elementów jest tzw. *Nadzorca* (*Monitor*), który podejmuje określone działania na podstawie zaistniałych wydarzeń. Może on np. wywołać w odpowiedniej chwili *debugger* lub przechwytywać w czasie rzeczywistym pakiety w sieci i na podstawie informacji w nich zawartych generować pseudolosowe dane. Z reguły działają one niezależnie od samego szkieletu Peach. Aby *Nadzorcy* mogli efektywnie współpracować z Peach, potrzebne są elementy spajające i ułatwiające komunikację na linii *Peach–Monitor*. Rolę taką spełniają *Agenci* (*Agents*), którzy są kolejnym elementem DDL wartym poznania.

Piszemy pierwszy program za pomocą Peach

Znamy już schemat działania Peach i jego najważniejsze składowe, więc możemy teraz przejść do napisania pierwszego programu. Standardem w świecie informatyki stało się już pisanie programów, których jedynym przeznaczeniem jest wyświetlenie napisu *Hello World*. Nie będę odstępował od tej tradycji. Zanim jednak przejdziemy do pisania praktycznego kodu, musimy odpowiednio przygotować nasze środowisko pracy. Aby w pełni docenić zalety szybkiego programowania, nie tylko musimy korzystać z *frameworków* to umożliwiających, ale także wykorzystywać oprogramowanie wspomagające.

Ogólne informacje o projekcie Peach

Peach jest wieloplatformowym *frameworkiem* służącym do testowania oprogramowania pod kątem występowania błędów bezpieczeństwa. Napisany jest w całości w języku Python. Potrafi testować praktycznie dowolne dane wejściowe – począwszy od protokołów sieciowych, a kończąc na specyficznych formatach plików. Cechuje się szybkim procesem tworzenia *fuzzerów*.

Przygotowanie środowiska

Po pierwsze, musimy zainstalować Pythona na komputerze. W chwili obecnej Peach współpracuje jedynie z *ActivePython*, którego możemy pobrać ze strony *http://www.activestate.com/Products/activepython*. Po pomyślnym zainstalowaniu AP, możemy pobrać Peach. Wraz z *frameworkiem* zostaną zainstalowane wymagane komponenty, takie jak *Twisted* czy *wxPython*. Po wykonaniu wszystkich czynności instalacyjnych możemy przejść do wyboru edytora, w którym będziemy pisać nasz kod XML.

Autorzy Peach zalecają takie aplikacje, jak *Microsoft Visual Studio*, *oXygen* i *XML Spy*. Charakteryzują się one przyjemnym środowiskiem i wspierają uzupełnianie składni przyśpieszające programowanie. Możemy do nich zaimportować cały schemat XML pochodzący z Peach i uzyskać podczas pisania pomocne podpowiedzi.

Osobiście korzystam z *Visual Studio 2005*. Pozwala ono na tworzenie rozbudowanych plików XML i zapewnia szybki system uzupełniania składni.

HelloWorld w Peach

Nadszedł czas na napisanie naszej pierwszej aplikacji. Oczywiście ktoś może zadać pytanie, jaki jest sens tworzenia *fuzzera*, który tak naprawdę nie wykonuje żadnych czynności testowych, a jedynie wyświetla ciąg znaków na ekranie. Już odpowiadam… Zaprezentowanie w pełni funkcjonalnego i rozbudowanego *fuzzera*, który byłby wyspecjalizowany w testowaniu konkretnego programu, wymagałoby znacznie więcej miejsca, co technicznie jest niemożliwe. Moim zadaniem jest przedstawienie w praktyce logicznego modelu budowy *frameworku* Peach i zachęcenie do zgłębiania jego tajników we własnym zakresie.

Na Listingu 5. przedstawiony jest kod *WitajSwiecie.xml*. Rozpoczyna się on od wpisu przedstawiającego wersję języka XML i kodowanie. Potem następuje właściwy program rozpoczynający się znacznikiem `<Peach>`, po którym następują parametry informujące o położeniu schematu XML, jego zgodności z normami, opisie itd. Następnie należy dołączyć do projektu dwa pliki XML (*PeachTypes.xml* oraz *defaults.xml*). Jest to operacja wymagana dla wszystkich programów tworzonych za pomocą Peach. Po dołączeniu plików następują operacje omówione wcześniej – definiowanie szablonu, testu oraz deklaracja uruchomienia. Po wywołaniu programu poprzez komendę *python peach.py Witaj.xml* naszym oczom ukaże się napis *Witaj Swiecie*!.

Program ten przedstawia standardową procedurę tworzenia programów w Peach. Każdy test zaczynamy od zdefiniowania odpowiedniego szablonu na podstawie danych, którymi dysponujemy (protokół, format pliku itd.). Następnie należy odpowiednio zdefiniować test – w należyty sposób przekierować generowane dane i określić cel *fuzzingu*. Potem wystarczy jedynie uruchomić program. Z czasem, wraz ze wzrostem naszego doświadczenia, możemy uzupełniać nasz program o nowe funkcje i poddawać go modyfikacjom.

Oczywiście Peach potrafi także modyfikować przekazywane dane. Mamy

Pliki XML a Peach

W celu wykonania działań opisanych w przygotowywanych plikach XML, musimy przekazać je bezpośrednio do Peach jako parametr wywołania programu: python peach.py ‹plik›.xml.

Składowe DDL

- Przestrzenie nazw (*Namespaces*),
- Typy danych (*Data types*),
- Szablony (*Templates*),
- Uruchomienia (*Runs*),
- Testy (*Tests*),
- Inne – *Agenci* (*Agents*), Include, Import itd.

do dyspozycji mutatory, generatory losowe i generowanie danych na podstawie określonych wzorców. Jest to jednak temat na kolejny artykuł (a nawet książkę) i ciężko omówić go w pojedynczej publikacji.

Udogodnienia w Peach

Wyobraźmy sobie sytuację, w której w kilka chwil możemy w sposób automatyczny poddać testom dane przechodzące przez nasz komputer. Taką możliwość daje nam duet *Peach – Wireshark*. Za pomocą *Wireshark* możemy monitorować i zapisywać ruch generowany w naszej sieci. Możemy sprawdzać budowę protokołów, za pomocą których nasz komputer komunikuje się z aplikacjami zewnętrznymi. Aby dokonać w pełni automatycznego *fuzzingu* protokołu przechwyconego przez *Wireshark*, wybieramy pojedynczy pakiet i eksportujemy go do formatu *PDML* (*XML packet detail*). Następnie otwieramy nasz plik *PDML* i zapisujemy nazwę protokołu znajdującą się za znacznikiem `<proto>`. Ostatnim krokiem jest wygenerowanie pliku *XML* obsługiwanego przez Peach komendą: `pyton peach.py -s pdml <nazwa_protokolu> > wynikowy.xml`. Teraz możemy już uruchomić nasz plik (lub dalej go modyfikować wedle własnych potrzeb) i czekać na efekty pracy fuzzera.

Listing 5. *Witaj.xml w Peach*

```
<?xml version="1.0" encoding="utf-8"?>
<Peach xmlns="http://phed.org/2007/Peach" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
                    instance"
   xsi:schemaLocation="http://phed.org/2007/Peach ../peach.xsd"
description="Witaj Swiecie!!!">

   <!—Podstawowe dowiazania -->
   <Include ns="default" src="file:defaults.xml" />
   <Include ns="pt" src="file:PeachTypes.xml" />

   <!—Nasz podstawowy szablon -->
   <Template name="WitajSwiecie">

      <String value="Witaj Swiecie!" />

   </Template>

   <!—Nasz podstawowy test -->
<Test name="WitajSwiecieTest">
      <Template ref="WitajSwiecie " />
      <Publisher class="stdout.Stdout" />
   </Test>

   <!-- Configure a single run -->
   <Run name="WitajSwiecieRun" description="WitajSwiecieToConsole">

   <Test ref="WitajSwiecieTest" />

   </Run>

</Peach>
```

Peach Builder

Jeżeli nie mamy czasu na tworzenie złożonego schematu XML albo nie znamy podstaw tego języka, możemy posłużyć się narzędziem Peach Builder dostarczanym wraz z frameworkiem Peach. Jest to graficzna nakładka na Peach pozwalająca stworzyć funkcjonalny fuzzer (wykorzystując wszystkie opisane wcześniej elementy) za pomocą kilku kliknięć myszką. Program znajduje się w katalogu instalacyjnym Peach.

Podsumowanie

Artykuł ten miał za zadanie udowodnić, że pisanie fuzzerów nie musi być wcale trudne i nieciekawe. Za pomocą Peach 2.0 możemy osiągnąć bardzo dobre efekty w procesie testowania oprogramowania, poznając jedynie nieskomplikowaną logikę frameworku. Nawet osoby, które nie mają doświadczenia w programowaniu, mogą przy użyciu Peach tworzyć własne testy. Cały proces tworzenia oprogramowania testującego jest intuicyjny oraz oparty na kilku podstawowych obiektach i typach danych. Należy także zwrócić uwagę na możliwość współpracy z takimi narzędziami jak *Wireshark* oraz na udostępnieniu przez twórców Peach 2.0 (dla leniwych) frontendu *Peach Builder*. Są to niewątpliwe zalety Peach, których próżno szukać w innych produktach. Myślę, że nauka obsługi Peach i efektywnego korzystania z jego możliwości pozwoli na znaczne zwiększenie tempa pracy i przyjemniejsze odnajdywanie błędów.

W Sieci

- *http://www.python.org* – strona główna Pythona,
- *http://www.python.org.pl* – polski support Pythona,
- *http://msdn.microsoft.com/vstudio* – Visual Studio,
- *http://peachfuzz.sourceforge.net* – Peach Fuzzing Platform,
- *http://peachfuzz.sourceforge.net/peach.xsd.html* – schemat XML Peach,
- *http://www.activestate.com/Products/activepython* – ActivePython,
- *http://www.wireshark.org* – Wireshark.

Wireshark

Jeden z najbardziej znanych *snifferów*, analizujący i zapisujący ruch sieciowy. Współpracuje z wieloma protokołami (obecnie obsługuje ich kilkaset) oraz typami połączeń – od *Ethernetu* po *FDDI*. *Wireshark* potrafi także pokazywać nagłówki poszczególnych pakietów. Nie są mu straszne wszelkiego rodzaju zabezpieczone połączenia – *IPSec*, *Kerberos* czy *WPA*. Za pomocą tysięcy wbudowanych filtrów i dzięki przyjaznemu interfejsowi graficznemu można bez problemu przechwycić wszystkie wartościowe informacje z sieci.

Piotr Łaskawiec
Student Informatyki Stosowanej na Politechnice Krakowskiej. Założyciel i przewodniczący Koła Naukowego PK IT Security Group (*www.pkitsec.pl*). Od wielu lat związany z tematyką bezpieczeństwa komputerowego. Pasjonat języka Python. W wolnych chwilach programuje i zajmuje się publicystyką.
Kontakt z autorem: *hellsource@gmail.com*

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

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

GRZEGORZ BŁOŃSKI

Atak na reputację

Stopień trudności

Zaburzona reputacja w przypadku człowieka może spowodować zachwianie stabilności jego pozycji w miejscu pracy czy środowisku, w którym żyje. Postawmy sobie pytanie: co może się stać w przypadku zaburzenia reputacji instytucji, koncernu, firmy czy banku?

Reputacja według słowników oznacza dobrą opinię, renomę, dobrą sławę. Reputacja to także doskonałe *narzędzie* pozwalające nam dokonywać wyboru. Kierujemy się nią na przykład podczas zakupu chociażby komputera. Szukamy producenta, który cieszy się dobrą renomą, którego produkty są dobrze oceniane przez innych użytkowników. W przypadku instytucji czy firm reputacja to także bardzo ważny element zaufania ze strony petentów bądź klientów. Każda znacząca firma i instytucja posiada dziś witrynę w Internecie. Wszędzie funkcjonują systemy informatyczne mniej lub bardziej dostępne dla ludzi. O tym, że te witryny internetowe i systemy narażone są na typowe ataki, dowiadujemy się zarówno z samego Internetu, jak i z mediów. Większość ataków, o których się dowiadujemy, ma na celu zdobycie określonych informacji lub zwyczajne skompromitowanie systemu. Zastanówmy się jednak – na przykładzie banku – co dzieje się w momencie, gdy strona logowania do serwisu zostaje pomyślnie zaatakowana, wynikiem czego jest okresowy brak dostępu do kont z poziomu przeglądarki dla całej rzeszy klientów tego banku. Każdy klient chcący się dostać do swojego konta w celu wykonania pożądanych operacji nie może tego zrobić. Część klientów, obdarzona dużą cierpliwością, przeczeka te problemy – by skorzystać z serwisu bankowego w późniejszym terminie. Co zrobi ta część, która nie potrafi lub nie chce cierpliwie poczekać? Wielu z nich nie będzie się długo zastanawiać i w najbliższym czasie zmieni bank na inny, co wpłynie niekorzystnie nie tylko na reputację banku, ale także na jego kondycję finansową. Oczywiście mowa tu o grupie znacznie liczniejszej niż jedna czy kilka osób.

Kolejna grupa klientów na pytanie ze strony swoich znajomych szukających odpowiedniego banku będzie odpowiadała *ten bank nie jest taki dobry, lepiej skorzystaj z innego* – to zjawisko także ma wpływ na obniżenie reputacji banku. W taki sposób można łatwo wyciągnąć wniosek, że ataki na systemy informatyczne wykonywane w określony sposób przy wykorzystaniu typowych technik mogą w określonych sytuacjach stać się atakiem na reputację.

Podejrzewam, że celem dużej liczby ataków w dzisiejszych czasach jest właśnie obniżenie reputacji atakowanej instytucji czy firmy.

Metody

Atakujący mogą używać szeregu metod do wykonania ataku na reputację celu. W większości przypadków rozpoznane ataki charakteryzują się między innymi wykorzystaniem techniki *Distributed Denial of Service* (DDoS), której celem jest *zalanie* serwisów firmowych ogromną ilością zapytań wysyłanych przy wykorzystaniu wcześniej stworzonych sieci – botnetów – na które składają się tysiące zarażonych komputerów w Internecie. Podczas takiego ataku trudno mówić o jednoznacznym źródle, co powoduje problem w ustaleniu bezpośredniego sprawcy. Atak DDoS

Z ARTYKUŁU DOWIESZ SIĘ

co to jest atak na reputację,

jakie metody wykorzystuje się do ataku na reputację,

jakie mogą być skutki takiego ataku.

CO POWINIENEŚ WIEDZIEĆ

znać podstawowe rodzaje ataków,

umieć określać rodzaj ataku po jego znamionach.

może spowodować załamanie kondycji na przykład wspomnianego wcześniej serwisu bankowego – i w efekcie doprowadzić do obniżenia zaufania ze strony klientów banku.

Kolejną metodą ataku na reputację jest wszystkim dobrze znany i popularny *deface* strony internetowej. Mechanizmy ataku *deface*, jakiekolwiek by nie zostały tutaj przytoczone, sprowadzają się do jednego – strona internetowa zostaje zmieniona. Efekt, jaki chce osiągnąć atakujący reputację właściciela strony, zostaje uzyskany, kiedy skompromitowaną, zmienioną stronę zobaczy jak największa ilość odwiedzających. Kiedy odwiedzający stronę potencjalni nowi klienci firmy widzą, że jest ona zmieniona przez osoby trzecie, a zawarte na niej treści nie są tymi, których oczekiwali – ich zaufanie do tej firmy zostaje mocno nadszarpnięte. Reputacja takiej firmy w oczach zarówno obecnych, jak i potencjalnych nowych klientów spada – czyli cel atakującego zostaje w pełni osiągnięty.

Atakujący wykorzystują jeszcze inną technikę do ataku na reputację. Opiera się ona na publikowaniu całej gamy fałszywych informacji w wielu miejscach jednocześnie – na grupach dyskusyjnych, forach, blogach. Rozproszone w ten sposób informacje często są powielane przez niczego nie świadomych użytkowników Internetu, co powoduje błyskawiczne rozprzestrzenianie się fałszywej informacji, która traktowana jest jak niezaprzeczalny fakt.

Bez względu na metody wykorzystywane w atakach na reputację cel jest zawsze taki sam – skompromitować cel ataku, pozbawić go zaufania klientów, obniżyć jego wartość jako kontrahenta czy spowodować obniżenie wyników finansowych firmy.

Bardzo częstą techniką wykorzystywaną do ataków na banki jest *phishing*, czyli podszywanie się pod pracownika banku w celu uzyskania potrzebnych danych. Ataki *phishingowe* skierowane są najczęściej na zdobycie informacji, które pozwolą na kradzież pieniędzy z czyjegoś konta. Faktem jest, że kradzież pieniędzy z konta bankowego – kiedy już zostaje ujawniona opinii publicznej – jest informacją, która wpływa niekorzystnie na wizerunek banku i sama w sobie jest już częścią ataku na reputację. W przypadku wykorzystania *phishingu* podczas ataku na reputację banku celem jest także zdobycie informacji przydatnych do rozsyłania SPAMu, które jest często wykorzystywaną metodą osłabiania reputacji. Przesyłka (email) otrzymana od instytucji czy firmy lub banku, zawierająca na przykład reklamy w zmasowanej liczbie, może spowodować irytację, ale także zapchanie się skrzynki odbiorczej – co w efekcie może spowodować utratę zaufania do takiej instytucji. Prawdopodobnie każdy, kto otrzymał w zbyt dużej ilości reklamy z jakiejś firmy, przestał traktować ją poważnie – w szczególności, kiedy reklamowe maile spowodowały utrudnienia w korzystaniu z poczty elektronicznej.

Metody ataku dobierane są zapewne dla każdego celu indywidualnie, co powoduje trudności w skutecznym wyizolowaniu takiego właśnie ataku.

Ryzyko utraty reputacji

Potencjalne ryzyko utraty reputacji jest czynnikiem mający ogromny wpływ na proces zwiększania bezpieczeństwa serwisu banku czy firmowej strony WWW.

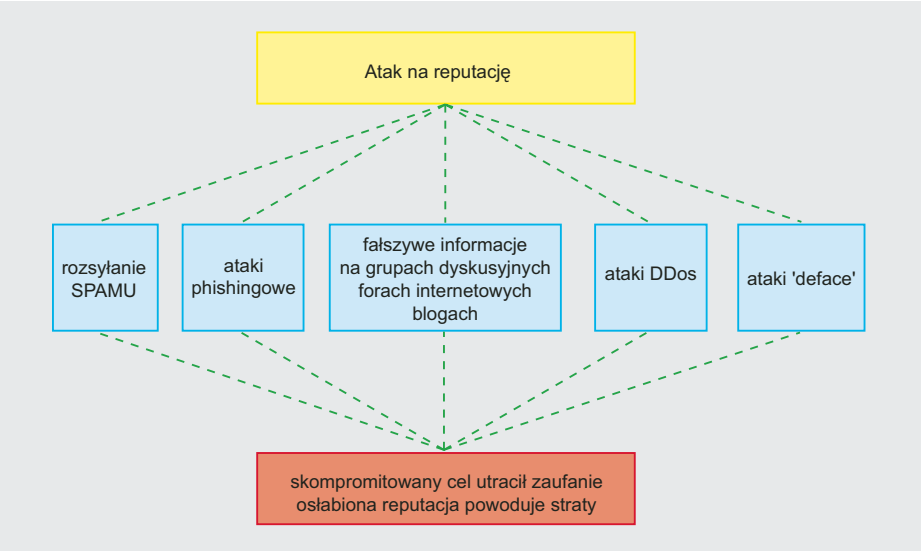
Jakkolwiek utrata dobrego imienia może mieć miejsce w następstwie na przykład błędów w funkcjonowaniu systemu, należy zawsze brać pod uwagę możliwość ingerencji osób trzecich i przed takimi sytuacjami się zabezpieczać.

Ryzykiem utraty reputacji obarczone są także różne instytucje. Przecież nietrudno wyobrazić sobie spadek poparcia dla określonej opcji politycznej w momencie, gdy prócz informacji (najczęściej nieprawdziwych) umieszczanych masowo na forach internetowych, grupach dyskusyjnych czy blogach, atakowana jest strona internetowa ugrupowania, a także prywatne witryny jej członków. Społeczeństwo, czytając nieprawdziwe informacje i widząc zaatakowane strony, może przestać ufać takiej opcji politycznej.

Każda instytucja, organizacja, koncern, firma, ale także i osoba prywatna jest w pewnym stopniu narażona na atak na reputację. Ryzyko takiego ataku wzrasta w momencie wykorzystywania technologii przekazu elektronicznego dostępnych w Internecie. Witryna internetowa niedostatecznie zabezpieczona, posiadająca luki, może stać się celem skutecznego ataku, w efekcie którego jej właściciel może mieć spory problem z ponownym uzyskaniem zaufania – naprawą utraconej reputacji.

W celu zapobiegania utracie reputacji zainteresowane podmioty powinny właściwie kształtować politykę współpracy z dostawcami sprzętu i oprogramowania wykorzystywanego w elektronicznej transmisji danych. W ramach tej polityki należy uwzględniać ograniczony dostęp osób trzecich do przetwarzanych danych, ale także właściwe informowanie (czy wręcz szkolenie) współpracujących instytucji i klientów na okoliczność wystąpienia jakichkolwiek przesłanek dających choćby cień podejrzenia ataku na reputację.

W przypadku instytucji takich jak banki, dbałość o niedopuszczenie do ataku na reputację powinna być



Rysunek 1. *Przykładowy schemat ataku na reputację*

jednym z priorytetów w zakresie zapewnienia bezpieczeństwa. Banki są instytucjami zaufania publicznego, którym społeczeństwo powierza oprócz swoich pieniędzy równie cenne dane osobowe, zawarte w umowach kredytowych, leasingowych i innych. Utrata zaufania przez bank może spowodować objawy paniki wśród klientów, czego efektem mogą być masowe rezygnacje z usług skompromitowanego banku, a także rozpowszechnianie informacji o jego *złej sławie*.

Reputacja – składnik cyfrowej tożsamości

Reputacja jest składnikiem cyfrowej tożsamości, którą posiada każda osoba, instytucja czy firma pojawiająca się w elektronicznych kanałach przekazu informacji. W mapie cyfrowej tożsamości, stworzonej przez francuskiego naukowca Freda Cavazzę, zajmuje ona eksponowane miejsce, co pozwala wysuwać wniosek że dbałość o reputację jest niezwykle istotna.

Z treści korespondencji, którą prowadziłem z Fredem wynika, że są sposoby na uchronienie reputacji przed atakami – przynajmniej jeśli chodzi o reputację pojedynczych osób. W przypadku firm oraz instytucji jest trochę trudniej, jednak także można skutecznie bronić swojej reputacji.

Fred twierdzi, że bardzo dobrą (a przede wszystkim skuteczną) techniką jest robienie *hałasu* (ang. *noise*) wokół swojej tożsamości cyfrowej. Większość z nas, zakładając przeróżne konta w Sieci używa tych samych loginów i haseł, często dodając opisy swojej osoby lub adresy i telefony – co pozwala na łatwe znalezienie nas w Internecie i próby ataku na naszą tożsamość cyfrową. Aby się przed tym uchronić, można wykorzystać wspomnianą technikę poprzez zakładanie różniących się od siebie publicznych profili w serwisach internetowych. Spowoduje to, że mechanizmy wyszukiwarek nie będą w stanie jednoznacznie nas zidentyfikować.

Takie działanie uchroni nas przed szybkim wykryciem przez potencjalnego atakującego i spowoduje, że atak na cyfrową reputację będzie utrudniony – co w efekcie może zniechęcić napastnika do dalszych prób.

W odniesieniu do instytucji czy firm taka technika nie jest możliwa do wykorzystania ze względu na potrzebę właściwej identyfikacji danego podmiotu, która jest niezbędna do jego funkcjonowania w Sieci bez problemów związanych z odnalezieniem przez potencjalnych klientów, kontrahentów itp.

Dobra reputacja jest jednym z głównych celów Public Relations, więc nie można mówić o samej reputacji bez odniesienia się do PR jako instrumentu komunikacji marketingowej. W działaniach PR obok reputacji duży nacisk kładzie się na wizerunek i zaufanie, a to są rzeczy bardzo mocno związane z reputacją.

Oczywistym staje się więc współdziałanie komórek PR oraz IT w firmie w celu ustalenia, wdrożenia oraz monitorowania określonych polityk bezpieczeństwa wraz z działaniami mającymi na celu minimalizację możliwości ataku na reputację. W przypadku firm skuteczną metodą obrony przed atakami na reputację jest perfekcyjna dbałość o szczegóły dotyczące właściwego przygotowania wszystkich danych dostępnych dla osób niezwiązanych z firmą. Dane, które mogą stać się przyczyną utraty reputacji firmy, muszą być bardzo dobrze chronione poprzez wdrażanie skutecznej i szczelnej polityki bezpieczeństwa, systemu szyfrowania danych oraz bardzo rygorystycznego podejścia do nadawania uprawnień wglądu do danych poszczególnym użytkownikom. Administrator serwisu internetowego firmy

Rysunek 2. *Mapa cyfrowej tożsamości Freda Cavazzy*

W Sieci

- *http://bankwide.com*,
- *http://boston-review.com*,
- *http://www.insecuremag.com*,
- *http://www.radicallytransparent.com/online-reputation-management-book-contents*,
- *http://www.oxfordmediaworks.com/blog/how-to-protect-your-reputation-online*,
- *http://www.openid.pl*,
- *http://www.identity20.pl*.

powinien przy każdej aktualizacji witryny sprawdzać dokładnie, czy aktualizacja nie powoduje wycieku danych. Bardzo istotnym elementem minimalizującym ryzyko ataku na reputację jest ciągłe monitorowanie ruchu w systemach informatycznych, w szczególności tych wystawionych na bezpośrednie działanie w Internecie. Kolejnym bardzo ważnym elementem pomagającym w unikaniu ataków na reputację jest aktualizacja oprogramowania do wersji nieposiadających żadnych luk i podatności na przeróżne ataki. Życie pokazuje, że niewiele firm monitoruje swoje serwisy w dostatecznym stopniu, a jeszcze mniej dokonuje aktualizacji oprogramowania.

Przykładem firmy, której reputacja została zaatakowana ze skutkiem w postaci utraty wiarygodności, jest *CastleCops* .Działalność *CastleCops* polega na walce z przestępczością informatyczną. Zorganizowana grupa hakerów przy wykorzystaniu tysięcy komputerów połączonych w botnet zaatakowała stronę CastleCops, próbując uniemożliwić korzystanie z serwisu. Był to typowy atak *Distributed Denial of Service*, który miał zakłócać próby logowania do serwisu jego użytkownikom. Gdy okazało się, że te próby ataku niestety zawodzą, hakerzy postanowili wykonać atak poprzez *bogus donations* czyli fałszywe darowizny. Niektórym użytkownikom serwisu płatniczego *PayPal* skradziono (wykorzystując *phishing*) dane dostępowe do konta i dzięki temu przelano pieniądze na konto CastleCops. Przelewane sumy były przeróżnej wielkości – od kilku dolarów do nawet 2800 dolarów. Klienci *PayPal* zgłaszali kradzieże pieniędzy z kont, a gdy podczas wyjaśniania każdej kradzieży okazywało się, że pieniądze zostały w zwykły sposób przelane na konto *CastleCops* – zaczęto oskarżać właśnie tę firmę o kradzież. *CastleCops* broniło się, próbując wykorzystywać FBI do wykrycia prawdziwych sprawców, lecz fakt spadku zaufania do firmy już miał miejsce. Wpływy z darowizn na rzecz *CastleCops* znacznie spadły.

Podsumowanie

Mam nadzieję, że odpowiedź na pytanie postawione na początku tego artykułu: *co może się stać w przypadku zaburzenia reputacji instytucji, koncernu, firmy czy banku?* jest teraz dużo łatwiejsza. Choć ataki na reputację nie są szeroko i dokładnie opisywane, nie oznacza to, że nie mają one miejsca na co dzień. Ataki mające na celu osłabienie reputacji są bardzo złożonymi przedsięwzięciami, które są często nie rozpoznawane jako takie. Na przykład atak *deface* na stronę firmy czy banku często traktowany jest jako dzieło *script kiddies*. Z pozoru niewinny *deface* nie jest wiązany z wcześniejszymi atakami DDoS na serwis internetowy. Co gorsza, informacje o SPAMie docierającym do klientów już zupełnie nie są traktowane jako część składowa ataku na reputację.

Źródłem ataku na reputację może być w zasadzie każdy rodzaj napastnika – począwszy od przypadkowego nastolatka, przez żądnego sławy hakera, nieodpowiednio potraktowanego klienta, źle opłacanego i traktowanego pracownika, a na konkurencyjnej firmie skończywszy. Zapewne wymienieni nie kończą listy potencjalnych źródeł ataku. Jakkolwiek przyczyn skłaniających do ataku na reputację może być naprawdę dużo, do najczęstszych należy zapewne chęć zdobycia sławy przez *hakera*. W mniejszym stopniu ataki takie są powodowane chęcią zdeklasowania konkurencyjnej firmy i obniżenia jej dochodów czy obnażeniem niskiego poziomu profesjonalizmu instytucji rządowych w podejściu do kwestii obecności w przestrzeni Internetu.

W dzisiejszych czasach zespoły specjalistów zajmujących się przeciwdziałaniem wszelkim atakom skierowanym w przeróżne formy przekazu informacji elektronicznej powinny analizować całe spektrum incydentów w celu dokładnego zrozumienia każdego przypadku i zastosowania w przyszłości skutecznych metod zapobiegania i obrony przed atakami konkretnego typu.

Grzegorz Błoński
Z wykształcenia jest informatykiem, certyfikowanym specjalistą IBM. Pracuje w dużej firmie o zasięgu światowym. Zajmuje się administracją oraz bezpieczeństwem sieciowym. Jest członkiem organizacji International Information Systems Forensics Association (IISFA), ISACA oraz Internet Society.
Kontakt z autorem: *mancymonek@mancymonek.pl*

MICHAŁ „GYNVAEL COLDWIND” SKŁADNIKIEWICZ

Format GIF okiem hakera

Stopień trudności

Pliki graficzne są dziś szeroko rozpowszechnionym nośnikiem informacji, spotyka się je praktycznie na każdym komputerze. Dobry programista powinien wiedzieć, jak wyglądają nagłówki poszczególnych formatów plików graficznych, a także – jak przechowywany jest sam obraz.

Z ARTYKUŁU DOWIESZ SIĘ

jak zbudowany jest plik GIF,

na co uważać podczas implementowania obsługi formatu GIF,

gdzie szukać błędów w aplikacjach korzystających z GIF,

gdzie ukryć, lub szukać ukrytych danych, w plikach GIF.

CO POWINIENEŚ WIEDZIEĆ

mieć ogólne pojęcie na temat plików binarnych,

mieć ogólne pojęcie na temat bitmap,

mieć ogólne pojęcie na temat kompresji.

Niniejszy artykuł, drugi z serii *Format graficzny okiem hakera* (pierwszy, *Format BMP okiem hakera*, został opublikowany w Hakin9 3/2008), ma na celu zapoznać Czytelnika z opracowanym przez CompuServe i przedstawionym w roku 1987 formatem GIF (ang. *Graphics Interchange Format*, Formatem Wymiany Grafiki), wskazać w nim miejsca, które można wykorzystać do przemycenia ukrytych danych, a także takie, w których programista może popełnić błąd podczas implementacji i wreszcie – zapoznać Czytelnika z samym formatem. Przykłady będą, w miarę możliwości, zilustrowane pewnymi bugami w istniejącym oprogramowaniu, znalezionymi przez autora oraz inne osoby.

Wstęp do GIF

Format GIF oferuje możliwość przechowania jednego lub więcej obrazów o maksymalnie 8-bitowej głębi kolorów (czyli 256 kolorów, chociaż to ograniczenie jest do ominięcia – jest to omówione w dalszej części artykułu). Budowa formatu GIF jest dużo bardziej złożona niż omówionego dwa miesiące temu formatu BMP. Dodatkowo dane obrazu są bezstratnie kompresowane (chociaż przy wykorzystaniu pewnego triku mogą też być nieskompresowane – będzie o nim później), a sam format umożliwia przechowywanie animacji, dzięki czemu stał się powszechnie używany na stronach WWW.

Istnieją dwie wersje formatu GIF, starsza – 87a oraz nowsza – 89a. Ten artykuł dotyczy wersji nowszej. Tyle tytułem wstępu, czas na właściwą część artykułu.

Kompresja LZW

Do zakodowania obrazu w formacie GIF wykorzystana jest kompresja LZW (od nazwisk pomysłodawców, Lempel-Ziv-Welch), a konkretniej jej wariant ze zmienną długością kodu. Jednak przed przejściem do tej wersji warto zapoznać się z podstawową wersją algorytmu.

Algorytm LZW, będący modyfikacją algorytmu LZ78, jest słownikowym algorytmem bezstratnej kompresji, w której słownik jest dynamicznie generowany podczas procesu kompresji lub dekompresji danych. Na Listingach 1. oraz 2. przedstawiony jest pseudokod – odpowiednio kompresji oraz dekompresji LZW. Ciągiem wejściowym może być na przykład sekwencja (strumień) 8-bitowych kodów (po prostu bajtów), natomiast ciąg wyjściowy powinien być sekwencją kodów o stałej, wybranej przez programistę, ilości bitów. Wielkość kodu wyjściowego nie może być mniejsza niż długość pojedynczego elementu wejściowego, a w zasadzie, jeżeli ma być mowa o jakiejkolwiek kompresji, powinna być większa. Przykładowo załóżmy, że jeden element wejściowy ma wielkość 8 bitów, zaś jeden element kodu wyjściowego niech ma

12 bitów (a przynajmniej 9 bitów). W takim wypadku element wejściowy może przyjąć jedną z 256 różnych wartości (2^8), natomiast element wyjściowy 4096 różnych wartości (2^12). Zarówno w przypadku kompresji, jak i dekompresji pierwszym krokiem jest stworzenie słownika ciągów, o wielkości równej ilości możliwych wartości elementu wyjściowego – czyli w przypadku naszego przykładu, o wielkości 4096 elementów słownikowych. Zakładamy, że słownik na początku jest pusty, następnie wpisujemy do niego wszystkie możliwe kombinacje, jakie może wyrazić kod wejściowy (czyli 256 kombinacji, ważne jest zachowanie kolejności). W pseudokodzie czynność tę można wyrazić w następujący sposób:

```
Dla I przyjmującego wartości od 0 do
                    255...
  Słownik[I] = I
```

W ten sposób pierwsze 256 elementów (czyli elementy od 0 do 255) słownika zostanie wypełnionych. Kolejnym wolnym elementem słownika będzie więc element 256. W trakcie kompresji kolejnym elementom słownika przypisywane będą kolejne ciągi, których wcześniej nie było w słowniku (warto dokładnie przeanalizować pseudokod kompresji, jest on bardzo prosty).

Dekompresja jest odrobinę (ale tylko odrobinę) bardziej skomplikowana, ponieważ okazuje się, że istnieje specjalny przypadek ciągu kompresowanego, powodujący generowanie na wyjściu kodu, który przy dekompresji jeszcze nie trafił do słownika. Przykładowo, dekompresor ma wypełniony słownik do elementu 270 włącznie, a nagle dostaje kod 271. Dzieje się tak w wypadku, gdy wejściowy ciąg zawiera sekwencję znak-ciąg-znak-ciąg-znak (gdzie poszczególne znaki są identyczne, poszczególne ciągi również), czyli na przykład ABBABBA. Na szczęście brakujący kod łatwo jest wtedy odtworzyć – jest to po prostu ostatni wypisany ciąg z dopisaną swoją pierwszą literą na końcu (czyli, jeśli ostatnio wypisany był ABB, brakującym ciągiem jest ABBA).

Dokładne działanie i przebieg kompresji oraz dekompresji nie jest tematem tego artykułu, pozostaje więc w kwestii Czytelnika przeanalizowanie pseudokodu oraz ewentualne doczytanie zasady działania LZW. Warto napisać przykładowy kod kompresujący i dekompresujący, np. w Pythonie lub Perlu (z uwagi na prostotę implementacji słownika w w/w językach).

GIF a kompresja LZW

W formacie GIF użyto LZW z dwoma modyfikacjami. Po pierwsze, do słownika (po jego zainicjowaniu) dodano dwie specjalne wartości: kod czyszczenia słownika (w przypadku 8-bitowego wejścia ma on kod 256) oraz kod końca danych (kod 257, musi on wystąpić po danych). Pierwszym wolnym kodem jest więc 258. Druga modyfikacja wywodzi się ze słusznej obserwacji, iż 12 bitów bywa nadmiarowe, szczególnie w wypadku, gdy w słowniku jest dużo mniej kodów, i kiedy można je zapisać za pomocą mniejszej liczby bitów. Modyfikacja zakłada użycie jak najmniejszej liczby bitów do zapisania kodu wyjściowego na początku oraz ewentualny wzrost liczby bitów używanych wraz z rozrostem słownika (czyli jeśli w słowniku jest mniej niż 512 elementów, to kody będą 9-bitowe, jeśli mniej niż 1024, ale więcej niż 512 – kody będą 10-bitowe etc). Początkowa wielkość kodu jest o jeden bit większa od wielkości kodu wejściowego (czyli dla 8-bitowego wejścia kod wyjściowy będzie miał najpierw 9 bitów) – wynika to z konieczności stworzenia możliwości zapisu kodu czyszczenia oraz kodu końca danych. W przypadku GIF maksymalna przewidziana wielkość kodu to 12 bitów. W momencie, gdy dekompresor napotka kod czyszczenia słownika, wielkość kodu redukuje się do swej początkowej wartości, a cały słownik powraca do stanu z początku dekompresji.

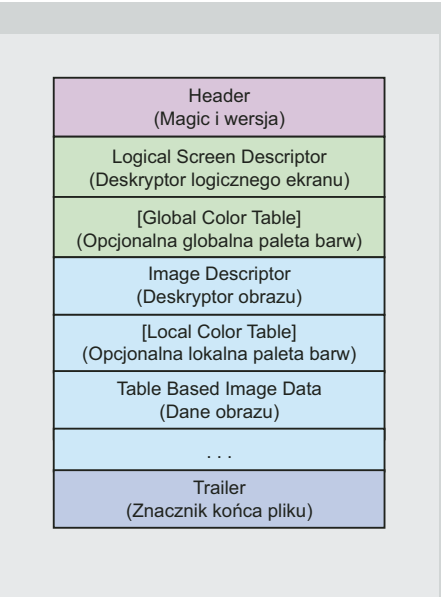
Rozważmy przykład kompresji 8-bitowego wejścia. Niech wejściowe dane (zapisane w oktetach heksadecymalnie) wyglądają następująco: 00 01 02 00 01. Początkowa wielkość kodu wyjściowego to 9 bitów, a pierwszym wolnym elementem będzie 258. Na początku odczytany zostanie znak 00, który znajduje już się w słowniku

Tabela 1. *Struktura GIF98aHeader*

Typ i nazwa pola	Opis
BYTE Signature[3]	Sygnatura (tzw. *magic*), zawsze „GIF”
BYTE Version[3]	Oznaczenie wersji, „87a” lub „89a”

Tabela 2. *Struktura GIF89aLSD*

Typ i nazwa pola	Opis
WORD Width	Szerokość ekranu logicznego
WORD Height	Wysokość ekranu logicznego
BYTE SizeOfGlobalColorTable:3	Wielkość globalnej palety barw
BYTE SortFlag:1	Znacznik posortowanej palety barw
BYTE ColorResolution:3	Głębia kolorów
BYTE GlobalColorTableFlag:1	Znacznik występowania globalnej palety barw
BYTE BackgroundColorIndex	Numer koloru tła
BYTE PixelAspectRatio	Proporcja rozmiarów piksela



Rysunek 1. *Uproszczona budowa pliku GIF*

(pozycja 00). Do niego zostanie dołączony drugi odczytany znak, czyli 01, jednak ciągu 00 01 nie ma w słowniku, w związku z czym na wyjście zostanie wypisany 9-bitowy kod 0, a ciąg 00 01 zostanie dodany na pozycję 258 do słownika. Znak 01 zostaje zapamiętany, a znak 02 zostaje do niego doczytany. Ponieważ, tak jak poprzednio, nowo powstały ciąg 01 02 nie znajduje się w słowniku, na wyjście zostaje wypisany kod 1 (kod znaku 01), a ciąg 01 02 zostaje dodany do słownika na pierwszą wolną pozycję, czyli 259. I znów, znak 02 zostaje zapamiętany i zostaje do niego doczytany kolejny znak wejścia, a więc 00. Analogicznie, jak wcześniej, na wyjściu znajdzie się kod 2 (kod znaku 02), a ciąg 02 00 będzie dodany do słownika na pozycję 260. Znak 00 zostaje zapamiętany, doczytany do niego zostaje znak 01. Ciąg utworzony przez te znaki, czyli 00 01, jest już w słowniku. Ponieważ jest to koniec ciągu, to kod ciągu 00 01 (czyli 258) zostaje wypisany na wyjście. I tak oto ciąg 00 01 02 00 01 (czyli 40 bitów) został skompresowany do ciągu 9-bitowych kodów 0 1 2 258 (czyli 36 bitów). Aby w pełni zachować zgodność ze standardem, należy wyemitować również kod końca danych, czyli 257. Natomiast należy wiedzieć, iż nie wszystkie dekompresory tego wymagają.

Dekompresja odbędzie się w następujący sposób: najpierw wczytany zostanie kod 0. Ze słownika pobrany zostanie ciąg odpowiadający temu kodowi, czyli 00. Ciąg ten zostanie wypisany na wyjście oraz zapamiętany. Następny pobrany kod to 1, w słowniku odpowiada mu jednoelementowy ciąg 01. Do słownika, na pozycję 258, zostaje dodany nowy ciąg, powstały z połączenia zapamiętanego ciągu 00 z pierwszym elementem nowego ciągu, czyli 01 (a więc razem 00 01).

Listing 1. *Pseudokod kompresji algorytmem LZW*

```
Wypełnij pierwszą część słownika
Niech ciąg pobranych elementów P jest
                        pusty
Dopóki ciąg wejściowy jest niepusty...
  Pobierz nowy element N
  Jeżeli ciąg P+N jest w słowniku...
    P = P+N
  W przeciwnym wypadku...
    Dodaj P+N do słownika
    Wypisz kod ciągu P
    P = N
Wypisz kod P
```

Nowy ciąg 01 zostaje wypisany na wyjście oraz zapamiętany. Z wejścia odczytany zostaje następny kod, czyli 2, w słowniku odpowiadający ciągowi 02. Analogicznie jak poprzednio, do słownika na pozycję 259 dodany zostanie wyraz 01 02, na wyjście zostanie wypisany kod 02 i zostanie on również zapamiętany. Ostatnim kodem na wejściu jest 258, odpowiadający w słowniku ciągowi 00 01. Do słownika na pozycję 260 trafia ciąg 02 00, a ciąg 00 01 zostaje wypisany na wyjście oraz zapamiętany. Podsumowując, na wyjście trafi ciąg 00 01 02 00 01, odpowiadający ciągowi wejściowemu przy kompresji. Należy również zauważyć, iż zarówno słownik utworzony przy kompresji, jak i słownik utworzony przy dekompresji są identyczne.

Ponieważ kompresja LZW użyta w GIF została objęta patentem (patent US 4558302, nadany w grudniu 1985, ale Unisys upomniał się o opłaty licencyjne dopiero w grudniu roku 1994; patent wygasł w roku 2005), środowisko, chcąc nadal używać formatu GIF, opracowało metodę kodowania kompatybilnego z dekompresorem LZW (sam dekompresor nie był objęty patentem), natomiast bez użycia kompresji. Piątego grudnia 1996 roku doktor Tom Lane na grupie dyskusyjnej *comp.graphics.misc* zaproponował, by w kodowaniu obrazu używać jedynie podstawowego słownika zbudowanego z pojedynczych znaków (czyli np. tylko pierwsze 256 elementów słownika w przypadku 8 bitów). Oczywiście w tym przypadku nie można mówić o kompresji, ale raczej o powiększeniu rozmiaru danych wyjściowych, ponieważ będą one przynajmniej o jeden bit większe na każdym elemencie. Do tego dochodzą jeszcze kody czyszczenia słownika, które mają zapobiec zwiększeniu wielkości wyjściowego kodu powyżej 9 bitów. Taki był też zamysł autora – jeżeli nie ma mowy o kompresji, to metoda nie narusza patentu, ale mimo wszystko tworzy poprawnego GIF'a, który jest poprawnie odczytywany przez dekompresor LZW.

Wracając do przykładowego ciągu 00 01 02 00 01, mógłby on zostać w tym wypadku zapisany po prostu jako 0 1 2 0 1 (gdzie każdy kod ma oczywiście 9 bitów).

W przypadku, gdy liczba wejściowych danych spowodowałaby powiększenie się słownika, można – jak proponował dr Lane – wyemitować kod czyszczący. Można również zwiększyć kod wyjściowy o kolejny bit, cały czas jednak używając jednoelementowych ciągów ze słownika. W takim wypadku nie jest wymagane emitowanie kodu czyszczącego, jednak wyjściowy ciąg będzie dłuższy niż w przypadku użycia kodów czyszczących.

LZW a steganografia

Jak widać na przykładzie propozycji dr Lane'a, dekompresor LZW potrafi odkodować nie tylko dane zakodowane ściśle według zaleceń algorytmu kompresji LZW, ale również takie, które tylko pozorują kompresję. Ten właśnie aspekt LZW pozwala na ukrycie dodatkowych informacji w ciągu skompresowanych danych, bez wpływu na wygląd samego obrazu.

Pierwszym sposobem jest nadmiarowe używanie kodów czyszczących. Załóżmy, iż mamy do ukrycia ciąg bitów. Jedynkę można zakodować jako umieszczenie kodu czyszczącego na kolejnej pozycji, a zero jako brak kodu czyszczącego

Listing 2. *Pseudokod dekompresji algorytmem LZW*

```
Wypełnij pierwszą część słownika
Niech poprzedni ciąg P będzie pusty
Dopóki ciąg wejściowy jest niepusty...
  Pobierz nowy kod K
  Jeśli K jest w słowniku...
    Niech C będzie ciągiem pobranym ze słownika z indeksu K
  W przeciwnym wypadku...
    Niech C będzie ciągiem P + P[0], gdzie P[0] to pierwszy znak ciągu P
  Wypisz ciąg C
  Dopisz do słownika w pierwsze wolne miejsce ciąg P+C[0]
  P = C
```

na tej pozycji – czyli inny, normalny, element ciągu. Załóżmy że standardowe, skompresowane dane (dla ułatwienia sprawy zakodowane metodą dr Lane'a, chociaż nie jest to konieczne) wyglądają tak: 1 2 3 1 2 3 1 2. Natomiast ciąg bitów, który chcemy ukryć, to 01001011. W takim wypadku wyjściowy kod mógłby wyglądać następująco (oznaczę kod czyszczący jako C): 1 C 2 3 C 1 C C 2 3 1 2. Oczywiście dane ponownie *urosły*, ale zawierają teraz dodatkową informację, a po dekompresji przestawią identyczny ciąg wyjściowy, jak wcześniej.

Drugi sposób opiera się o możliwość takiego zapisu kodu kompresowanego, by dekompresor przy dekompresji stworzył dwa identyczne wpisy w słowniku. Rozważmy następujący ciąg: 1 1 1. Dekompresor najpierw zapamięta ciąg 01, następnie umieści w słowniku ciąg 01 01 (np. na pozycji 258), po czym zapamięta 01. Przy odczycie następnej 1 ponownie doda do słownika ciąg 01 01 na kolejną pozycję (tym razem 259). I tak oto dekompresor otrzymał w słowniku dwa kody reprezentujące ciąg 01 01. To, który kod zostanie użyty do zapisu ciągu 01 01, może zależeć od poszczególnych bitów ukrywanych danych. Należy zauważyć, iż można sprowokować dekompresor do umieszczenia np. 256 identycznych wpisów w słowniku (w końcu słownik ma 4096 elementów – miejsca aż nadto). W takim wypadku wybór użytego kodu może służyć do zapisu 8 bitów informacji. Należy pamiętać, iż maksymalna ilość informacji zapisanych w skompresowanym ciągu zależeć będzie również od zawartości (wyglądu) kompresowanej bitmapy (prawdziwie losowa bitmapa, o bardzo wysokiej entropii, nie sprawdzi się przy tej metodzie).

Trzeci sposób związany jest ze specjalnym przypadkiem podczas dekompresji, kiedy wczytany kod nie figuruje jeszcze w słowniku. W takim wypadku dekompresor wypisuje zapamiętany ciąg z dodanym pierwszym wyrazem tego ciągu oraz tenże wynikowy ciąg zapisuje do słownika na pierwszej wolnej pozycji. Należy jednak zauważyć, iż kod, który się pojawi w tym wypadku nie musi być kolejnym kodem – ważne, żeby go nie było w słowniku. Można więc wartość takich kodów uzależnić od ukrywanych danych. Oczywiście, nie trzeba uzależniać tego sposobu od występowania sekwencji znak-ciąg-znak-ciąg-znak w bitmapie. Równie dobrze można kompresor zmusić do wypisania nieistniejącego kodu oraz do ominięcia (stworzenia, ale nie używania) jednego elementu w słowniku – wtedy wynik będzie taki sam.

Struktura pliku

Oprócz kompresji LZW format GIF posiada rozbudowaną (w stosunku np. do BMP czy TGA) strukturę opisującą zarówno same obrazy (jeden plik GIF może zawierać wiele obrazów), jak i dodatkowe elementy – takie, jak rozszerzenia aplikacji czy komentarze do obrazów. Plik GIF (patrz Rysunek 1.) zawierający jeden obraz składa się przynajmniej z nagłówka (ang. *Header*), deskryptora logicznego ekranu (ang. *Logical Screen Descriptor*, w skrócie LSD), globalnej lub lokalnej palety barw (ang. *Global / Local Color Table*, w skrócie GCT lub LCT), deskryptora obrazu (ang. *Image Descriptor*), skompresowanych i podzielonych na bloki danych (ang. *Table Based Image Data*, patrz Rysunek 2.) oraz znacznika końca obrazów (ang. *Trailer*). GIF zawierający więcej obrazów zawiera po prostu zwielokrotniony deskryptor obrazu oraz podzielone na bloki dane. Oprócz wyżej wymienionych struktur format GIF może zawierać również inne bloki – takie, jak rozszerzenie aplikacji (ang. *Application Extension*), rozszerzenie komentarzy (ang. *Comment Extension*), rozszerzenie sterowania grafiką (ang. *Graphic Control Extension*) czy wreszcie rozszerzenie zwykłego tekstu (ang. *Plain Text Extension*). Te bloki są jednak opcjonalne i nie będą opisane w artykule. Zachęcam jednak Czytelnika do zapoznania się z wyżej wymienionymi blokami – są one bardzo dobrze opisane w standardzie GIF89a.

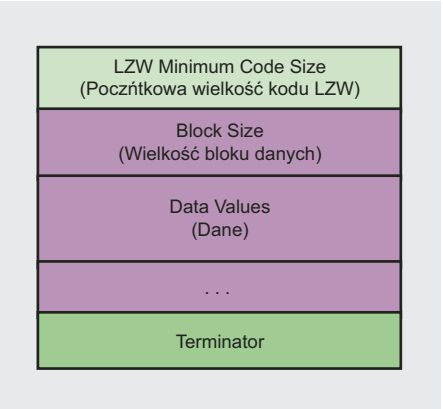
Dalsza część artykułu poświęcona jest opisowi poszczególnych nagłówków, ze wskazaniem miejsc, w których programista może popełnić błąd oraz miejsc, w których można ukryć dodatkowe dane.

Header

Na samym początku pliku znajduje się nagłówek (patrz Tabela 1.) zawierający sygnaturę pliku GIF (3 bajty, których reprezentacja ASCII to po prostu *GIF*) oraz użytą wersję standardu (również 3 bajty, dostępne są dwie wersje: *87a* oraz nowsza, *89a*). Ten nagłówek nie daje za dużego pola manewru, jednak

Tabela 3. *Struktura GIF98aID*

Typ i nazwa pola	Opis
BYTE Separator	Zawsze 0x2C
WORD Left	Pozycja X na logicznym ekranie
WORD Top	Pozycja Y na logicznym ekranie
WORD Width	Szerokość obrazu
WORD Height	Wysokość obrazu
BYTE SizeOfLocalColorTable:3	Wielkość lokalnej palety barw
BYTE Reserved:2	Pole nieużywane
BYTE SortFlag:1	Znacznik posortowanej palety barw
BYTE InterlaceFlag:1	Znacznik przeplotu
BYTE LocalColorTableFlag:1	Znacznik występowania lokalnej palety barw



Rysunek 2. *Budowa Table Based Image Data*

okazuje się, że nie wszystkie programy zwracają uwagę na wersję. Przykładem może być Apple Safari, które w ogóle nie przetwarza pola wersji – można więc użyć go do przechowania 3 bajtów informacji. Należy pamiętać, że ten GIF będzie działał wtedy jedynie pod Safari – jest to więc jednocześnie skuteczna metoda ograniczenia wyświetlania GIF'a tylko do tej przeglądarki.

Logical Screen Descriptor

Zaraz po nagłówku następuje struktura LSD, opisująca logiczny ekran. Logiczny ekran to po prostu przestrzeń, w którą zostaną wrysowane obrazy. Przestrzeń ta powinna być na tyle duża, by pomieścić każdy obraz zawarty w danym pliku GIF. Może być oczywiście również większa. Format GIF umożliwia wykorzystanie kilku (lub nawet wszystkich) mniejszych, osobnych obrazów (zawartych w tym samym pliku) do stworzenia jednej dużej grafiki. W takim wypadku poszczególne obrazy umieszczane są w różnych pozycjach na ekranie logicznym (patrz Rysunek 3.). Niestety, nie każdy program obsługujący GIF obsługuje jednocześnie poprawnie wiele obrazów na jednym ekranie logicznym (przykładowo IrfanView traktuje pliki GIF z wieloma obrazami jak animacje).

Mówiąc o wielu obrazach, warto wspomnieć o technice zapisu tworzenia 24-bitowych GIF'ów, opracowanej przez Andreasa Kleinerta (jak pisałem we wstępie, maksymalną wielkością palety kolorów w GIF'ie jest 256). Metoda opiera się o fakt, iż każdy obraz może mieć swoją własną, lokalną paletę. Całość polega na podzieleniu oryginalnego obrazu na fragmenty po 256 pikseli oraz zapisanie ich jako oddzielne obrazy, odpowiednio umieszczone na logicznym ekranie. Każdy obraz ma swoją lokalną paletę barw, która zawiera jedynie barwy potrzebne do odtworzenia 256 pikseli, które przedstawia obraz (Łatwo się domyślić, iż w tym wypadku również nie można mówić o kompresji – wyjściowy GIF będzie dużo większy choćby od pliku BMP, który zawierałby tę samą grafikę).

Nasuwające się od razu pytanie brzmi *a co, jeśli obraz jest większy od logicznego ekranu?*, oraz *a co, jeśli obraz zostanie umieszczony poza logicznym ekranem*. Prawidłową reakcją aplikacji powinno być przycięcie obrazu do logicznego ekranu. W przypadku niektórych aplikacji może dojść do przepełnienia bufora (w takim wypadku możliwość wykonania kodu jest wysoce prawdopodobna), jednak z uwagi na oczywistość tej możliwości bardzo niewiele aplikacji zawiera taki błąd.

Bardzo ciekawe zachowanie w przypadku, gdy obraz jest umieszczony poza ekranem logicznym, i jest od niego większy, wykazuje przeglądarka Opera. Rysunek 4. przedstawia wyświetloną stronę, której kod wygląda następująco:

```
<img border="1" src="test.gif"/>
```

Jak widać na rysunku, ramka została wyrysowana w miejscu zupełnie innym niż sam obraz. Dodatkowo Opera zachowuje się tak, jak gdyby obrazek znajdował się w ramce, natomiast nie było go w miejscu, gdzie faktycznie jest (zwróć uwagę na menu kontekstowe). Zachowanie takie jest niegroźne, ale równocześnie wysoce niestandardowe – a zatem interesujące.

Tabela 2 pokazuje budowę struktury LSD. Poza polami o oczywistym przeznaczeniu (*Width*, *Height*, *BackgroundColorIndex*) jest w niej kilka pól wymagających dokładniejszego opisu.

Pierwszym z nich jest flaga *GlobalColorTableFlag*. Flaga ta ustawiona jest jedynie, gdy GIF zawiera globalną paletę kolorów (niektóre źródła twierdzą, że 99.5% GIF'ów spełnia ten warunek). Globalna paleta kolorów jest opcjonalna – równie dobrze obrazy mogą wykorzystywać lokalną paletę barw. W przypadku, gdy flaga jest ustawiona, pole *SizeOfGlobalColorTable* zawiera wielkość palety kolorów. Wielkość (w sensie ilości elementów) musi zostać wyliczona z następującego równania:

```
IlośćElementów =
(SizeOfGlobalColorTable + 1)^2
```

Stąd właśnie bierze się ograniczenie ilości kolorów do 256 (*SizeOfGlobalColorTable* może przyjąć co najwyżej wartość 7, czyli z równania wyjdzie liczba 256).

W przypadku, gdy GIF nie zawiera GCT, pole *SizeOfGlobalColorTable* może zostać użyte do przechowania dowolnych danych.

Pole *ColorResolution* zawiera informację o głębi kolorów. Aby wyliczyć ilość bitów przypadających na piksel, należy dodać do wartości pola 1. Należy zauważyć, iż możliwy jest przypadek, w którym paleta barw jest większa niż głębia kolorów – w takim wypadku nieużywana część palety może zostać wykorzystana do przechowania dodatkowych danych. Możliwa jest również odwrotna sytuacja – głębia kolorów będzie większa niż wielkość palety. Zazwyczaj aplikacje traktują wtedy brakującą część palety (jeśli pojawią się do niej odwołania) jako czarną. Natomiast programiści przeglądarki Apple Safari zapomnieli przewidzieć taką możliwość oraz zarezerwowali zbyt małą ilość pamięci dla palety, przez co możliwe stało się wyświetlenie na ekran fragmentu pamięci znajdującego się za paletą (identyczny błąd opisany był w przypadku BMP i przeglądarek Firefox i Opera w Hakin9 3/2008). Na szczęście dla użytkowników, programiści Apple nie zaimplementowali pobierania kolorów z bitmapy w tagu `<canvas>`, przez co tego błędu nie da się w żaden sensowny sposób wykorzystać.

Kolejnym polem jest flaga *SortFlag*, która, jeżeli jest ustawiona, mówi, iż w palecie barw kolory zostały posortowane od najważniejszych do najmniej ważnych. Taka informacja może być istotna w przypadku, gdy chcemy zmniejszyć ilość kolorów w GIF'ie, tracąc jednocześnie jak najmniej z jakości obrazu. Zazwyczaj jednak ta flaga jest ignorowana, dzięki czemu możemy w niej ukryć jeden bit danych.

Ostatnim polem jest pole *PixelAspectRatio*, które jest zazwyczaj ignorowane i może posłużyć do przechowania ukrytych informacji.

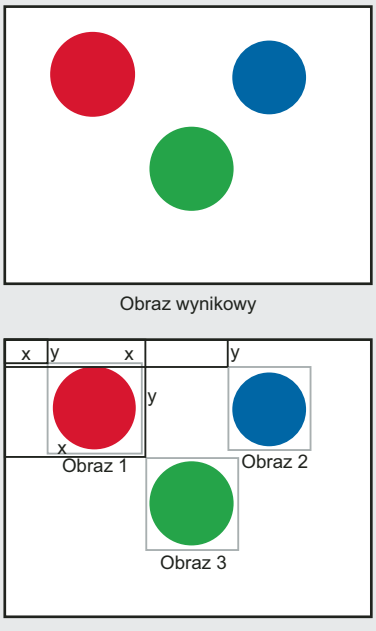
Tabela 4. *Struktura GIF98aRGB*

Typ i nazwa pola	Opis
BYTE Red	Wartość barwy czerwonej
BYTE Green	Wartość barwy zielonej
BYTE Blue	Wartość barwy niebieskiej

Warto również zwrócić uwagę na możliwość zadeklarowania bardzo dużej bitmapy (np. 65535x65535). Program, który będzie próbował zaalokować pamięć dla takiej bitmapy (prawie 4GB), spotka się prawdopodobnie z odmową ze strony menadżera pamięci. Jeżeli programista nie sprawdzi, czy alokacja się powiodła i będzie starał się odczytywać dane, doprowadzi to do próby zapisania danych do nieistniejącej pamięci, co skończy się prawdopodobnie przymusowym zakończeniem programu z informacją o błędzie. Gorszy przypadek zakłada możliwość takiego umieszczenia obrazu na logicznym ekranie, by jego dane spowodowały nadpisanie wrażliwych danych w pamięci.

Może się również zdarzyć, iż programista postanowi zaalokować pamięć od razu na bitmapę reprezentowaną w RGB. W takim wypadku skorzysta zapewne z równania `szerokość*wysokość*3`. Wynik takiego równania powinien zostać zapisany w zmiennej o wielkości minimum 64 bitów, jednak często zostaje wpisany po prostu do zmiennej o wielkości 32 bitów, co prowadzi w prostej linii do błędu typu przepełnienie zmiennej całkowitej (ang. *Integer Overflow*). Przykładowo, dla szerokości i wysokości równych kolejno 65535 oraz 21862 wynik wynosił będzie 4298178510, czyli heksadecymalnie 10030FFCE. Po zapisaniu tej liczby do 32-bitowej zmiennej dostaniemy heksadecymalnie jedynie 0x0030FFCE, czyli 3211214 (około 3MB). Programista zaalokuje więc prawdopodobnie 3MB, natomiast pozwoli, aby obraz wyrenderowany został w dowolnym miejscu, które wchodzi w zakres 0 do 65535 – szerokość obrazu, oraz 0 do 21862 – wysokość obrazu. Tego typu błąd prowadzi do wykonania kodu atakującego, i w konsekwencji do przejęcia kontroli nad programem.

Innym przypadkiem jest ekran logiczny, w którym zarówno wysokość, jak i szerokość są równe 0. Warto zwrócić uwagę iż wielkość (0 - szerokość obrazu) da – w zależności od użytej arytmetyki – albo bardzo dużą liczbę, albo liczbę ujemną. W tym drugim wypadku aplikacja może przepuścić taki obraz do renderowania, i w konsekwencji zakończyć działanie z błędem.



Rysunek 3. *Przykład użycia trzech obrazów na ekranie logicznym*

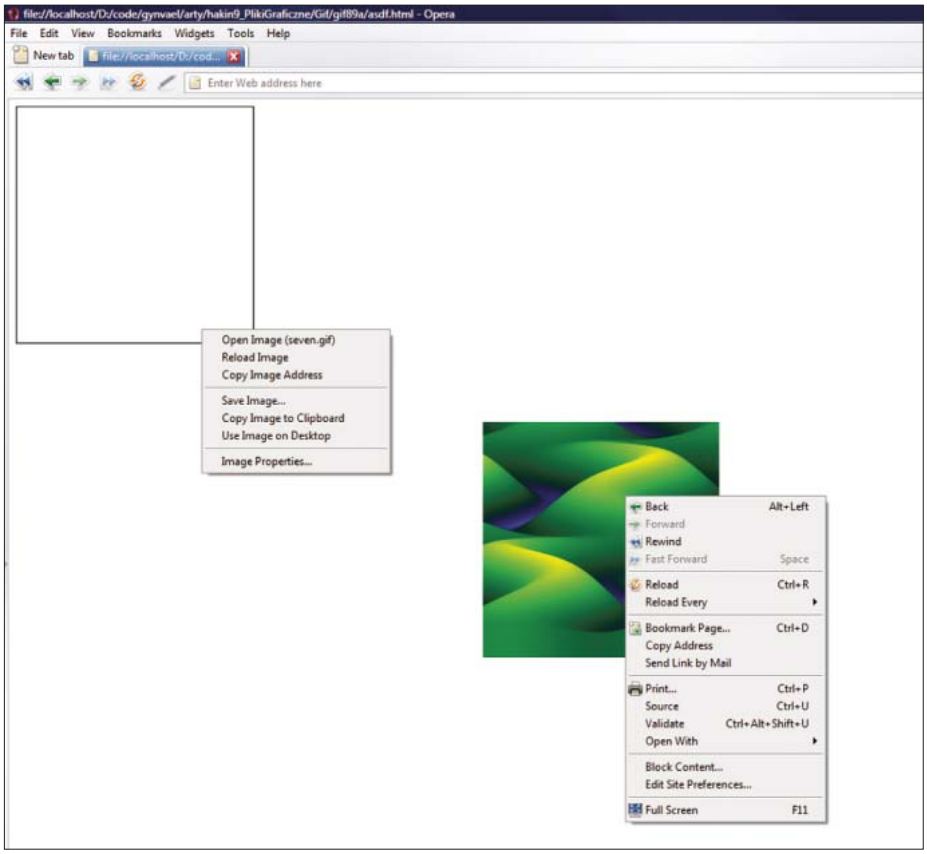
Image Descriptor

Struktura *Image Descriptor* pojawia się zawsze przed danymi obrazu i zaczyna się od bajtu 0x2C. Poza oczywistymi polami, i polami analogicznymi do występujących w strukturze LSD, znajdują się w niej dwa interesujące pola. Pierwszym z nich jest pole *Reserved*, które jest ignorowane i może posłużyć do ukrycia 2 bitów informacji. Drugim jest flaga, która, jeśli jest ustawiona, wskazuje na zapisanie obrazu z przeplotem (czyli wiersze nie są po kolei). To pole może zostać również wykorzystane do przechowania jednego bitu informacji, ale pod warunkiem, że obraz zostanie odpowiednio zakodowany.

W przypadku tej struktury należy uważać na wszelkie przejawy przepełnienia zmiennej całkowitej, analogiczne do tych pojawiających się w przypadku LSD.

GCT i LCT

Budowa globalnej i lokalnej palety barw jest identyczna. Paleta jest to po prostu tablica 3-bajtowych struktur (patrz Tabela 4.), opisujących dany kolor. Warto w tym miejscu pamiętać o steganograficznej metodzie polegającej na wpisaniu (o ile dostępne jest miejsce w palecie) danej barwy więcej niż jeden raz, dzięki czemu wybór koloru użytego do opisania danej barwy mógłby posłużyć do ukrycia danych.



Rysunek 4. *Opera skołowana przez GIF*

Warto pamiętać również o tym, że GIF może nie mieć ani palety globalnej, ani palety lokalnej. Co wrażliwsze aplikacje mogą reagować rzuceniem wyjątku na taką sytuację.

Table Based Image Data

Dane, oprócz tego że skompresowane, przechowywane są w specjalnych pakietach (patrz Rysunek 2.). Pierwszy bajt danych, *LZW Minimum Code Size*, jest informacją o początkowej ilości bitów przypadających na jeden wyjściowy, skompresowany kod danych. Aby z tego *LZW Minimum Code Size* uzyskać faktyczną wielkość kodu, należy do tej wartości dodać 1. Przykładowo, dla 8-bitowych bitmap, pole to ma wartość 8, a więc początkową wielkością kodów jest 8+1, czyli 9 bitów. Maksymalną wielkością kodu jest, jak zostało wspomniane już wcześniej, 12 bitów.

Okazuje się, iż niektóre aplikacje i biblioteki mają *na sztywno* ustawiony słownik na 4096 elementów (2^12) i nie zawsze sprawdzają, czy podany *LZW Minimum Code Size* nie przekroczy tej wielkości. Taki błąd został znaleziony przez autora w bibliotece SDL_Image 1.2.6 (w 1.2.7 został już poprawiony). Nieprawidłowość polegała na stworzeniu słownika o stałej wielkości 4096 bajtów, a następnie dopuszczenie, aby *LZW Minimum Code Size* miało inną wielkość. Tak oto wypełniając kolejne wpisy słownika przekraczało się w końcu 4096 elementów i dochodziło do błędu ze znanej klasy przepełnienia bufora (*http://vexillium.org/?sec-sdlgif*).

Zaraz po pierwszym polu znajdują się bloki danych. Każdy blok składa się z dwóch elementów. Pierwszym z nich jest jednobajtowe pole *BlockSize*, mówiące o wielkości partii danych (od 1 do 255), a drugim – odpowiedniej wielkości tablica danych. Specjalnym przypadkiem jest *BlockSize* o wartości 0 – jest to terminator danych, informujący dekoder, iż nie ma więcej bloków z danymi.

Ponieważ nie jest wymagane, aby *BlockSize* był zawsze maksymalnej wielkości (jeśli jest oczywiście odpowiednia ilość danych), to można, sterując wartością tego pola, użyć go do przechowania ukrytych danych. Zauważmy, że jeśli mamy skompresowane dane wielkości 10000 bajtów, to możemy równie dobrze stworzyć 39 bloków o wielkości 255 bajtów i jeden o wielkości 55 bajtów, jak i bloki o kolejno różnych wielkościach, np. odpowiadających zakodowanym informacjom – przykładowo o wielkościach kolejno 65, 76, 65, 32, 77, 65, 32, 75, 79, 84, 65 bajtów itd. (ciąg po skonwertowaniu na ASCII tworzy zdanie *ALA MA KOTA*).

Oczywiście optymalnym rozwiązaniem (pod względem wynikowej wielkości pliku), jest użycie bloków o największej możliwej wielkości. Jednak i tu jest pewna możliwość manewrów. Wracając do naszego przykładu, zamiast przygotować zestaw bloków 39x255 i 1x55, można zrobić 38x255, 1x254, 1x56. Należy zauważyć, iż ilość bloków pozostanie stała, a jednocześnie optymalna, oraz że umożliwia to zapis dodatkowych informacji (format GIF został chyba stworzony dla steganografów).

Same dane są oczywiście zakodowane (skompresowane) algorytmem LZW opisanym w pierwszej części artykułu. W tym miejscu pojawiają się kolejne dwie pułapki.

Pierwszą z nich jest przypadek, gdy po dekompresji dane LZW są większe od przewidzianej (wyliczonej z równania `wysokość * szerokość`) wielkości. W kodzie nieuważnego programisty można w tym miejscu znaleźć błąd przepełnienia bufora.

Drugim możliwym wariantem jest niedostateczna ilość danych. W przypadku, gdy program wyświetlający informacje nie wyczyścił (nie wypełnił kolorem tła) logicznego ekranu, można się spodziewać wyświetlenia części starych informacji zawartych w pamięci ekranu (tego rodzaju błąd może doprowadzić nawet do zdalnego ujawnienia informacji, ale o tym pisałem już wyżej). Ostatnia ciekawostka wynika z możliwości dopisania do skompresowanego ciągu dodatkowych danych, które przez prawidłowo napisany dekoder zostaną po prostu zignorowane.

Bezpieczna implementacja

Najbardziej wrażliwymi miejscami w implementacji każdego formatu są miejsca oznaczone w dokumentacji formatu jako *must* (musi zostać/musi być) oraz *should* (powinien). Okazuje się, iż mimo umieszczenia w specyfikacji informacji o konieczności lub powinności zrobienia czegoś w określony sposób czy przekazania konkretnej wartości (ewentualnie mieszczącej się w podanym zakresie), to na pewno znajdzie się osoba, która w to miejsce wstawi inną wartość lub zrealizuje to inaczej. Zazwyczaj programista zakłada, że jeśli standard narzuca określony sposób realizacji pewnej czynności, to nie trzeba sprawdzać, czy tak faktycznie jest. I tak niestety rodzą się poważne błędy. Jednym z przykładów takiej sytuacji, wymienionym już wcześniej w tym artykule, jest pole *LZW Minimum Code Size*, które według dokumentacji musi mieć wielkość 12 bitów. Można je jednak technicznie przestawić na dużo większą wartość. Dobry programista powinien implementować obsługę formatu według standardu, ale traktować go tylko z umiarkowanym zaufaniem i sprawdzać, czy w otrzymanym z zewnątrz pliku GIF faktycznie wszystko jest tak, jak musi (lub powinno) być.

Podsumowanie

GIF jest stosunkowo skomplikowanym formatem przechowywania informacji o obrazach. Należy więc zachować szczególną czujność przy implementacji jego obsługi. Dla *bughunterów* natomiast format GIF może okazać się kopalnią większych lub mniejszych luk i błędów.

W Sieci

- *http://www.w3.org/Graphics/GIF/spec-gif89a.txt* – standard GIF89a,
- *http://vexillium.org/?sec-sdlgif* – SDL_Image 1.2.6 GIF Buffer Overflow,
- *http://en.wikipedia.org/wiki/Graphics_Interchange_Format* – Wikipedia: GIF,
- *http://www.math.ias.edu/doc/libungif-4.1.3/UNCOMPRESSED_GIF* – Artykuł o nieskompresowanych GIF'ach,
- *http://dk.aminet.net/docs/misc/GIF24.readme* – Tworzenie 24-bitowych GIF'ów,
- *http://pl.wikipedia.org/wiki/LZW* – Wikipedia: Kompresja LZW.

Michał Składnikiewicz
Inżynier informatyki, ma wieloletnie doświadczenie jako programista oraz *reverse engineer*. Obecnie jest koordynatorem działu analiz w międzynarodowej firmie specjalizującej się w bezpieczeństwie komputerowym.
Kontakt z autorem: *gynvael@coldwind.pl*

Strony rekomendowane

Misją serwisu jest zaprezentowanie języków programowania oraz ułatwienie użytkownikowi ich szybkiej nauki.

http://www.cjp.xt.pl

Strona internetowa firmy Fit Consulting specjalizującej się w nowoczesnych rozwiązaniach informatycznych, zaczynając od sprzedaży sprzętu komputerowego i świadczeniu usług po zaawansowane rozwiązania zarządzania przedsiębiorstwem.

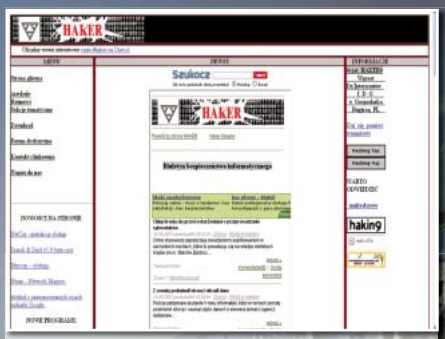
http://www.fit-consulting.pl/

Portal poświęcony technikom programowania oraz sposobom ochrony przed zagrożeniami jakie płyną z Internetu. Przedstawione techniki służą do celów edukacyjnych, nie należy ich wykorzystywać w niewłaściwy sposób.

http://www.hackerzy.pl

Witryna poświęcona w całości tematyce hakingu. Początkujący w tej dziedzinie znajdą na niej działy, ktore im umożliwią rozpoczęcie nauki.

http://www.haker.ocom.pl/



Do niedawna termin hacking był zarezerwowany tylko dla profesjonalistów. Na tym portalu można dowiedzieć się więcej o tym zagadnieniu. Znajdują się na nim również aktualności i obszerny dział downloads.

http://www.hakerczat.prv.pl/index.html

Strona firmy świadczącej profesjonalne usługii IT, specjalizującej się w wypożyczaniu serwerów i macierzy. Firma prowadzi także centrum szkoleniowe.

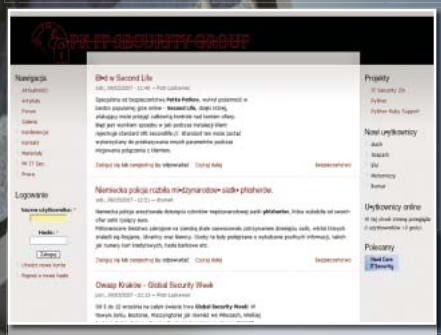
http://www.itlpolska.pl

Portal dla wszystkich zainteresowanych tematyką bezpieczeństwa, audytem, IT Governance w sieciach i systemach teleinformatycznych, a także ochroną danych.

http://www.locos.pl/

Portal internetowy poświęcony w całości branży IT. Podzielony jest na trzy działy poświęcone sprzętowi komputerowemu, grom i rozrywce oraz najnowszym aplikacjom.

http://www.pcarena.pl/index.php



Strona koła naukowego PK IT Security Group. Koło ma na celu dostarczanie informacji związanych z bezpieczeństwem.

http://www.pkitsec.pl/

Misją serwisu jest dostarczenie dużej ilości informacji z zakresu informatyki. Znajdują się na nim ciekawe artykuły, najświeższe informacje z rynku IT, recenzje książek jak i kursy tworzenia portali www.

http://www.swww.pl/

Strona dla każdego webmastera i hackera. Jeśli zawsze chciałeś stworzyć swoją stronę internetowej lub poznać haking od podszewki, to ten serwis Ci w tym pomoże.

http://web4u.neth.pl/

Serwis poświęcony branży IT oferujący codzienne newsy, artykuły, recenzje magazynów, testy oraz forum dyskusyjne.

http://www.webhat.pl/index.php

Strony rekomendowane

MARIUSZ RÓG

Zdalne zarządzanie: NetBus Pro

Stopień trudności

Artykuł przedstawi trojana NetBus. W prosty sposób wyjaśni zasadę działania oraz poszczególne funkcje aplikacji. NetBus jest jednym ze starszych trojanów, jakie ukazały się w sieci – powstał w 1998 roku. Został napisany przez Szweda, Carla Fredrika Neiktera. Jest to jeden z nielicznych dobrych programów typu backdoor.

Z ARTYKUŁU DOWIESZ SIĘ

jak prawidłowo skonfigurować serwer NetBus i używać klienta,

poznasz zalety oraz wady używania trojana do zdalnego zarządzania,

poznasz funkcjonalność programu oraz zapoznasz się z architekturą aplikacji.

CO POWINIENEŚ WIEDZIEĆ

czym są programy typu backdoor oraz jak się przed nimi zabezpieczyć,

jak konfigurować porty w zaporze ogniowej systemu Windows,

powinieneś posiadać podstawowe informacje na temat sieci LAN oraz konfiguracji protokołów komunikacyjnych.

NetBus, będąc na początku typową aplikacją *backdoor*, przez kilkanaście lat przekształcił się w interesujący program do zdalnego zarządzania komputerami. Jest całkiem bezpieczny, gdyż sygnatura jego kodu znajduje się w większości programów antywirusowych. Wszelka próba uruchomienia klienta lub serwera jest od razu zauważana przez aplikacje antywirusowe. Z tego też względu przed uruchomieniem programu należy skonfigurować program antywirusowy. Wystarczy wykluczyć aplikację z listy analizowanych programów. Narzędzie jest teraz prostsze w obsłudze oraz posiada szereg unikalnych funkcji. Dlatego też jest ciekawą alternatywą dla w pełni komercyjnych i skomplikowanych systemów zdalnego zarządzania.

Instalacja

Proces instalacji wersji 2.10 jest prosty. Po uruchomieniu instalatora oraz wybraniu miejsca instalacji instalator zadaje pytanie, które komponenty mają zostać zainstalowane.

Ze względu na mało intuicyjny podział aplikacji *NetBus* należy wyjaśnić, do czego służą poszczególne komponenty. Aplikacja *NetBus* podzielona jest na dwie części: serwer oraz klient. Część kliencka instalowana jest na komputerze zarządzającym, zaś część serwerowa – na komputerach zarządzanych. Po instalacji, w zależności od wybranych opcji, instalowane są dwa pliki wykonywalne *NetBus.exe* oraz *NBSvr.exe*. Program *NetBus* jest to aplikacja kliencka służąca do administracji innymi komputerami. *NBSvr* jest zaś serwerem udostępniającym usługi. W katalogu instalacyjnym oprócz plików wykonywalnych znajduje się kilka bibliotek dynamicznych.

Konfiguracja serwera

Kluczem do sprawnego oraz bezpiecznego zarządzania komputerem jest prawidłowa konfiguracja serwera. Domyślnie serwer jest wyłączony. Aby go włączyć, należy uruchomić plik *NBSvr.exe*. Po chwili ukazuje się okienko z listą podłączonych klientów (dzięki temu jesteśmy w stanie sprawdzić, kto zarządza komputerem). Okno przedstawione zostało na Rysunku 2.

Autor aplikacji twierdzi, iż jednym serwerem może zarządzać kilka klientów jednocześnie bez wprowadzania konfliktów. Można to wykorzystać używając jednego serwera jako np. repozytorium plików. Oprócz listy klientów, okno zawiera informacje o aktywnych serwerach. Do dyspozycji są trzy serwery – właściwy serwer *NetBus* oraz serwer *telnet* i *http*.

Konfiguracja serwera odbywa się przez wciśnięcie przycisku *Setting*. Włącza on okno konfiguracyjne pokazane na Rysunku 3.

W zakładce *General* dostępne są ogólne opcje związane z serwerem. Aby uruchomić serwer, należy zaznaczyć opcję

Accept connections. Następnie należy ustawić port, na którym serwer będzie nasłuchiwał połączeń.

Domyślnym portem jest 20034. Aby serwer pracował poprawnie, należy odblokować wskazany port w konfiguracji zapory ogniowej. Opcjonalnie można w polu *Password* określić hasło. Ze względu na dostęp do większości krytycznych zasobów komputera zaleca się stosowanie hasła.

Następnie należy wybrać stopień widoczności dla serwera. Do dyspozycji są cztery tryby widoczności przedstawione w Tabeli 1.

Należy być szczególnie ostrożnym przy użyciu trybu *Only in tasklist*, gdyż blokuje on całkowicie dostęp do okna konfiguracji, a co za tym idzie – dostęp do trybów. W tym przypadku możliwość zmiany trybu widoczności ma tylko klient znający hasło dostępu.

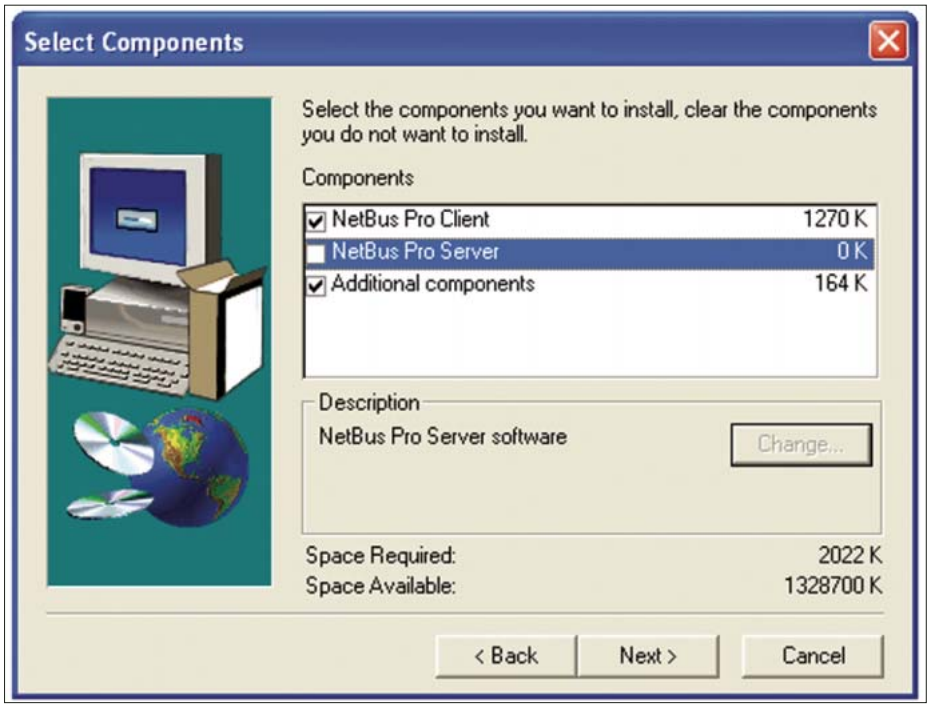
Do wyboru jest także lista trybów dostępu do komputera, odpowiednio od najbardziej restrykcyjnego do pełnego dostępu. Tryby te zostały opisane w Tabeli 2. Zmiana konfiguracji (w pewnym ograniczonym zakresie) może odbywać się również z poziomu klienta. Należy jednak zwrócić uwagę, iż ze względów bezpieczeństwa możliwość zdalnego modyfikowania trybu jest dostępna tylko w trybie pełnego dostępu.

Ostatnim elementem kompletującym konfigurację jest umożliwienie startowania serwera automatycznie. W zależności od potrzeb można włączyć tę opcję lub uruchamiać serwer ręcznie.

W wersji 2.10 opcja wyłączająca logowanie jest nieaktywna. Program zawsze będzie logował połączenia do pliku *Log.txt*, nie ma też możliwości zmiany nazwy pliku.

Tabela 1. *Lista trybów widoczności*

Widoczność	Opis
Full visible	Serwer będzie aktywny tylko, gdy okno aplikacji będzie włączone. Po wyjściu z programu serwer automatycznie kończy pracę.
Minimize as trayicon	Tryb podobny do pierwszego, lecz w tym przypadku serwer może być zminimalizowany do paska tray.
Only in tasklist	Tryb widoczności, w którym okno konfiguracji nie włącza się w ogóle. Serwer w tym trybie widoczny jest tylko za pośrednictwem managera zadań (taskmgr).
Invisible (95/98)	Tryb identyczny z poprzednim, lecz dostosowany do systemów Windows 95 i 98.



Rysunek 1. *Wybór komponentów przy instalacji NetBus*

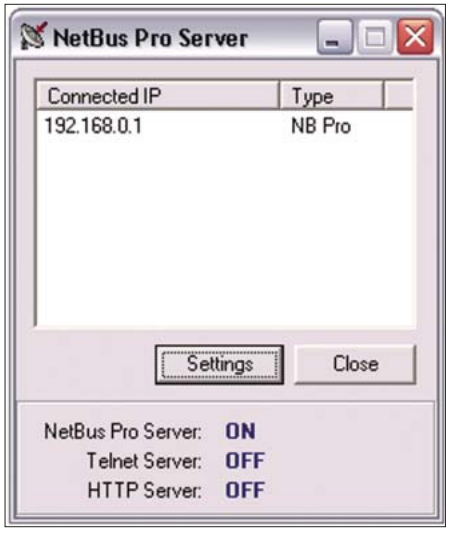
Dostęp do serwera

Każdy administrator zdaje sobie sprawę, że zabezpieczenie takiej funkcjonalności tylko hasłem nie jest dobrym rozwiązaniem. Jednak NetBus posiada dodatkową formę zabezpieczeń w postaci listy akceptowanych adresów, które mogą się połączyć z serwerem. Ustawia się ją przy pomocy klienta, przy użyciu menu *Server admin* (opcja *Restrict access*). Pozwala ona ręcznie określić konkretne adresy lub zakresy adresów, które są upoważnione do połączenia z serwerem. Przykładowe prawidłowe wpisy listy to:

- 192.168.0.1 (zezwala na połączenie z adresu 192.168.0.1),
- 192.168.0.* (zezwala na połączenia z adresów rozpoczynających się oktetami 192.168.0),
- 192.168.0.1-10 (zezwala na połączenie z każdego adresu z zakresu od 192.168.0.1 do 192.168.0.10).

Uruchomienie klienta

Prawidłowo skonfigurowany serwer udostępnia usługi na wskazanym porcie. Aby z nich skorzystać, należy uruchomić klienta za pomocą pliku wykonywalnego *NetBus.exe*. Sygnatura trojana zawarta jest również w programie klienckim. Należy więc odpowiednio skonfigurować program antywirusowy – tak, aby pozwolił załadować biblioteki aplikacji. W przeciwnym wypadku użytkownikowi zostaną zgłoszone błędy związane z brakiem możliwości załadowania procedur aplikacji.



Rysunek 2. *Okno włączonego serwera NetBus*

Jeśli klient został uruchomiony prawidłowo, to wyświetli się okno z listą hostów. Domyślnie dodany jest jeden host związany z adresem lokalnym klienta. Jeśli więc podczas instalacji zaznaczono opcje klienta i serwera, to istnieje możliwość podłączenia od razu do komputera klienta. Aby dodać dowolny host, należy znać jego adres IP lub nazwę hosta w sieci lokalnej. Odbywa się to przy użyciu menu opcji *New...* z menu *Host*. Następnie należy podać adres hosta, numer portu, dowolną nazwę identyfikującą komputer oraz opcjonalnie hasło i nazwę użytkownika na serwerze. Po wciśnięciu przycisku *OK* host zostanie dodany do listy. Mając tak przygotowaną listę, można dowolnie łączyć się z wybranym komputerem poprzez dwukrotne kliknięcie na element listy lub za pomocą menu podręcznego i opcji *Connect*. Należy zwrócić uwagę na fakt, że klient w konkretnym momencie może być podłączony tylko do jednego serwera. Jest to trochę kłopotliwe, jeśli zamierza się wykonać określoną operację na kilku hostach.

Funkcje kontrolne

Podłączony klient ma do dyspozycji bardzo szeroki wachlarz funkcji związanych ze zdalnym zarządzaniem komputera. Niektóre z tych funkcji umożliwiają głęboką ingerencję w sam system, na którym zainstalowany jest *NetBus*. Zaleca się dużą ostrożność w ich użyciu. Funkcje podzielone są na logicznie powiązane moduły.

Chat manager

Moduł *Chat manager* jest przydatną funkcją pozwalającą na komunikację w obie strony (między użytkownikiem klienta i użytkownikiem

Rysunek 3. *Ustawienia serwera NetBus*

Tabela 2. *Tryby dostępu do komputera z zainstalowanym serwerem*

Widoczność	Opis
Basic access	Podstawowy tryb umożliwiający tylko otrzymanie informacji o hoście.
Spy mode access	Umożliwia tylko śledzenie działań użytkownika.
Full access	Tryb dający pełną kontrolę nad funkcjonalnością serwera NetBus.

Tabela 3. *Lista funkcji podmenu Spy functions*

Nazwa funkcji	Opis funkcji
Keyboard listen	Pozwala na podsłuchiwanie komunikatów klawiatury.
Capture screen image	Generuje zrzut ekranu do pliku.
Capture camera video	Przechwytuje obrazy z kamery podłączonej do serwera.
Record sound	Przechwytuje dźwięk z mikrofonu podłączonego do serwera.

Tabela 4. *Lista funkcji podmenu Cool functions*

Nazwa funkcji	Opis funkcji
CD-ROM	Pozwala na wysuwanie oraz wsuwanie podstawki napędu optycznego.
Disable keys	Umożliwia tylko śledzenie działań użytkownika.
Key click	Włącza dźwięk klawiszy w systemie.
Swap mouse	Zamienia miejscami przyciski myszki.
Go to URL	Uruchamia wskazany adres w domyślnej przeglądarce internetowej.
Send text	Wstawia wskazany tekst w pole edycyjne (jeśli istnieje) aktywnego okna.

Rysunek 4. *Moduł Chat Manager*

serwera). Rysunek 4 przedstawia okno uruchomionego modułu *Chat Manager*.

Górne pole służy do wprowadzania, a dolne – do wypisywania otrzymanego tekstu. Aby rozpocząć rozmowę, wystarczy nacisnąć przycisk *Start*. W tym momencie na ekranie serwera pojawi się podobne okienko rozmowy i zostanie uruchomiony system konwersacji. Ciekawym jest fakt, że rozmowa odbywa się w czasie rzeczywistym, tzn. każdy z naciśniętych znaków jest natychmiast transportowany protokołem do odbiorcy i wypisywany na ekranie. Niesie to za sobą kilka problemów implementacyjnych, których autor w wersji 2.10 nie rozwiązał. Dla przykładu – próba poprawienia wpisanego tekstu klawiszem *Backspace* powoduje wypisanie u odbiorcy pionowej kreski. Oprócz tego moduł posiada kilka błędów implementacyjnych, np. możliwe jest pisanie tekstu w okienku odczytu, zaś wpisanie tekstu, a następne naciśnięcie przycisku *Start* powoduje naruszenie ochrony pamięci oraz brak możliwości przewijania tekstu u odbiorcy.

Moduł, pomimo swoich niedoskonałości, posiada pewną dodatkową funkcjonalność rekompensującą niedopatrzenia autora – mianowicie wbudowany moduł *Message manager*. Uruchamia się go za pomocą przycisku *Messages...* w oknie modułu *Chat manager*. Uruchomiony moduł widać na Rysunku 5. Służy on do wywoływania okienek informacyjnych przy wykorzystaniu *WinApi* na hoście. Okienka mogą mieć różną postać w zależności od rodzaju wybranych opcji, wszystkie jednak bazują na funkcji *MessageBox* i jej flagach. Informacja zwrotna przekazywana jest do klienta, a ten wywołuje własną funkcję *MessageBox* z odpowiedzią.

Rysunek 5. *Moduł Message manager*

File manager

Moduł *File manager* pozwala, dzięki protokołowi aplikacji, ingerować w system plików hosta. Wykorzystanie modułu nie jest skomplikowane – użycie sprowadza się do naciskania odpowiednich przycisków w okienku. Niestety, *File manager* nie radzi sobie dobrze z plikami zajętymi przez inne aplikacje. Przegrywanie pewnych plików z systemu hosta w tym przypadku jest po prostu niemożliwe. Na dokładkę użytkownik klienta nie jest poinformowany o problemie kopiowania plików, zmuszony jest czekać. Jedyne, co pozostaje w takiej sytuacji, to wciśnięcie przycisku *Cancel* w okienku transportowym. Uruchomiony moduł można zobaczyć na Rysunku 6. Obrazki reprezentujące przyciski są raczej intuicyjne. Po prawej stronie okienka ułożone są trzy przyciski – odpowiednio: przejście do folderu powyżej, usunięcie folderu oraz stworzenie nowego folderu. Operacji na strukturze katalogów wykonuje się przy użyciu przycisków z dołu ekranu i nie wymagają one wyjaśnień. Niestety, funkcje udostępniania stworzone zostały pod system Windows 95/98. Na systemach opartych o NT ta funkcjonalność nie będzie działać.

Windows manager

Windows manager to skomplikowany w obsłudze moduł, pozwalający w pewnym zakresie zarządzać oknami uruchomionych programów na systemie hosta. Dla osoby zarządzającej lista okien na serwerze widoczna jest w postaci drzewa. W początkowej fazie korzystania z modułu ciężko jest się zorientować w hierarchii okien. Na szczęście po krótkiej chwili można już wykonać podstawowe czynności związane z poszczególnymi okienkami. Zaznaczenie opcji *Show only visible windows* oraz *Show only named windows* pozwala odfiltrować drzewo z okien, które nie są widoczne i nie posiadają nazwy. Funkcjonalność modułu nie jest zbyt rozbudowana, sprowadza się do prostego wyłączania okienek, zmiany wielkości, zmiany położenia oraz ustawiania tzw. fokusu. Pozwala również na modyfikacje czterech flag okna (*Is visible*, *Is enabled*, *Is checked* oraz

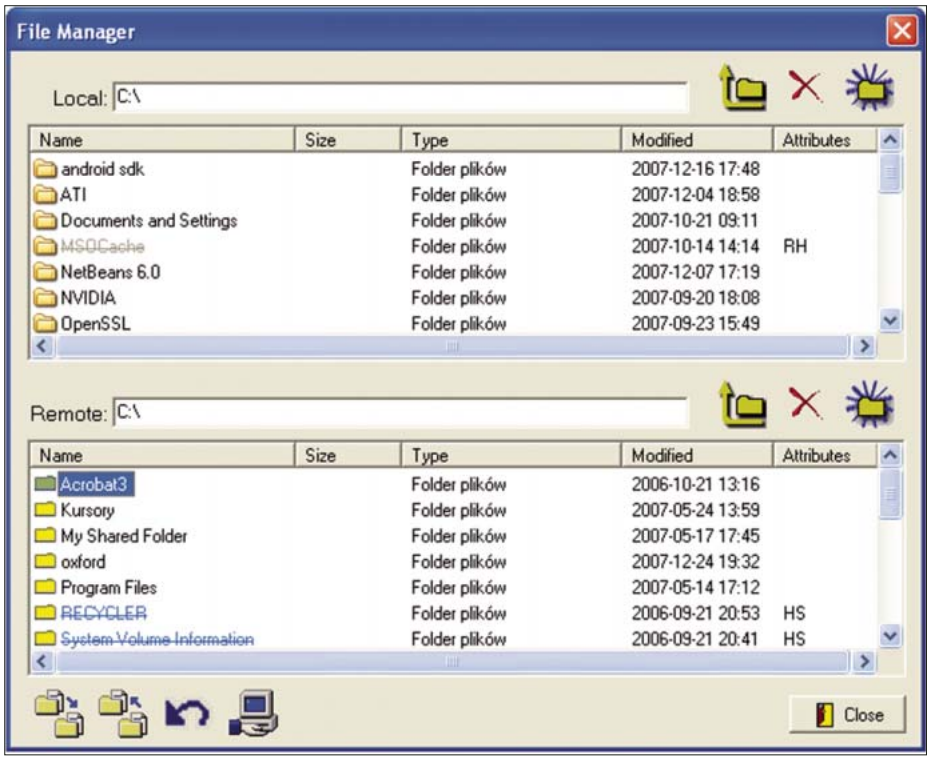
Tabela 5. *Lista poleceń skryptów*

Polecenie	Opis polecenia
MessageBox	Wyświetla okno informacyjne. Parametry: typ okna, wiadomość.
DeleteFile	Kasuje plik. Parametr: ścieżka do zdalnego pliku.
NewFolder	Tworzy nowy katalog. Parametr: ścieżka do katalogu.
DownloadFile	Pobiera plik ze zdalnego systemu plików.
UploadFile	Ścieżka do katalogu zdalnego.
AddRegData	Dodaje wpis do rejestru. Parametry: ścieżka w rejestrze, nazwa i wartość wpisu.
	Usuwa wpis z rejestru. Parametry: ścieżka w rejestrze, nazwa wpisu.
RunPlugin	Uruchamia plugin z biblioteki dynamicznej.
ExecuteFile	Uruchamia plik wykonywalny. Parametr: ścieżka do uruchamianego pliku.
PlaySound	Odtwarza dźwięk. Parametr: ścieżka do pliku w formacie *wav*.
ShowImage	Wyświetla obraz. Parametr: ścieżka do pliku obrazu.
OpenCD	Otwiera napęd CD-ROM. Parametr: liczba 1, reprezentująca wartość *true*.
ScreenDump	Zapisuje obraz ekranu do podanego katalogu. Parametr: katalog zapisu obrazu.
ExitWindows	Zamyka system Windows.
DisableKeys	Wyłącza wybrane klawisze. Parametr: zestaw klawiszy do wyłączenia.
KeyClick	Aktywuje dźwięk klawiszy. Parametr: liczba 1, reprezentująca wartość *true*.
SendText	Wpisuje tekst w pole edycyjne aktywnego okna. Parametr: tekst do wpisania.
SwapMouse	Zamienia klawisze myszki. Parametr: liczba 1, reprezentująca wartość *true*.

Always on top). W module brakuje niestety podglądu wykonywanych czynności. Wszystkie operacje wykonuje się zatem bazując na współrzędnych oraz pikselach. Oczywiście zawsze można użyć funkcji *Capture screen image* w celu podejrzenia wyniku działania modułu.

Registry manager

Jak sama nazwa wskazuje, moduł pozwala przeglądać i modyfikować strukturę rejestru hosta. Jest to bardzo niebezpieczne, a sam moduł nie jest tak poręczny, jak aplikacja Edytor Rejestru w systemie Windows. Moduł umożliwia edycję, tworzenie i kasowanie kluczy w rejestrze oraz modyfikację, tworzenie i kasowanie samych wpisów. Nie dostarcza jednak funkcjonalności przeszukiwania rejestru. Korzystając z *Registry managera* użytkownik musi posiadać podstawową wiedzę o drzewie kluczy rejestru. Dlatego też używanie tego modułu rekomenduję tylko doświadczonym użytkownikom.



Rysunek 6. *Moduł File Manager*

Spy functions

NetBus posiada cztery funkcje służące do śledzenia działań użytkownika. Wraz z opisem znajdują się one w Tabeli 3. Biorąc pod uwagę uwarunkowania prawne – ich używanie jest zabronione, jeśli osoba śledzona nie jest o tym poinformowana. Najbardziej niebezpieczna jest funkcja przechwytująca zdarzenia klawiatury. Dzięki niej od razu możemy wychwycić loginy i hasła do dowolnego systemu, do którego użytkownik będzie się logował lub zwyczajnie podglądać pisane przez nieświadomego użytkownika teksty.

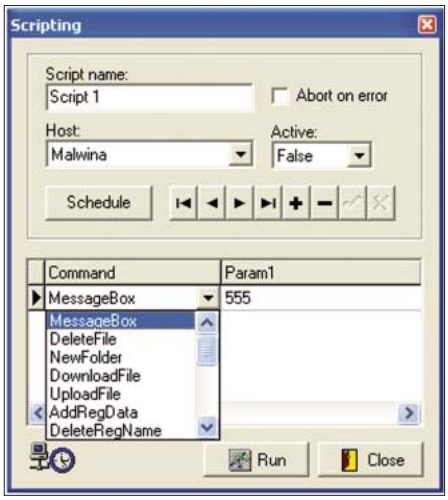
Cool functions

Niewątpliwie *Cool functions* jest zestawem najmniej przydatnych funkcji w programie, z reguły służących do zabawy i denerwowania użytkownika. Listę funkcji oraz ich opis znaleźć można w Tabeli 4. Należy przyznać, iż funkcje z tej kategorii nie są dostępne w innych programach służących do zdalnego zarządzania.

Rysunek 7. *Uruchomione okno Find Host*

Zarządzanie hostami

Menu *Host* posiada kilka funkcji, z których część posiada oczywiste



Rysunek 8. *Okno edycji skryptu*

znaczenie, a inne wymagają omówienia. Są to funkcje, dzięki którym *NetBus* ma prawo być traktowany jako aplikacja do zdalnego zarządzania komputerami.

Find Host

Uruchomienie funkcji *Find Host* powoduje otwarcie okienka przedstawionego na Rysunku 7.

Służy ono do znalezienia hostów z zainstalowanym serwerem *NetBus* w sieci lokalnej. Dzięki tej funkcjonalności administrator nie ma potrzeby pamiętania adresów wszystkich serwerów. Aby odnaleźć interesujące hosty, należy podać zakres adresów IP do przeszukania, po czym określić port, na którym serwery nasłuchują połączeń, a na końcu – ilość gniazd (*Sockets*) używanych podczas szukania. Następnie, po naciśnięciu przycisku *Start*, uruchomiony zostanie proces przeszukiwania. Każdy z adresów z zakresu zostanie odpytany o połączenie na wskazanym porcie. Jeśli host będzie miał aktywny serwer oraz zezwoli na połączenie, to zostanie dodany do listy *Found hosts*. Z listy można wybrać znalezione serwery i dodać je przyciskiem *Add* do głównej listy.

Script

Funkcja *Script* jest potężnym narzędziem, które pozwala na wykonanie sekwencji operacji na zdalnym serwerze. Okno edycji skryptu przedstawione zostało na Rysunku 8. Stworzenie skryptu rozpoczyna się od wybrania opcji *Script* z menu *Host*. Następnie należy podać nawę skryptu, jednoznacznie identyfikującą jego działanie. Nie jest zabronione powtarzanie nazw skryptów, tak więc zaleca się używanie nazw unikalnych. Opcja *Active* ustawiona na wartość *false* pozwala na wykonanie skryptu w późniejszym czasie. Skrypt składa się z zestawu poleceń opisanych w Tabeli 5. Naciśnięcie przycisku *Run* uruchamia proces wykonywania wybranych poleceń na zdalnym hoście. Uruchomienie skryptu można uzależnić od czasu na serwerze. Służy do tego opcja *Schedule*.

Broadcast

Ostatnia z omawianych funkcji aplikacji *NetBus* jest ściśle związana z funkcją *Script*. Służy ona do dystrybucji skryptu na wiele serwerów w tym samym czasie. Umożliwia to masowe wykonywanie zaplanowanych zadań. Przed wykonaniem funkcji *Broadcast* należy stworzyć skrypt w menu *Script*. Następnie trzeba go wybrać z listy wyboru *Script name* w oknie *Broadcast*. Ostatnią czynnością do wykonania jest podanie zakresu adresów IP, do których będzie wysłany skrypt. Uruchomienie wysłania odbywa się przez naciśnięcie przycisku *Run*. W tym momencie każdy z hostów z wskazanego zakresu otrzymuje kopię skryptu.

Podsumowanie

Aplikacja *NetBus* posiada kilka błędów, które potrafią być dokuczliwe. Mimo wszystko dostarcza wielu użytecznych i unikatowych funkcji, pozwalających w przyjemny sposób zarządzać komputerami. Dużym plusem aplikacji jest fakt, iż użytkownik zdalnego komputera może bezproblemowo pracować w czasie wykonywania operacji przez aplikację. Same funkcje narzędzia są logicznie poukładane i intuicyjne. Jak na tak niewielki program, wachlarz możliwości jest naprawdę szeroki. W zupełności wystarczy dla małych i średnich sieci.

Mariusz Róg
Autor jest programistą Javy. Posiada wiedzę z zakresu metod sztucznej inteligencji oraz rozwiązań klasy biznesowej i rozwiązań mobilnych. Obecnie pracuje na stanowisku Specjalisty ds. Produkcji Oprogramowania w firmie BLStream Sp. z o. o. BLStream jest międzynarodowym dostawcą i integratorem nowoczesnych systemów informatycznych oraz producentem oprogramowania mobilnego dla sektora medialnego, telekomunikacyjnego, finansowego i ubezpieczeniowego. Główna siedziba firmy mieści się w Szczecinie, a jej przedstawicielstwa i centra programistyczne ulokowane są we Wrocławiu, Warszawie, Helsinkach i Lwowie.
Kontakt z autorem: *mariusz.rog@gmail.com*

PRZEMYSŁAW SKOWRON

Suhosin: Bezpieczne aplikacje PHP

Stopień trudności

Bezpieczeństwo aplikacji WWW bardzo kuleje, a prostota ich tworzenia w języku PHP skutkuje ich bardzo dużą popularnością. Problem z utrzymywaniem aplikacji pochodzących z trzeciej ręki rośnie; nie każda firma przeprowadza testy bezpieczeństwa swoich produktów, a i one nie zawsze są skuteczne w 100%.

Praca administratora serwera WWW nie należy do najłatwiejszych. Pomijając zapewnienie odpowiedniej wydajności i wysokiej dostępności, powinien on również dbać o bezpieczeństwo. Często, nie mając wpływu na jakość kodu aplikacji webowych, musi poradzić sobie z tymi problemami sam. Na środowisko, w którym pracują takie aplikacje, czyha wiele niebezpieczeństw. Powszechnie znane ataki typu XSS, SQL Injection, a ostatnio bardzo popularne i bezwzględne XSRF – to nie wszystko, z czym zmaga się programista i administrator. Ataki przepełnienia bufora, *NULL (%00) byte*, *Denial of Service* są tylko przykładami kolejnych podatności, jakie czekają na wyeliminowanie. W tym artykule postaram się przybliżyć kilka mniej lub bardziej znanych ataków na aplikacje webowe; problemy, z jakimi można się spotkać utrzymując serwer WWW z programistami nie do okiełznania, ale przede wszystkim – jak skutecznie radzić sobie w takich sytuacjach.

Warstwy bezpieczeństwa aplikacji webowych

Tworzenie możliwie bezpiecznego środowiska dla aplikacji webowych może być – wbrew pozorom – bardzo złożone. Możliwości jest wiele i zawsze należy zastanowić się, z czego można zrezygnować, a z czego na pewno nie. Idąc od najniższej warstwy, w zależności od wyboru systemu operacyjnego, można stosować specjalnie przygotowane kernele (np. w systemie Linux) lub wbudowane opcje konfiguracji pracy systemu (Linux / Windows 2003 Server), które utwardzają te systemy. Dalej jest warstwa aplikacji, ale jeszcze nie tej, którą będziesz musiał okiełznać jako administrator serwera WWW. Dostępna jest tutaj cała masa rozwiązań: od specjalnych kompilatorów (np. dodatkowo *ProPolice* do *gcc*), bibliotek podmieniających niebezpieczne funkcje na ich bezpieczniejsze odpowiedniki, przez zamykanie demonów usług w środowiskach odizolowanych od reszty systemu, degradację uprawnień, z jakimi pracują aplikacje, aż po przeróżne rozwiązania będące specjalnymi bibliotekami języków, w których programista tworzy aplikacje webowe. Owe aplikacje mają za zadanie odfiltrowanie niebezpiecznego kodu wstrzykniętego przez atakującego. Na samym końcu mamy firewalle aplikacyjne. Oczywiście można próbować stosować do obrony systemy IDS/IPS, jednak nie jest to ani łatwe, ani przyjemne. Nie są również tematem tego artykułu. Celowo pominąłem jeszcze jedną z grup rozwiązań podnoszących poziom bezpieczeństwa aplikacji webowych. Do tej grupy należy gwiazda tego artykułu, Suhosin.

Suhosin

Anioł Stróż (*Suhosin* w języku koreańskim) wypełnia niszę w całej gamie rozwiązań zabezpieczających pracę aplikacji webowych. Jest on przeznaczony do ochrony aplikacji stworzonych w języku PHP w wersji 4 i 5. Jest niezależny od programisty i to pozwala być pewnym, że masz pełną kontrole nad środowiskiem, w jakim będzie pracował jego twór.

Z ARTYKUŁU DOWIESZ SIĘ

jak zwiększyć bezpieczeństwo serwera WWW przy świadczeniu usług hostingowych,

z jakimi problemami możesz się spotkać utrzymując nieznane aplikacje webowe,

czym jest Suhosin i jak go używać.

CO POWINIENEŚ WIEDZIEĆ

znać podstawy systemu Linux,

potrafić zainstalować i skonfigurować środowisko dla serwera WWW z obsługą PHP,

znać podstawy PHP i HTML.

Architektura

Suhosin składa się z dwóch rozłącznych części. Każda z nich udostępnia inne funkcje. Rekomendowane jest używanie obydwu jednocześnie. Z uwagi na fakt, że jedna część to łatka na kod interpretera PHP, a druga to moduł, który może być ładowany jako extension – rozwiązanie to jest niezależne od platformy sprzętowej, używanego systemu czy serwera WWW! Rzadko kiedy rozwiązania tego typu są w podobnym stopniu elastyczne.

Suhosin Patch

Aby zainstalować łatkę, należy pobrać ze strony www.php.net źródła PHP w wersji, która Cię interesuje (ja wybrałem wersję 5.2.5) i łatkę Suhosina ze strony *http://www.hardened-php.net/suhosin/download.html*. Autor Suhosina dba o to, by łatki do wszystkich nowych wersji PHP były dostępne możliwie jak najszybciej od daty ich wydania. Bez trudu znajdziesz tam łatki także dla PHP w wersji 4, nawet tak archaicznej jak 4.3.11. Jak zbudować PHP z Suhosin? Patrz Listing 1.

Właściwie to wszystko co należy zrobić. W dalszym etapie należy skompilować PHP z tak nałożoną łatką. Operacja ta nie różni się niczym od standardowych instalacji PHP, więc ten krok pominę. Co osiągniesz, mając tak przygotowane PHP?

Zadaniem Suhosin Patch jest ochrona silnika interpretera PHP. Polega ona na obserwacji buforów – tak, by nie następowały ich przepełnienia (ang. *buffer overflow*). Autor łatki wykorzystał *kanarki* (ang. *canary value*), mechanizm podobny do stosowanego w kompilatorach. Wstawiając *kanarka* przed adresem powrotu i odkładając go w obszarze pamięci, który nie może być zmodyfikowany, uniemożliwia się zmianę wartości `RET` (adres powrotu) bez nadpisania kanarka. Podobną ochroną otoczone są także inne struktury danych, takie jak listy i tablice asocjacyjne. Wymieniona została również funkcja `realpath()`, która w różnych wersjach powodowała problemy przy współpracy z interpreterem PHP. Nie można zapomnieć również o ochronie przed atakami typu format string, zarówno w przypadku samego silnika PHP, jak i rozszerzeń. Wbudowane mechanizmy zabezpieczające silnik PHP nie są konfigurowalne, tzn. nie można ich włączyć albo wyłączyć. Bardzo mocnym argumentem za stosowaniem łatki jest skuteczna ochrona

Listing 1. *Nakładanie łatki Suhosin na PHP*

```
rez@dojo-labs:~/php_with_suhosin$ wget http://download.suhosin.org/suhosin-patch-
                    5.2.5-0.9.6.2.patch.gz
# Należy rozpakować źródła PHP oraz łatkę.
rez@dojo-labs:~/php_with_suhosin$ tar zxj php-5.2.5.tar.bz2
rez@dojo-labs:~/php_with_suhosin$ gzip -d suhosin-patch-5.2.5-0.9.6.2.patch.gz
# Proponuję przeprowadzić próbę nałożenia łatki, zanim faktycznie to zrobisz:
rez@dojo-labs:~/php_with_suhosin$ patch --dry-run -p0 < suhosin-patch-5.2.5-0.9.6.2.patch
# Jeżeli nie zobaczyłeś błędów przy symulacji nakładania łatki, można to zrobić już
                    bez parametru -dry-run:
rez@dojo-labs:~/php_with_suhosin$ patch -p0 < suhosin-patch-5.2.5-0.9.6.2.patch
# Dla pewności możesz sprawdzić kod, jaki zwróciła ta operacja:
rez@dojo-labs:~/php_with_suhosin$ echo $?
0
# Wartość 0 oznacza zakończenie operacji sukcesem, bez błędów.
```

Listing 2. *Instalacja Suhosin Extension*

```
rez@dojo-labs:~/php_with_suhosin$ wget http://download.suhosin.org/suhosin-0.9.23.tgz
#Rozpakowujemy źródła:
rez@dojo-labs:~/php_with_suhosin$ tar zxf suhosin-0.9.23.tgz
#Kompilujemy rozszerzenie:
rez@dojo-labs:~/php_with_suhosin$ cd suhosin-0.9.23
rez@dojo-labs:~/php_with_suhosin$ phpize
rez@dojo-labs:~/php_with_suhosin$ ./configure
rez@dojo-labs:~/php_with_suhosin$ make && make install
```

Listing 3. *Formularz wgrywania pliku*

```
<form method="post" action="up.php" enctype="multipart/form-data">
<input type="hidden" name="MAX_FILE_SIZE" value="30000">
File to upload:<br>
<input type="file" name="file" size="40"><br>
<input type="submit" value="Upload">
```

Listing 4. *Przejęcie wgrywanego pliku i zapisanie na dysku*

```
<?
    move_uploaded_file($_FILES['file']['tmp_name'],"/tmp/".$_FILES['file']['name']);
?>
```

Listing 5. *Przykładowy kod programu w C*

```
#include <stdio.h>
int main() { printf("test ELFa\n"); return 0; }
```

Listing 6. *Sprawdzenie typu pliku*

```
rez@dojo-labs:~/tmp$ file a
a: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8,
                    dynamically linked (uses shared libs), not stripped
```

Listing 7. *Wycinek z logów Suhosin – porzucenie próby uploadu pliku ELF*

```
Jan 25 01:33:49 dojo-labs suhosin[8532]: ALERT - uploaded file is an ELF executable -
                    file dropped (attacker '127.0.0.1', file '/var/www/t/up.php')
```

Listing 8. *Skrypt wypisujący zawartość zmiennej 'test', wysłanej metodą GET*

```
<?php
echo $_GET['test'];
?>
```

Listing 9. *Wywołanie lwp-request w celu testu skryptu r.php*

```
rez@dojo-labs:~/tmp$ lwp-request -Sd "http://localhost/t/r.php?test=`perl -e 'print
                    "A"x25'`"
GET http://localhost/t/r.php?test=AAAAAAAAAAAAAAAAAAAAAAAAA --> 200 OK
```

PHP przed wykorzystywaniem błędów w implementacji języka. Często, gdy była odkrywana dziura w interpreterze, wersja PHP z Suhosinem nie była podatna z uwagi na wewnętrzne mechanizmy ochrony struktur danych. W celu bliższego poznania ich implementacji zapraszam do studiowania suhosin-patch-5.2.5-0.9.6.2.patch – to bardzo przyjemna i pouczająca lektura.

Suhosin Extension

Zaczynamy instalację rozszerzenia Suhosin'a. (Listing 2). W pliku php.ini ustawiamy ładowanie rozszerzenia:

```
extension=suhosin.so
```

Należy pamiętać o tym, by plik suhosin.so znajdował się w katalogu wskazanym przez dyrektywę `extension_dir` w pliku *php.ini*.

Przejdźmy do listy możliwości *Suhosin extension*. Z tego względu, iż jest ona bardzo długa, przedstawię tylko te najciekawsze.

Upload plików

Suhosin posiada dość ciekawy mechanizm restrykcji dotyczących uploadu plików. Za pomocą dyrektyw w pliku *php.ini* można skonfigurować limit upload'owanych plików w ramach jednego zapytania (`suhosin.upload.max_uploads`), a także zabronić uploadu określonego typu plików.

Mając prosty formularz do wgrywania plików (Listing 3) oraz skrypt przejmujący wgrywany plik (Listing 4), uruchamiamy przeglądarkę WWW i wchodzimy pod adres, gdzie dostępny jest formularz: *http://suhosin-test/up.html*

Na uboczu napiszmy prosty program, który wypisze coś na ekranie (Listing 5). Skompilujmy i sprawdźmy czy program działa:

```
rez@dojo-labs:~/tmp$ gcc a.c -o a
rez@dojo-labs:~/tmp$ ./a
test ELFa
```

Upewnijmy się, że mamy do czynienia z plikiem typu ELF, np. za pomocą aplikacji *file* (Listing 6.)

Spróbujmy wrzucić tę aplikację na serwer. Patrząc na kod *up.php* (Listing 4.), aplikacja 'a' powinna znaleźć się w katalogu */tmp*.

```
rez@dojo-labs:~/tmp$ ls -lht /tmp/a
ls: /tmp/a: No such file or directory
```

Aplikacja nie znalazła się więc w oczekiwanym miejscu – spójrzmy zatem w logi (Listing 7.)

Dzięki dyrektywie `suhosin.upload.disallow_elf` upload nie powiódł się. Przy takiej blokadzie nie uda się umiejscowić na serwerze WWW kodu *exploita/robaka* lub innego niebezpiecznego oprogramowania w formacie *ELF*.

Dostępne są również inne dyrektywy, uniemożliwiające upload plików binarnych (`suhosin.upload.disallow_binary`) lub usuwające binarną zawartość z pliku (`suhosin.upload.remove_binary`).

Do szczęścia brakuje jeszcze oddanie decyzji o tym, czy plik podany przez użytkownika w formularzu jest pożądany, zewnętrznej aplikacji. Brakuje? Za pomocą suhosin.upload.verification_script możemy podpiąć dowolną aplikację, która – zwracając określoną wartość – decyduje, czy plik zostanie zaakceptowany (kod 1 oznacza akceptację pliku).

Wykroczenie? Akcja!

Domyślną akcją, jaką wykona Suhosin, kiedy wykryje nadużycie ze strony użytkownika lub kodu, jest zablokowanie związanej z tym zdarzeniem akcji (kodu). Czasami daje to niepożądany efekt. Dzięki ustawieniom suhosin.filter.action można ustawić przekierowanie podając adres URL lub ścieżkę do skryptu, który zostanie uruchomiony zamiast blokady akcji. Przy ustawieniach:

```
suhosin.filter.action =
      http://www.google.com
suhosin.get.max_value_length = 25
```

I skrypcie PHP *r.php*, przedstawionym na Listingu 8, spróbujemy odwołać się do niego przy pomocy aplikacji `lwp-request` (Listing 9). Wszystko działa.

Listing 10. *Próba przekroczenia limitu długości argumentu test w skrypcie r.php*

```
rez@dojo-labs:~/tmp$ lwp-request -Sd "http://localhost/t/r.php?test=`perl -e 'print
                     "A"x26'`"
GET http://localhost/t/r.php?test=AAAAAAAAAAAAAAAAAAAAAAAAAA --> 302 Found
GET http://www.google.com --> 302 Found
GET http://www.google.pl/ --> 200 OK
```

Listing 11. *Pierwszy test na długość Cookie*

```
rez@dojo-labs:~/tmp$ printf 'GET /t/r.php?test=123 HTTP/1.0\r\nCookie: testowe=012345\
                     r\n\r\n' | nc localhost 80
HTTP/1.0 200 OK
Connection: close
X-Powered-By: PHP/5.2.5
Content-type: text/html
Content-Length: 3
Date: Fri, 25 Jan 2008 01:56:50 GMT
Server: lighttpd/1.4.18
123
```

Listing 12. *Drugi test na długość Cookie*

```
rez@dojo-labs:~/tmp$ printf 'GET /t/r.php?test=123 HTTP/1.0\r\nCookie:
                     testowe=0123456\r\n\r\n' | nc localhost 80
HTTP/1.0 302 Found
Connection: close
X-Powered-By: PHP/5.2.5
Location: http://www.google.com
Content-type: text/html
Content-Length: 0
Date: Fri, 25 Jan 2008 01:56:56 GMT
Server: lighttpd/1.4.18
```

Listing 13. *Rekurencja*

```
<?php
function r($a) {
    echo "Wywolanie $a ";
    ++$a;
    r($a);
}
    r(1);
?>
```

Natomiast przekraczając maksymalną dopuszczalną wartość długości zmiennej `'test'.(25 znaków)` – patrz Listing 10.

Zostaliśmy przekierowani na stronę *www.google.pl*.

Limity dla zawartości zmiennych w zapytaniu

Mając na uwadze ataki typu *Path Traversal*, *DoS*, *Buffer Overflow*, chcielibyśmy ograniczyć dopuszczalną zawartość zmiennych przechowywanych w tablicach GET, POST czy COOKIE w ramach zapytania HTTP.

Suhosin doskonale do tego się nadaje. Pozwala ograniczać ilość zmiennych w zapytaniu (`suhosin.request.max_vars` dla POST i COOKIE), a także długość ich nazw i wartości. Umożliwia również filtrowanie wartości zmiennych pod kątem ASCIIZ (ciągi znaków zakończone symbolem *%00*), dzięki czemu dużo trudniej oszukać skrypt PHP. Dlaczego? Wstrzykując znak końca łańcucha w środek wyrażenia unieważniamy (*komentujemy*) dalszą część wyrażenia.

Chcąc uniemożliwić ustawienie długiego ciągu znaków jako wartości *Cookie*, możesz użyć dyrektywy:

```
suhosin.cookie.max_value_length = 6
```

Spreparujemy dwa zapytania:

- w pierwszym (Listing 11) *Długość cookie* `'testowe'` wynosi 6 bajtów, stąd zawartość zmiennej `'t'` podanej metodą `GET`. została wyświetlona.
- w drugim (Listing 12) *Długość cookie* `'testowe'` wynosi teraz 7 bajtów, żądanie zostało przejęte przez suhosin i wykonano akcję przekierowania (*kod 302*) na adres *http://www.google.com*.

Warto zwrócić uwagę na pole `Content-Length`, które wynosi 0 – nie został podany żaden content.

Limit na nieskończoną rekurencję

Nawet zaawansowany programista jest w stanie nieświadomie napisać skrypt, który pochłonie każde dostępne zasoby serwera WWW. Często pułapką są skrypty korzystające z rekurencji. Suhosin potrafi radzić sobie w takich przypadkach przy pomocy dyrektywy `suhosin.executor.max_depth`. Domyślnie limit ustawiony jest na 0, a tym samym jest wyłączony. Ustawmy go na rozsądną wartość:

```
suhosin.executor.max_depth = 3
```

Napiszmy prosty skrypt korzystający z rekurencji (Listing 13).

Odwołajmy się do niego (Listing 14), upewnijmy się, czy na pewno jest to sprawka Suhosina (Listing 15)

W ten sposób poskromiony skrypt (np. z nieskończoną pętlą rekurencji) nie zrobi krzywdy serwerowi WWW.

Limit rezerwacji pamięci

Zdarza się tak, że część serwisu/aplikacji WWW ma większe potrzeby na użycie pamięci od całej reszty. W *php.ini* ustawiamy `memory_limit = 16M`, ale pozwalamy programistom na użycie funkcji, która może w sposób dynamiczny zmienić ten limit. Umożliwia to funkcja `ini_set()`. Jeżeli jednak obawiamy się o to, że programista będzie zbyt rozrzutny, możemy ustawić twardy limit zużycia pamięci bez względu na to, jak limit zostanie zmieniony przez funkcje `ini_set()`. Na pomoc przybywa dyrektywa `suhosin.memory_limit`.

Spójrzmy, co się stanie, jeśli przy ustawieniu `suhosin.memory_limit = 20M` spróbujemy wyjść poza wyobrażenia administratora o tym, ile możemy zażądać pamięci.

Ilość przydzielonej pamięci (patrz Listing 16) nie zmieniła się z uwagi na zakończone niepowodzeniem wywołanie `ini_set()`. Potwierdzenie w logach (patrz Listing 17.).

Restrykcyjne załączanie plików z innych usług HTTP

Starym, ale często wciąż wykonalnym atakiem jest wymuszenie, by serwis WWW

Listing 14. *Odwołanie do skryptu korzystającego z rekurencji*

```
rez@dojo-labs:~/tmp$ printf 'GET /re.php HTTP/1.0\r\n\r\n' | nc suhosin-test 80
HTTP/1.0 200 OK
Connection: close
X-Powered-By: PHP/5.2.5
Content-type: text/html
Content-Length: 22
Date: Fri, 25 Jan 2008 02:11:15 GMT
Server: lighttpd/1.4.18

Wywolanie 1 Wywolanie 2
```

Listing 15. *Fragment logów Suhosin*

```
Jan 25 03:11:15 dojo-labs suhosin[9530]: ALERT - maximum execution depth reached
                    - script terminated (attacker '127.0.0.1', file '/var/www/t/
                    re.php', line 6)
```

Listing 16. *Zmiana limitu rozmiaru pamięci*

```
<?php
ini_set('memory_limit', '250M');
echo ini_get('memory_limit');
?>
```

Listing 17. *Fragment logów Suhosin – przekroczenie limitu pamięci Suhosin*

```
Jan 25 03:25:04 dojo-labs suhosin[9716]: ALERT - script tried to increase memory_limit
                    to 262144000 bytes which is above the allowed value (attacker
                    '127.0.0.1', file '/var/www/t/1.php', line 2)
```

Listing 18. *Załączenie pliku z innego serwera WWW*

```
<?php
$t="http://localhost/index.php";
include $t;
?>
```

Listing 19. *Załączenie zawartości pliku /etc/passwd*

```
<?php
include "/etc/passwd";
?>
```

załączył kod PHP pochodzący z innego serwera WWW (najczęściej specjalnie spreparowany kod przez atakującego). Spójrzmy przykładowo na skrypt załączający kod z innego serwera WWW (Listing 18) o zawartości takiej, jak na Listingu 19.

Pozwoli on na wyświetlenie zawartości pliku */etc/passwd*. Jeżeli tylko mamy możliwość manipulacji zawartością kodu, który jest załączany przez serwer WWW, to ograniczeni jesteśmy jedynie uprawnieniami, możliwościami ich eskalacji i wyobraźnią.

Domyślnie Suhosin nie pozwala na takie załączanie plików za pomocą protokołów *http://*, *ftp://* czy *php://*. Robi to zdecydowanie lepiej od dyrektyw wbudowanych w PHP, ponieważ te ostatnie można obejść w specyficznych sytuacjach. Jeżeli obawiasz się, że kod serwisu, który utrzymujesz, korzysta z podobnego rodzaju załączania plików, to nic straconego. Za pomocą suhosin.executor.include.whitelist można zdefiniować adresy, z których można wykonywać takie akcje.

To nie koniec możliwości Suhosin extension. Do ciekawszych należą na pewno:

- obrona przed atakiem HTTP Response Splitting,
- eksperymentalne wsparcie dla filtrowania SQL Injection,
- włączenie/wyłączenie możliwości skorzystania z funkcji `eval()`,
- szyfrowanie cookies,
- zablokowanie `"/e"` w funkcji `preg_replace()` – czyli usunięcie furtki do odpowiednika funkcji `eval()` w funkcji `preg_replace()`,
- zaawansowana konfiguracja logowania,
- i dużo,dużo więcej...

Inne zalety Suhosin

Pomimo tak okazałej listy możliwości, Suhosin posiada także inne, mniej oczywiste zalety. Czasem, ale bardzo rzadko, zdarza się, że programiści czy administratorzy zgłaszają problem typu: *Nie działa, od kiedy zainstalowałem Suhosin*. Szczerze mówiąc, prócz trafienia na błędy w samym Suhosin, które należy jak najszybciej zgłosić do autora (o nim wspomnę później), taka sytuacja może się pojawić. Nie dlatego, że Suhosin tak wpływa na pracę silnika interpretera PHP lub go destabilizuje, ale dlatego, że aplikacja jest źle napisana. Dzięki ochronie struktur danych Suhosin może blokować i informować o tym, że jedno z rozszerzeń dołożonych do PHP posiada błędny kod, nadpisujący wartości, których nie powinno. Na czas przystosowywania konfiguracji Suhosina można ustawić tzw. simulation mode za pomocą dyrektywy `suhosin.simulation = On` i, testując aplikację albo serwis WWW, obserwować logi PHP. W tym trybie pracy wszystkie akcje są logowane, ale nie podejmowane realnie.

Wydajność

Tyle zalet, ale ile to kosztuje? Steffan Esser twierdzi, że niewiele. Według jego testów z użyciem skryptu *bench.php*, pochodzącego z rezpozytorium CVS PHP, jest to narzut ‹ 9% w porównaniu do PHP bez *Suhosin Patch* i *Suhosin Extension*. W pliku *bench.php* znajdziemy serie testów przy użyciu rekurencji, wyliczania hash'y, ciągów liczb (np. *Fibonacciego*), operacji na dużych tablicach. Dla serwisów WWW narzut będzie mniejszy, gdyż nie składają się one w przeważającej mierze z takich funkcji. Wyjątkiem będą serwisy, które udostępniają np. API do obliczania wcześniej wymienionych wartości (lub podobnych) i – czasem – zdolni do wszystkiego programiści PHP. Te kilka procent to kropelka w morzu możliwości CPU(s), a liczba uzyskanych w zamian korzyści jest imponująca.

Wady (o ile istnieją)

Czas na wady. Trzeba nałożyć łatkę, przekompilować samodzielnie PHP, skompilować rozszerzenie, załadować je, pamiętać o tym, że załadowany jest Suhosin i wiedzieć, jaką politykę ustawiliśmy jako reakcję na naruszenie reguł naszego Anioła Stróża PHP. Tylko czy to są wady? Moim zdaniem, raczej konsekwencje stosowania bardzo rozsądnego rozwiązania podnoszącego poziom bezpieczeństwa serwera WWW i samej aplikacji. Pamiętać należy o tym, że Suhosin to nie panaceum na problem serwisów WWW podatnych na ataki. Nie mamy tutaj możliwości filtrowania wszystkich danych wprowadzanych przez użytkownika. Możemy np. jedynie ograniczyć długość zmiennej przekazywanej metodą GET do 64 znaków, ale nie zatrzymamy w ten sposób próby wstrzyknięcia XSS'a `"<script>alert(666);</script>"`. Mieć świadomość możliwości rozwiązania to podstawa.

Podsumowanie

Suhosin to jedno z ciekawszych i skuteczniejszych rozwiązań podnoszących poziom bezpieczeństwa aplikacji PHP, a także stabilność systemu, który je hostuje. Już domyślna konfiguracja jest bardzo rozsądna. Rozwiązanie to napisał, rozwija i utrzymuje Stefan Esser, jeden z członków zespołu zajmującego się bezpieczeństwem w ramach projektu PHP. Stefan aktualnie nie jest już członkiem tego zespołu, ale wciąż propaguje bezpieczne PHP. Uskutecznia ten proces, prowadząc wcześniej projekt *Hardened PHP*, a teraz Suhosin, który przejął funkcjonalność z projektu *utwardzonego* PHP i jest dalej rozwijany. Szczególnie należy rozważyć zastosowanie Suhosina w przypadku, gdy administrator i/lub specjalista ds. bezpieczeństwa nie ma wpływu na jakość kodu aplikacji czy serwisu PHP. Co bardzo ważne – nie zapominajmy, że *Anioł Stróż* to tylko dodatek zwiększający szanse w nieustannie trwającej walce z napastnikami. Należy odpowiednio konfigurować całe środowisko – na ile pozwalają opcje konfiguracji i zdrowy rozsądek.

W Sieci

- *http://www.hardened-php.net/suhosin/index.html*,
- *http://www.hardened-php.net/stefan_esser.24.html*,
- *http://www.hardened-php.net/suhosin/a_feature_list:realpath.html*,
- *http://cvs.php.net/viewvc.cgi/ZendEngine2/bench.php?content-type=text%2Fplain&view=co*,
- *http://www.owasp.org*,
- *http://www.owasp.org/index.php/Category:Attack*.

Przemysław Skowron
Autor ma 24 lata. Jest specjalistą ds. bezpieczeństwa teleinformatycznego w jednym z największych portali w Polsce. Czynny aktywista i członek organizacji OWASP.
Kontakt z autorem: *przemyslaw.skowron@gmail.com*

ROADSHOW

28–29 Maja 2008

Warszawa | Hotel Courtyard by Marriott

NETWORK GigaCon™

FORUM nowoczesnych technologii i rozwiązań teleinformatycznych

Szukasz nowoczesnych i tanich rozwiązań sieciowych i telekomunikacyjnych?

Na bezpłatnej konferencji **Network GigaCon 2008** będziesz miał okazję zobaczyć najlepsze z nich! Jeżeli poszukujesz elementów infrastruktury sieciowej, wydajnych rozwiązań dla Internetu, łączy do transmisji danych, nowych systemów niezawodności i zabezpieczania sieci, a poza tym interesują Cię nowoczesne technologie w telefonii – zgłoś się już dziś na naszą konferencję!

To już **dziewiąte doroczne spotkanie** branży sieciowej. Odbędzie się 28 i 29 maja 2008 roku w Hotelu Courtyard by Marriott w Warszawie. Wstęp jest bezpłatny.

Tematyka sesji wykładowych;

- Infrastruktura sieciowa
- Niezawodność sieci
- Bezpieczeństwo sieci
- Wydajne rozwiązania dla Internetu
- Usługi transmisji danych
- Integracja usług teleinformatycznych
- Telefonia IP
- IP TV

WSTĘP **BEZPŁATNY**

Informacje i rejestracja
www.network.gigacon.org

Kontakt
e-mail: kamila.tarlowska@software.com.pl
tel. 022 427 36 47
fax. 022 244 24 59

STANISŁAW JAGIELSKI

VTL remedium na taśmowe kłopoty

Stopień trudności

Systemy informatyczne składają się obecnie z wielu warstw: większość z nich wpisuje się w funkcje, które ma zapewnić system IT w ujęciu ITIL. Ważnym obszarem ITIL jest zapewnienie ciągłości działania systemu IT, m. in. poprzez dostarczanie infrastruktury i narzędzi umożliwiających wykonywanie kopii zapasowych danych.

Z ARTYKUŁU DOWIESZ SIĘ

artykuł przedstawia zagadnienia związane z nowymi trendami w technice,

tworzenia kopi zapasowych, skupiając się na zastosowaniach wirtualnych bibliotek taśmowych (VTL),

opisano pokrótce przyczyny zainteresowania rozwiązaniami VTL i ich główne funkcje.

przedstawione zostaly spodziewane kierunki rozwoju, tej techniki.

CO POWINIENEŚ WIEDZIEĆ

znać podstawowe zagadnienia związane z systemami kopi zapasowych,

posiadać ogólną wiedzę na temat napędów i bibliotek taśmowych,

znać zagadnienia kompresji danych.

Kopie te są potrzebne zarówno z punktu widzenia operacyjnego, jak i coraz częściej w związku z wymaganiami formalnymi dotyczącymi przechowywania i dostępności do danych, które to wymagania są w pewnych przypadkach (bankowość, ubezpieczenia itp.) niezwykle wysokie. W związku z powyższym również wymagania co do infrastruktury kopii zapasowych stają się coraz bardziej złożone. Tradycyjnie głównym elementem tej infrastruktury jest biblioteka taśmowa składająca się zazwyczaj z wielu napędów taśmowych, robotyki oraz zestawu wolumenów taśmowych obsługiwanych w obrębie biblioteki. Podstawowe parametry takiej biblioteki to wydajność zapisu i odczytu skorelowana z liczbą napędów taśmowych oraz ich technologią, pojemność urządzenia ograniczona liczbą półek na wolumeny i ewentualnie możliwość zastosowania biblioteki w różnej topologii – obecnie zazwyczaj SAN, ale również DAS. Z jakąkolwiek biblioteką jednak nie mielibyśmy do czynienia, to głównym problemem związanym z jej używaniem są napędy taśmowe, a właściwie ich zawodność wynikająca z wysokiej złożoności mechanicznej takiego urządzenia. W zależności od źródeł mówi się o tym, że ta wada bibliotek taśmowych jest odpowiedzialna za stosunkowo wysoką – od kilku do kilkunastu procent – zawodność systemów kopii zapasowych budowanych w oparciu o urządzenia taśmowe.

Jak poprawić skuteczność systemów kopii zapasowych?

Jak w każdym systemie, tak i w systemie kopii zapasowych najwyższy wzrost jakości otrzymujemy wtedy, gdy poprawimy jakość najbardziej zawodnego elementu (lub wyeliminujemy ten element z systemu); w przypadku systemu kopii ten element to napędy taśmowe. Od dawna już stosuje się w miejsce napędów taśmowych systemy dyskowe, tj. rozwiązania, gdzie kopia zapasowa zapisywana jest na dysku magnetycznym (a częściej – wolumenie logicznym utworzonym w obrębie macierzy dyskowej), chronionym techniką RAID. Taki sposób wykonywania kopii, nazywany w skrócie D2D (ang. *disk to disk*), szczególnie ostatnio zyskuje na popularności wraz z gwałtowną obniżką cen systemów macierzowych. Macierz dyskowa w porównaniu z biblioteką taśmową jest o rzędy wielkości bardziej niezawodna, bywa również wydajniejsza – raczej jednak pod kątem czasu dostępu do danych niż prędkości transferu (np. napędy LTO-4 mogą zapisywać i odczytywać dane z prędkością powyżej 100 MB/s, stąd biblioteka składająca się z kilku napędów może z łatwością oferować zagregowaną wydajność rzędu TB/s). Wydaje się więc, że prostą sprawą powinno być zastąpienie bibliotek taśmowych przez tanie, pojemne systemy dyskowe. Jednak nie jest to rozwiązanie idealne. Można wymienić co najmniej kilka tego powodów: systemy D2D zazwyczaj korzystają

z systemów plików zdefiniowanych na wolumenach dyskowych – w tym przypadku groźne okazują się normalne zagrożenia czyhające w systemach plików, czyli wirusy, możliwość przypadkowego uszkodzenia systemu plików i inne. Użycie systemów plików implikuje również niemożność współdzielenia wolumenów – tj. dany wolumen (zapasowy) jest dostępny tylko i wyłącznie z poziomu jednego z serwerów mediów, czyli z punktu widzenia systemu kopii zapasowych mamy do czynienia z topologią DAS. Na koniec być może najbardziej ważki argument przemawiający przeciw technice D2D: tam, gdzie system kopii zapasowych używany jest od lat, zazwyczaj wypracowano i dopracowano bardzo efektywny zestaw procedur tworzenia kopii z wykorzystaniem bibliotek taśmowych, a ich zamiana na macierz dyskową i D2D pociągnie za sobą konieczność zmiany praktycznie wszystkich w/w procedur. Każdy, kto choć raz prowadził projekt informatyczny, doskonale zdaje sobie sprawę z tego, że to właśnie modyfikacja procedur jest najtrudniejszym elementem wprowadzania jakichkolwiek zmian w infrastrukturze IT.

Jak zjeść ciastko i nadal je mieć?

Z powyższych rozważań wynika, że najlepiej byłoby posiadać repozytorium kopii zapasowych, które zachowywałoby się jak biblioteka taśmowa, ale równocześnie nie posiadało negatywnych cech napędów taśmowych. Lista życzeń co do takiego urządzenia jest bardzo prosta: dane składowane na systemie dyskowym chronionym techniką RAID, wysoka wydajność zarówno transferu, jak i dostępu do danych, współdzielenie wolumenów zapasowych w sieci SAN, duża pojemność przy atrakcyjnej cenie, odporność na wirusy oraz przypadkowe uszkodzenia struktury i danych ze strony użytkownika.

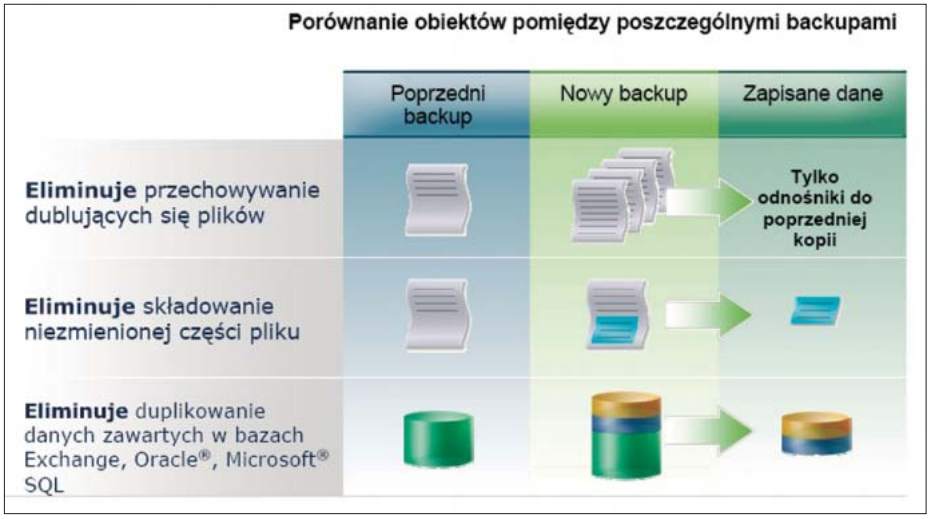
Ponadto, jak każde nowe urządzenie, powinno ono mieć do zaoferowania funkcjonalności wykraczające poza zwykłą syntezę biblioteki taśmowej i prostego systemu dyskowego – wydaje się, że to właśnie takie funkcjonalności powinny być wyróżnikiem rozwiązań konkretnych producentów. Takie repozytorium to VTL – *Virtual Tape Library* (wirtualna biblioteka taśmowa), czyli urządzenie zbudowane w oparciu o macierz dyskową, lecz (niemal) w pełni emulujące tradycyjną bibliotekę taśmową.

Czy zauważyłeś zmianę?

Główna funkcjonalność VTL to oczywiście emulacja biblioteki taśmowej wraz z jej wszystkimi elementami. Jest to cecha na tyle silna, że niektórzy dostawcy rozwiązań ograniczają się do dostarczenia oprogramowania emulującego bibliotekę taśmową, resztę – czyli integrację z systemem dyskowym – pozostawiając użytkownikom końcowym. Emulacja powinna być na tyle dobra, aby użytkownik nie zauważał różnicy – poza oczywiście wzrostem odporności na awarie i szybkością działania, np. załadowania wirtualnej kasety do wirtualnego napędu taśmowego. Przede wszystkim jednak różnicy nie powinien zauważyć system kopii zapasowych – w tym system operacyjny wraz ze sterownikami dla odpowiedniego typu emulowanego urządzenia taśmowego. (Pewne oprogramowanie systemu kopii zapasowych wiodącego producenta wykazywało błędy we współpracy z VTL – uznawało, że czas, jaki upłynął pomiędzy żądaniem zamontowania wolumenu, a zgłoszeniem jego gotowości jest zbyt krótki i zgłaszało błąd biblioteki taśmowej.) Ogólnie rzecz biorąc, im więcej typów napędów (a także modeli bibliotek taśmowych) dana VTL może emulować, tym lepiej. W praktyce jednak wystarcza, że emulowane są najpopularniejsze typy napędów taśmowych: dla systemów otwartych LTO i DLT/SDLT, a dla systemów zaawansowanych – napędy serii 3590/3592 i T9000. Jeśli zaś chodzi o modele bibliotek, to warto zwrócić uwagę, czy dana VTL potrafi emulować tylko biblioteki sterowane komendami SCSI, czy również ACLS. Dla konkretnego zastosowania zasadnicze znaczenie może mieć obecna infrastruktura repozytorium kopii albo też specyficzny model licencjonowania użycia bibliotek taśmowych przez dany system kopii zapasowych.

Czy warto naśladować dokładnie?

Wierne naśladownictwo jest dobre, ale ulepszenia są zawsze mile widziane – stąd pewne elementy biblioteki wirtualnej mogą (a nawet powinny) różnić się od analogicznych elementów biblioteki



Rysunek 1. *Korzyści płynące ze stosowania de-duplikacji danych*

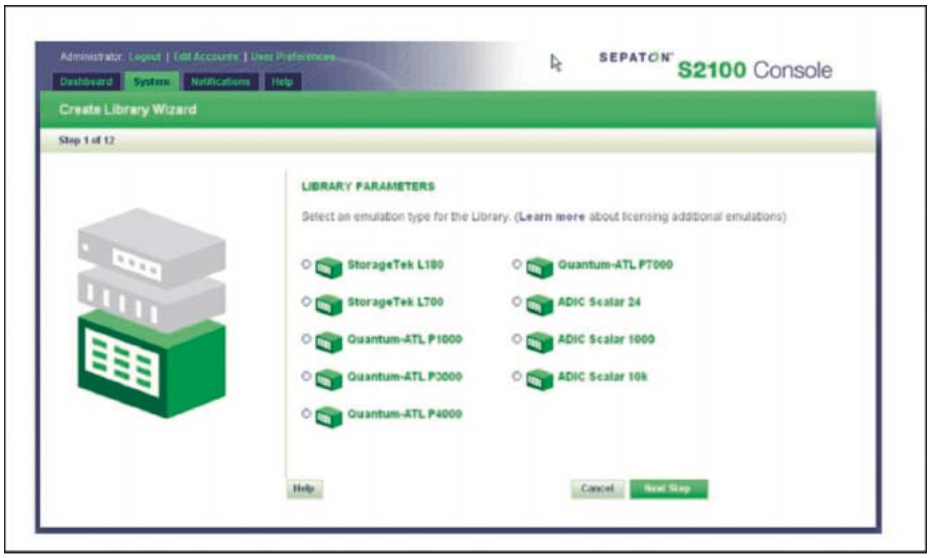
Rysunek 2. *Wirtualna biblioteka taśmowa Sepaton S2100-DS2*

rzeczywistej. Dobrym tego przykładem jest wolumen taśmowy, czyli praktycznie najmniejszy obiekt repozytorium widziany przez system kopii: w większości przypadków zarządzanie wolumenami jest łatwiejsze, gdy mamy do czynienia z większą ilością mniejszych wolumenów, niż w sytuacji odwrotnej. Z drugiej strony, niektóre systemy kopii zapasowych licencjonowane są ze względu na liczbę półek/taśm w bibliotece: w tym przypadku ze względów kosztowych preferowana jest ta właśnie odwrotna sytuacja. Większość VTL pozwala zdefiniować wirtualne wolumeny taśmowe o dowolnej wielkości, a niektóre z nich potrafią określić maksymalną pojemność kasety i alokować żądaną przestrzeń w miarę potrzeb. Takie podejście pozwala systemowi dynamicznie zmieniać swoje parametry, a stąd już krok do sprzedaży pojemności VTL w modelu *pojemność na żądanie*, modelu obecnego już w wielu urządzeniach składowania danych. Być może jeszcze bardziej istotną różnicą pomiędzy rzeczywistą a wirtualną biblioteką taśmową jest reakcja napędu taśmowego na zmniejszający się strumień danych, co dla realnego napędu taśmowego jest jednym z większych problemów. Powoduje to zmniejszenie wydajności zapisu, ale przede wszystkim prowadzi do szybszego zużywania się zarówno samego urządzenia, jak i zapisywanego wolumenu taśmowego. Napędy taśmowe najnowszych technologii do wyeliminowania tego efektu wymagają strumienia danych rzędu dziesiątek MB/s, natomiast wirtualny napęd taśmowy jest zupełnie niewrażliwy na zmiany strumienia danych.

Jak wynieść wirtualną taśmę?

Jedną z podstawowych procedur dla systemu kopii zapasowych jest przechowywanie duplikatów kopii (zazwyczaj zwanych klonami) poza lokalizacją, w której znajduje się system chroniony. Oryginalne kopie są przechowywane w bibliotece, aby umożliwić szybkie odtworzenie danych w przypadku awarii, klony zaś wynoszone są do innej lokalizacji w celu umożliwienia odtworzenia danych na wypadek katastrofy. Tradycyjne biblioteki taśmowe radzą sobie z tym bardzo dobrze: pod kontrolą oprogramowania systemu kopii tworzony jest klon, taśma klonowa podawana jest do portu wejścia/wyjścia biblioteki, zabierana stamtąd i przewożona w odpowiednie miejsce. Jak jednak uzyskać ten sam efekt w przypadku VTL? Nie da się przecież wyjąć wolumenu logicznego z macierzy dyskowej. Niestety, nie wszyscy producenci bibliotek wirtualnych poradzili sobie z tym problemem. Ci, którzy go rozwiązali, stosują różne techniki, a pierwsza z nich polega na skorzystaniu z… rzeczywistej biblioteki taśmowej. Metoda ta ma dwie odmiany. Pierwsza wykorzystuje fakt, że oprogramowanie wirtualizujące współdziała z biblioteką dopiętą bezpośrednio do VTL i na żądanie lub według zdefiniowanych polis kopiuje lub przenosi dane (kopie) z wolumenu wirtualnego na wolumen w rzeczywistej bibliotece. Ta odmiana ma kilka ograniczeń: po pierwsze, aby taka operacja była możliwa, wolumeny wirtualne i rzeczywiste muszą mieć takie same rozmiary – nie możemy skorzystać z kompresji, ani dowolnie zdefiniować pojemności wolumenu wirtualnego. Po drugie, operacja wykonywana jest na poziomie VTL, a w związku z tym nie jest śledzona przez oprogramowanie systemu kopii zapasowych. Wybrane modele VTL potrafią jednak obejść to ograniczenie, umożliwiając uzyskanie rzeczywistych wolumenów w żaden sposób nie różniących się od takich wolumenów rzeczywistych, które byłyby zapisane bezpośrednio przez system kopii zapasowych. Identyczność jest zapewniania m. in. na poziomie użytej technologii zapisu (np. LTO-3) oraz formatu zapisu danych (np. tar, OTF itp.), co pozwala na użycie tak przygotowanych wolumenów bezpośrednio do odtwarzania danych z pominięciem VTL. W przeciwnym bowiem wypadku, odtwarzanie danych przeniesionych z wolumenów wirtualnych na rzeczywiste wymaga obecności VTL: albo po to, aby dane wczytać do VTL i udostępnić systemowi kopii, albo aby udostępnić je systemowi bez wczytywania ich do VTL, ale za jej pośrednictwem.

Druga z technik wykorzystujących rzeczywistą bibliotekę opiera się na integracji oprogramowania VTL z oprogramowaniem systemu kopii zapasowych – w ten sposób, że oprogramowanie VTL zawiera w sobie oprogramowanie serwera wolumenów danego systemu kopii, i to ten serwer wolumenów działający na VTL wykonuje pod kontrolą oprogramowania systemu kopii operację klonowania. Warto zauważyć, że nie jest to sytuacja tożsama z zastosowaniem VTL i biblioteki rzeczywistej pracujących pod kontrolą zewnętrznego (względem VTL) serwera mediów – w tym przypadku dane muszą przepływać po sieci używanej do tworzenia kopii zapasowych, a sam proces wymaga albo dodatkowego serwera wolumenów, albo dodatkowo obciąża już istniejące. Metoda wykorzystująca bibliotekę rzeczywistą ma jednak zawsze co najmniej jedną sporą wadę – wykorzystuje bibliotekę rzeczywistą, czyli urządzenie, którego chcielibyśmy się pozbyć używając VTL.



Rysunek 3. *Łatwy sposób definiowania emulowanych bibliotek taśmowych*

Wracając do listy sposobów wynoszenia wolumenu wirtualnego poza lokalizację systemu chronionego – druga metoda wykorzystuje technikę znaną z rynku macierzy dyskowych: replikację danych z użyciem sieci IP. Na potrzeby tej metody stosowane są zarówno techniki kompresji, jak i szyfrowania replikowanych danych – tak, aby transfer był bezpieczny i wydajny. W mechanizmie replikacji kopii danych pomiędzy dwoma VTL również możemy znaleźć zastosowanie obu technik opisanych dla pierwszej metody, czyli bez i z serwerem wolumenów danego systemu kopii. Replikacja z wykorzystaniem sieci IP ma dwie zasadnicze zalety: nie używamy w żaden sposób biblioteki rzeczywistej i – co równie ważne – nie przewozimy taśm, w związku z czym nie ma możliwości ich zagubienia lub kradzieży.

Jako pewną odmianę tej metody wprowadzono możliwość albo wykonania faktycznej replikacji (czyli utworzenia drugiej kopii danych), albo przeprowadzenia wirtualnego wyniesienia wolumenu z lokalizacji pierwotnej (czyli replikacji zawartości wolumenów do drugiej lokalizacji i usunięcia oryginalnego wolumenu). Ta ostatnia opcja najbardziej przypomina wynoszenie wolumenów taśmowych z rzeczywistej biblioteki taśmowej, lecz nie posiada żadnej wady rozwiązania rzeczywistego.

Czy to jeszcze kompresja?

Na początku zdefiniowaliśmy podstawowe parametry biblioteki taśmowej, m. in. pojemność – rozumianą jako iloczyn pojemności pojedynczej kasety i liczby kaset możliwych do przechowania w bibliotece. W sposób oczywisty pojemność ta będzie się zwiększała, jeśli urządzenie taśmowe będzie kompresowało zapisywane dane. W praktyce dla napędów LTO w zależności od prędkości dostarczania danych i ich typu uzyskuje się kompresję w zakresie od 1:1,5 do 1:3. Również dla większości VTL umożliwiono kompresję zapisywanych danych, przy czym różni producenci podeszli do tego zagadnienia w odmienny sposób. Stosunkowo proste rozwiązanie polega na zastosowaniu mniej lub bardziej standardowych algorytmów kompresji, i to albo realizowanych programowo, albo przy użyciu rozwiązań sprzętowych zintegrowanych w ścieżce przepływu danych w obrębie VTL. Rozwiązania tego typu gwarantują stopień kompresji podobny do uzyskiwanych w systemach tradycyjnych – aby uzyskać lepsze rezultaty, trzeba było wymyślić coś nowego. Wymyślono więc metodę, która zresztą jest stosowana nie tylko w technologii VTL, a mianowicie de-duplikację, czyli metodę bazującą na prostej prawdzie – większość danych które teraz właśnie kopiujesz, skopiowałeś już wcześniej. Strumień danych przesyłany jest do VTL, tam jest zapisywany, podlega oglądowi i jeśli jego całość lub część zostały już uprzednio zapisane na wolumenach wirtualnych, to duplikat jest usuwany, a zachowywana jest tylko informacja o jego istnieniu. Dużo oczywiście zależy od sposobu oglądu – czy jest on realizowany sprzętowo, czy programowo, czy zastosowano metody porównywania sum kontrolnych, czy też porównanie odbywa się ze świadomością typu zawartości. Szczególnie ta ostatnia metoda jest wysoce efektywna, jednak zazwyczaj, aby zapewnić odpowiednią wydajność zapisu, de-duplikacja jest w takim przypadku wykonywana w VTL już po skończeniu sesji nagrywania kopii przez system.

W ten sposób zrzut stu plików calc.exe jest zapisany tylko raz, a z piętnastu kopii różnych wersji tego artykułu są zapisywane tylko fragmenty je różniące, co w prosty sposób prowadzi do kompresji rzędu 1:25, a nawet większej. Czynnik takiego rzędu powoduje znaczne powiększenie pojemności logicznej VTL, a co za tym idzie – obniżenie ceny składowania jednostki danych (MB, GB) do tego stopnia, że również pod tym względem VTL zaczyna być konkurencyjna

Rysunek 4. *Wirtualna Biblioteka Taśmowa Sepaton S2100-ES2*

względem biblioteki rzeczywistej, nawet biorąc pod uwagę cenę ewentualnej licencji dla de-duplikacji.

Ale czy to jest bezpieczne?

Wraz ze wzrostem ilości przechowywanych danych wymagania co do pojemności VTL rosną, stąd omówiona powyżej technika de-duplikacji; jednak innym, równie ważnym zagadnieniem jest ochrona danych w rozumieniu ochrony dostępu do nich oraz ochrony ich poufności. Pierwsze zagadnienie zyskuje już na samej logice dostępu do zasobów taśmowych – wirtualnych lub rzeczywistych – mianowicie, odwrotnie niż w przypadku systemu typu *kopia na dysk*, wykorzystującego standardowe systemy plików, dostęp do danych zgromadzonych na wolumenach taśmowych jest możliwy praktycznie tylko poprzez oprogramowanie systemu kopii. Ochrona dostępu jest zazwyczaj zapewniana właśnie przez mechanizmy tegoż systemu: autentykację, autoryzację oraz – w niektórych przypadkach – specyficzny format zapisu danych. Poufność danych wymaga również zabezpieczenia nie tylko dostępu do nich, ale też możliwości (a właściwie niemożliwości) ich odczytania przez osoby niepowołane: w tym przypadku najczęściej stosuje się szyfrowanie składowanych danych, a także odpowiedni sposób wirtualnego niszczenia danych po ich logicznym usunięciu z wolumenów. Przykładowo, jeden z wytwórców VTL stosuje technikę trzykrotnego nadpisywania wirtualnych wolumenów różnym ciągiem bitów, co przewyższa formalne wymagania Departamentu Obrony USA względem niszczenia danych poufnych, tajnych i ściśle tajnych.

Czy będę miał zawsze dostęp do (kopii) danych?

Jak pamiętamy, głównym powodem chęci użycia VTL było zastąpienie wysoce zawodnego elementu systemu kopii zapasowych – rzeczywistej biblioteki taśmowej – elementem możliwie niezawodnym. Zachodzi więc pytanie, czy rzeczywiście VTL spełnia nasze wymagania? Wiemy już, że kopie danych przechowywane są w VTL na macierzy dyskowej, ale jednocześnie macierz ta jest obsługiwana przez silnik VTL, tj. serwer wraz z oprogramowaniem, i o ile nie dziwi nas, że do budowy bibliotek wirtualnych stosuje się wyłącznie macierze bez pojedynczego punktu awarii, to jak ma się sprawa z silnikiem? Większość dostawców stosuje rozwiązania z wieloma silnikami, przede wszystkim po to, aby podwyższyć ogólną wydajność systemu. W niektórych modelach posiadających wiele silników możliwe jest również wykorzystanie silników w trybie *aktywny/aktywny*. W takim przypadku w trakcie normalnej pracy dwa lub więcej silników dzieli pomiędzy siebie obciążenie, natomiast gdy jeden z nich przestanie poprawnie funkcjonować całość obciążenia automatycznie jest przenoszona na inny silnik, dzięki czemu proces tworzenia lub odtwarzania kopii zapasowych nie jest zakłócany wcale lub tylko minimalnie. Zaawansowane modele takiego typu bibliotek wirtualnych potrafią powrócić do normalnego trybu pracy po ustaniu przyczyn niepoprawnego funkcjonowania silnika.

Zielona czy niebieska?

Jeśli już przekonaliśmy się, że VTL to remedium na nasze taśmowe kłopoty, to warto zastanowić się, jaką VTL powinniśmy kupić, przy czym kolor urządzenia powinien mieć oczywiście drugorzędne znaczenie. Dwie główne kategorie kryteriów: finansowe i techniczne trzeba – jak zwykle – bardzo dokładnie skonfrontować ze swoimi możliwościami i wymaganiami. Już sama decyzja o zakupie VTL da się przeliczyć na konkretne oszczędności, także inwestycyjne, ale przede wszystkim operacyjne: niektórzy z dostawców mówią o oszczędnościach rzędu dziesiątków tysięcy dolarów. Również odpowiedni dla naszego środowiska dobór modelu i wyposażenia VTL może gwarantować większe oszczędności. Trzeba tu określić, czy nasze dane i sposób ich kopiowania nadają się do de-duplikacji, czy taniej jest replikować dane, czy przewozić kasety? Prawdopodobnie więcej pytań należy sobie zadać na temat zgodności wybranego VTL z istniejącym lub budowanym systemem kopii zapasowych, a także możliwości rozwoju zarówno ilościowego (zazwyczaj kwestia pojemności) jak i funkcjonalnego danego urządzenia. Duże znaczenie dla podjęcia właściwej decyzji powinno również mieć przewidzenie możliwych dróg rozwoju systemu, w tym również tych związanych z wymaganiami formalnymi. Warto pamiętać i o tym, że VTL to nie tylko po prostu lepsza biblioteka taśmowa, ale także urządzenie, które pozwala nam czasem diametralnie zmienić architekturę naszego systemu kopii; prosty projekt zamiany biblioteki taśmowej na nową może zyskać całkiem inny wymiar.

Podsumowanie

Na pierwszy rzut oka wydaje się, że uzyskaliśmy już to, o czym marzy każdy administrator systemu kopii zapasowych – pozbyliśmy się najbardziej zawodnego ogniwa systemu, wymieniając je na bardziej zaawansowane, o lepszych parametrach i większej niezawodności. Niemniej jednak nowe rozwiązanie, jak zawsze, generuje nowe pytania i wątpliwości. Ostatnio na przykład pojawiły się pytania, czy składowanie danych w formie de-duplikowanej jest zgodne z formalnym wymaganiem składowania ich w postaci niezmienionej. Następna kwestia to rozważenie, czy rozwój VTL nie powinien prowadzić do powstania zintegrowanego urządzenia, będącego w gruncie rzeczy całościowym systemem kopii zapasowych – na rynku już są obecne rozwiązania tego typu. Jeszcze jedna szansa dla VTL to powstanie na jego bazie urządzenia będącego nie tylko repozytorium kopii zapasowych, ale ogólnie repozytorium danych. Toczą się już prace, które mają doprowadzić do powstania VTL, która – mając świadomość zawartości danych – będzie je mogła organizować w swoisty system plików i udostępniać użytkownikom, np. jako usługę Web. Na rynku wirtualnych bibliotek taśmowych należy się więc chyba spodziewać sporo nowości. Rozwój rynku jest bardzo dynamiczny, a jego postrzeganie bardzo pozytywne – nie tylko przez małe innowacyjne firmy technologiczne, ale także przez gigantów rynku pamięci masowych.

Stanisław Jagielski
Dyrektor Konsultingu i Szkoleń w firmie S4E S.A. Rozwiązaniami pamięci masowych, a przede wszystkim systemami kopi zapasowych zajmuje się od prawie 10 lat. Posiada szerokie doświadczenia zarówno w projektowaniu, jak i wdrażaniu tego typu systemów.
Kontakt z autorem: *Stanislaw.Jagielski@s4e.pl*

PIOTR CICHOCKI

Współczesne rozwiązania wielosilnikowe

Stopień trudności

Zagadnienia związane z bezpieczeństwem systemów komputerowych w przedsiębiorstwach nabrały ogromnego znaczenia w ciągu ostatnich lat. Powodem zaistniałej sytuacji stał się wzrost ilości różnych typów złośliwego oprogramowania oraz metod rozpowszechniania go w sieci Internet.

Dzięki temu wielosilnikowe systemy antywirusowe oraz antyspamowe zaczynają odgrywać na świecie coraz większą rolę w procesie ochrony systemów komputerowych. Niestety, na podstawie obserwacji stwierdzono, iż poziom świadomości dotyczący istnienia takich rozwiązań jest nadal zbyt niski wśród polskiej kadry zarządzającej oraz niewielkiej liczby administratorów. Producenci rozwiązań wielosilnikowych prześcigają się w konstruowaniu nowych, zapewniających dużą elastyczność konfiguracji oraz prostych w administracji systemów zabezpieczeń.

Celem artykułu jest omówienie rozwiązań wielosilnikowych oraz zwrócenie uwagi na podstawowe korzyści płynące z ich zastosowania.

Według informacji FBI Crime and Security Survey z 2006 roku, na 98% przedsiębiorstw posiadających oprogramowanie antywirusowe, 84% zostało zainfekowanych przez wirusy. Co jest przyczyną zaistniałej sytuacji?

Gdy na świecie pojawiły się pierwsze złośliwe programy a zaraz za nimi pierwsze programy antywirusowe, niewiele osób przypuszczało, że rozwój wirusów nastąpi w tak gwałtowny sposób. Pierwsze wirusy, pomimo niewielkich szkód jakie mogły spowodować, wstrząsały opinią publiczną oraz budziły nie tylko niepokój, ale wywoływały sensację a w mediach można było usłyszeć o pierwszym wirusie komputerowym. Czasy kiedy na jeden czy dwa wirusy wystarczał program antywirusowy aktualizowany co kilka tygodni odeszły w niepamięć. Zarówno autorzy szkodliwego oprogramowania, jak również producenci oprogramowania AV, prześcigają się w pomysłach, Ci pierwsi na skuteczny atak, drudzy na skuteczną obronę. W środku tej walki pojawiają się użytkownicy końcowi (administratorzy, użytkownicy zdani na łaskę administratorów, lub też użytkownicy domowi). Bezpieczeństwo posiadanych systemów komputerowych, sieci lokalnych, korporacyjnych czy wreszcie domowych, zależy od świadomych wyborów dokonywanych przez wymienione, przykładowe grupy użytkowników końcowych. Czy użytkownicy końcowi powinni decydować się na uzależnienie od jednego producenta oprogramowania antywirusowego? Jeśli zależy im na elastyczności rozwiązań, niezależności, bezpieczeństwie danych oraz poprawieniu reakcji na zagrożenia to zdecydowanie powinni zastanowić się nad rozwiązaniem opartym na co najmniej kilku silnikach antywirusowych różnych producentów.

Dlaczego wiele silników?

Przede wszystkim stosowanie rozwiązań wielosilnikowych umożliwia korzystanie z wielu szczepionek w momencie wykrycia nowego złośliwego programu. Pomimo, iż producenci oprogramowania antywirusowego oraz antyspamowego zapewniają, że czas reakcji na nowe zagrożenia jest wysoki, w rzeczywistości nie jest tak zawsze. Nie zdarza się, aby jeden producent oprogramowania, za każdym razem

Z ARTYKUŁU DOWIESZ SIĘ

czym są wielosilnikowe rozwiązania antywirusowe oraz antyspamowe,

jakie korzyści płyną z ich zastosowania,

jak walczyć z wyciekiem informacji poprzez wiadomości e-mail,

jak działają niektóre techniki antyspamowe,

jak działają rozwiązania wielosilnikowe,

jak zapewnić bezpieczeństwo zgodnie z normą ISO 27001.

CO POWINIENEŚ WIEDZIEĆ

znać podstawowe zagadnienia związane z bezpieczeństwem informatycznym.

dostarczał jako pierwszy szczepionkę na wszystkie, nowopojawiające się typy zagrożeń. Biorąc pod uwagę fakt, iż wirusy mogą rozprzestrzeniać się z zawrotną prędkością, chociażby drogą poczty elektronicznej, nigdy nie ma pewności, że akurat producent, którego silnik został zastosowany, zareaguje w krótkim czasie na zaistniałe zagrożenie (Tabela. 1). Czas reakcji na zagrożenia rozsyłane w poczcie elektronicznej w podziale na różnych producentów). Problem tzw., wąskiego gardła, czyli uzależnienia od jednego silnika, zostaje wyeliminowany, a szansa na zidentyfikowanie nowego wirusa, w krótkim czasie, znacznie wzrasta w przypadku stosowania wielu silników.

Organizacja *VirusBulletin*, której działalność polega na prowadzeniu analiz programów antywirusowych różnych producentów w oparciu o najbardziej niebezpieczne wirusy oraz publikowaniu raportów, wykazuje, iż nie ma na świecie silnika antywirusowego, który byłby doskonały i przeszedł wszystkie testy z bardzo dobrym wynikiem. Producenci oprogramowania wykorzystują różne algorytmy identyfikujące wirusy. W związku z tym faktem, wirusy znalezione przez poszczególne mechanizmy analizy heurystycznej mogą posiadać różne nazwy. Ponadto w rozwiązaniach wielosilnikowych mogą znajdować się obok siebie silniki producentów, których siedziby zlokalizowane są w bardzo odległych miejscach globu, w różnych strefach czasowych. Dzięki temu wzrasta szansa otrzymania w szybkim czasie odpowiedniej szczepionki i zmniejszenie ryzyka zainfekowania systemów komputerowych. Ponadto rozwiązania wielosilnikowe obejmują wszystkie typy systemów komputerowych. W tradycyjnych rozwiązaniach producenci często wymagają zakupu wersji dedykowanej dla konkretnego systemu komputerowego.

Tabela 1. *Czas reakcji na szkodnika Trojan-Downloader-14439*

Oprogramowanie	Czas reakcji (H:M)
CA eTrust	95:34
Kaspersky	4:27
McAfee	16:11
Microsoft	29:56
NOD32	10:21
Sophos	5:46
Symantec	17:13
Trend Micro	75:44

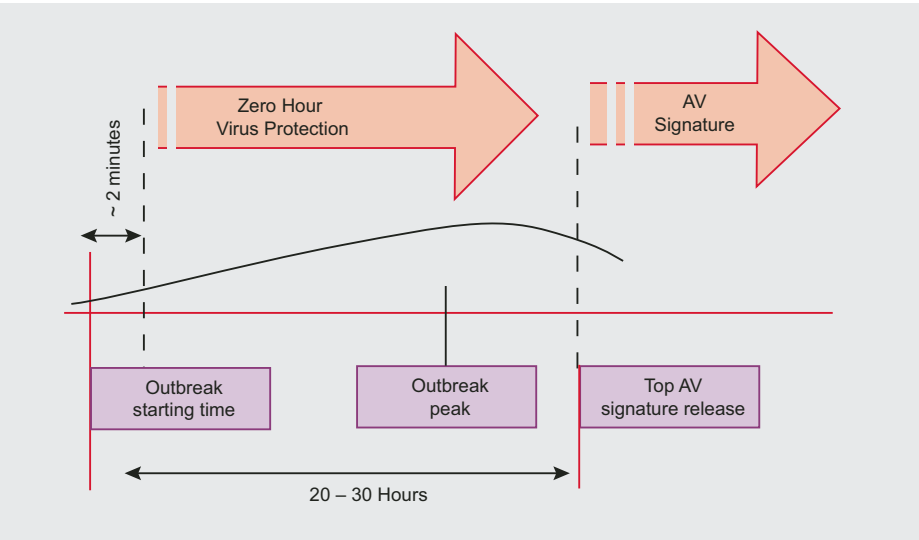
Omawiana sytuacja może być kłopotliwa i nadszarpnąć budżet firmy, która zakupując konkretne oprogramowanie antywirusowe lub też antyspamowe, nie przewidziała, iż zmiana systemu serwerowego lub też klienckiego pociągnie za sobą wykupienie nowej licencji na oprogramowanie chroniące przed zagrożeniami.

Wyciek informacji poprzez pocztę elektroniczną

Według informacji opublikowanych w listopadzie 2007 r. przez firmę SOPHOS, 70% przedsiębiorstw obawia się o przypadkowe wysłanie wiadomości e-mail zawierającej poufne informacje do niewłaściwego adresata. Ponadto według informacji z tego samego źródła wynika, iż ok. 50% pracowników przyznało, iż zdarzyła im się taka sytuacja. Dla firm jest to niewątpliwie problematyczna kwestia, ponieważ pomyłka nawet jednego z pracowników, może doprowadzić do przechwycenia poufnych danych przez konkurencję, co z kolei może wiązać się ze stratami finansowymi lub kompromitacją firmy. Biorąc pod uwagę, iż obecnie w procesie wymiany informacji dużą rolę odgrywa komunikacja za pomocą poczty elektronicznej, rośnie również prawdopodobieństwo omyłkowego przesłania ważnych informacji do nieodpowiedniej osoby. Klienci biznesowi powinni przeciwdziałać temu zagrożeniu wszelkimi dostępnymi metodami. Dlatego też warto rozważyć zastosowanie technologii, która umożliwia w sposób elastyczny tworzenie reguł oraz nakładanie restrykcji na wychodzącą, jak również przychodzącą pocztę elektroniczną.

Techniki antyspamowe

Obecnie w rozwiązaniach wielosilnikowych umożliwiających blokowanie spamu stosowanych jest wiele technik antyspamowych. Omówiono kilka wybranych, najskuteczniejszych według autora artykułu. Pierwszą techniką, która umożliwia osiągnięcie dużej skuteczności w eliminowaniu spamu jest GreyListing, czyli tzw. szare listy. Mechanizm szarych list działa w następujący sposób: po wysłaniu wiadomości e-mail z serwera nadawcy, serwer adresata zwraca błąd tymczasowy, co wiąże się z chwilowym odrzuceniem wiadomości. W przypadku standardowej konfiguracji serwerów pocztowych, serwer nadawcy wysyła e-mail ponownie po ustalonym czasie. Z kolei spamerzy z reguły stosują metodę *wystrzelić i zapomnieć* (ang. *fire and forget*), dlatego ich narzędzia, serwery spamujące nie czekają na odpowiedź od serwera, do którego wysyłają wiadomość, a tym samym nie otrzymują informacji o błędzie. W związku z tym faktem, spam nie jest wysyłany ponownie do tego samego adresata. W uproszczonej technice szarych list rozpoznawanie wiadomości e-mail, które były już raz wysłane



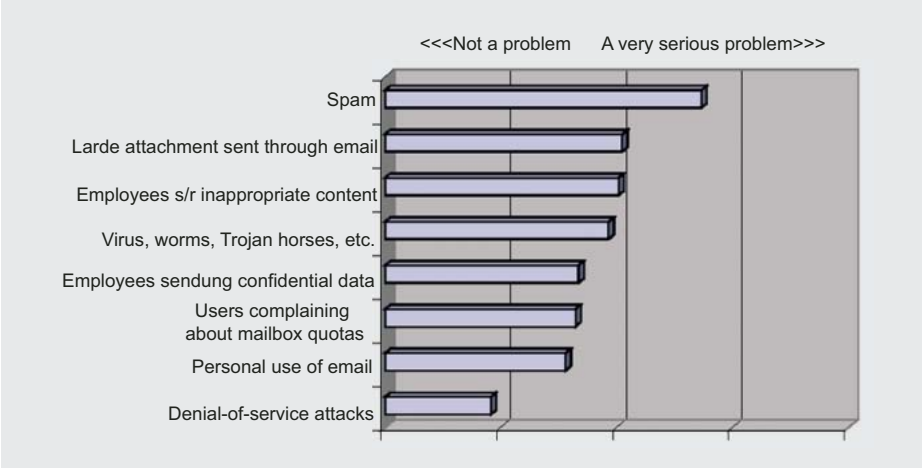
Rysunek 1. *Zero-hour protection – analiza statystyczna*

do serwera adresata, następuję po adresie IP serwera nadawcy. Poza tym zastosowano regułę kilkuminutowego odstępu czasowego przy odbieraniu przez serwer wiadomości pochodzących z tego samego adresu IP. Oznacza to, iż kilka e-maili wysłanych z identycznego adresu IP, jeden po drugim, bez zachowania wymaganej przerwy w czasie, spowoduje odrzucenie ich. W przypadku tej metody mogą nastąpić opóźnienia w dostarczaniu wiadomości e-mail, jednakże zauważono, iż rzadko kiedy jest to dostrzegane przez użytkowników końcowych.

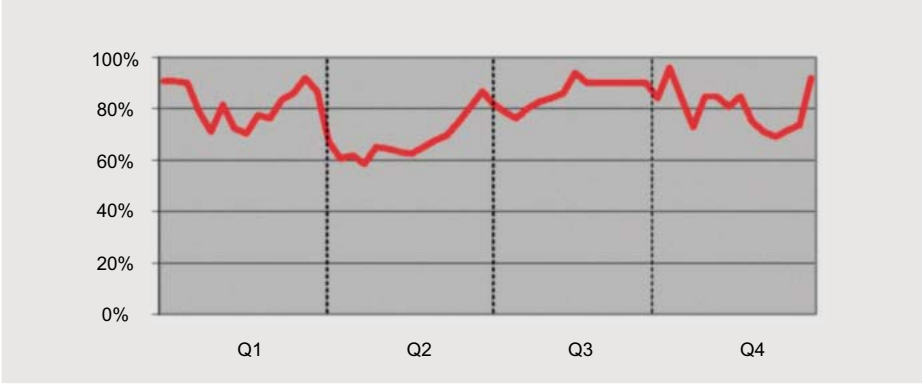
Kolejną techniką, która zasługuje na uwagę jest tzw. filtr Bayesian. Omawiana metoda filtrowania poczty elektronicznej oparta jest na nalizie statystycznej częstotliwości występowania określonych znaków lub wyrazów kluczowych we wszystkich przychodzących wiadomościach e-mail. Poza tym analizie poddawana jest cała treść wiadomości e-mail a skuteczność metody jest niezależna od języka. Przy założeniu, iż w określonym czasie do serwera adresata dotarła pewna ilość wiadomości e-mail charakteryzujących się specyficznym elementem, część z nich została zaklasyfikowana jako spam, a część jako bezpieczne wiadomości. Na podstawie ilości maili z obydwu grup jest dokonywana analiza a następnie dalsze zaklasyfikowanie wiadomości zawierających określone treści. Filtry bayesowskie potrafią wyfiltrować nawet do 99% spamu. Automatycznie dostosowują się do zmian w spamie i potrafią identyfikować spam poprzez analizowanie bezpiecznych wiadomości docierających do określonego adresata. W przypadku tychże filtrów istnieje konieczność dostosowania do własnych potrzeb, aby w przyszłości unikać błędnych klasyfikacji wiadomości e-mail.

Następną metodą antyspamową, która wydaje się być najbardziej wygodna jest zastosowanie heurystyki. Jest to metoda oparta na filtrach, które poszukują pewnych fraz, wyrazów, ciągów znaków pisanych wielkimi literami lub też innych elementów charakterystycznych dla spamu. Filtry heurystyczne umożliwiają wychwycenie spamu nawet w 90% sprawdzonych wiadomości. Wadą dotyczącą tychże filtrów jest fakt, iż zbudowane są na zestawie statycznych reguł. Każda zmiana w technikach spamerskich pociąga za sobą dopisanie nowych reguł. Na szczęście bazy reguł są automatycznie aktualizowane poprzez sieć Internet. Do zalet filtrów heurystycznych można zaliczyć możliwość szybkiego zainstalowania na serwerach pocztowych i natychmiastową gotowość do analizowania przychodzących wiadomości e-mail.

Należy zwrócić również uwagę na metodę antyspamową o nazwie *HashCash*. Jest to technika, która ma gwarantować, iż e-mail, który otrzyma adresat nie jest spamem. Związana jest z tzw. płaceniem za e-mail mocą procesora. W momencie wysyłania listu niezbędne jest wykonanie działania, które wymaga dość dużej pracy procesora. Proces ten jest w zasadzie nieodczuwalny dla nadawcy w przypadku wysyłania małej ilości e-maili. Odbiorca odbiera wiadomość i pochłania to po jego stronie małą ilość zasobów. Jednakże przedmiotowe działanie zostało skonstruowane w taki sposób, że wykonywane jest dla każdego odbiorcy. W związku z tym wysyłanie masowej ilości e-maili staje się bardzo trudne.



Rysunek 2. *Najważniejsze zagrożenia*



Rysunek 3. *Global spam levels*

Dostępność rozwiązań wielosilnikowych na rynku

Na świecie istnieje wiele systemów oferujących rozwiązania antyspamowe oraz antywirusowe m. in. *Easy Antispam's Email Protection Services* firmy Interjuncture Corp., *MessageLabs Anti-Spam* rozwiązanie oferowane przez firmę MessageLabs, *SMI!*, którego producentem jest firma M2 NET, *IronPort Anti-Spam* firmy IronPort Systems oraz inne.

Istnieje kilka sposobów instalowania/użytkowania rozwiązań do ochrony serwerów pocztowych:

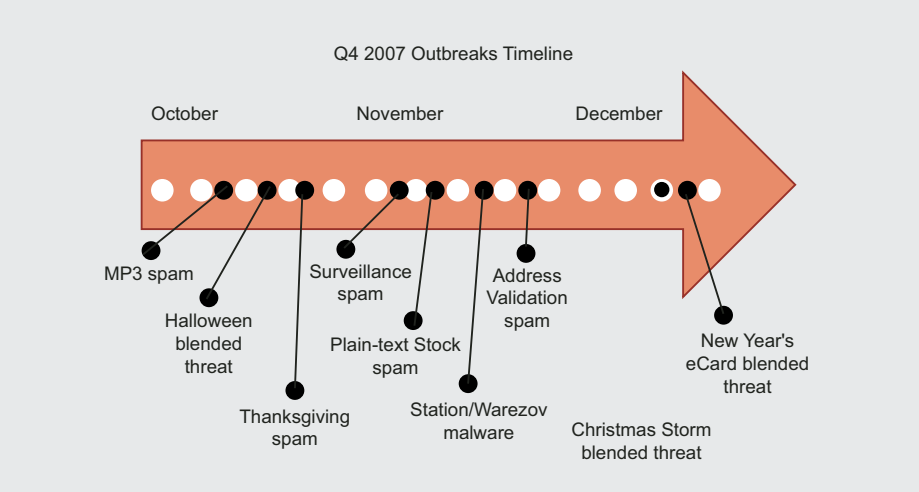
- Oprogramowanie może zostać zainstalowane na wydzielonym serwerze lub nawet na serwerze poczty w danej firmie. W celu zapobiegania obciążeniu serwera pocztowego zalecane jest instalowanie rozwiązań do ochrony SMTP na wydzielonym serwerze lub zastosowanie dedykowanego

serwera dostarczanego przez producenta. Rozwiązanie to może zostać zainstalowane przed lub za firewallem. W przypadku umieszczenia oprogramowania przed firewallem, należy odblokować na nim ruch z oprogramowania do portów TCP 80, 88, 25 oraz UDP 53. Zainstalowanie narzędzia przed serwerem pocztowym sprawia, że oprogramowanie działa jako MX, a jego zadaniem jest odebranie poczty (tym samym zachowuje się jak serwer pocztowy).

W przypadku zastosowania rozwiązania do ochrony konkretnego serwera pocztowego (np. Domino czy Exchange) oprogramowanie instalowane jest na serwerach poczty. Dedykowane oprogramowanie przechwytuje pocztę odbieraną przez mailbox serwera, a następnie rozpoczyna analizę poczty pod kątem kontroli treści oraz ochrony antywirusowej. W tym procesie stosowane są polityki i grupy reguł, które wcześniej zostały skonfigurowane przez administratora w danej firmie. Po zakończeniu analizy poczta, która została uznana za bezpieczną, zostaje doręczona do skrzynek. Oprogramowanie zintegrowane z konkretnym serwerem pocztowym realizuje również kontrolę treści.

Klienci mogą również zdecydować się na skorzystanie z zewnętrznej usługi. W standardowym wydaniu usługa zarządzalna polega na przekierowaniu poczty kierowanej do danego przedsiębiorstwa na serwery firmy, która oferuje tego typu usługi. Proces skanowania poczty jest realizowany przez oprogramowanie znajdujące się w kilku centrach danych zlokalizowanych w różnych krajach. Dzięki temu rozwiązaniu serwery firmy nie są narażone na bezpośrednie ataki i znacząco zmniejsza się obciążenie łącz telekomunikacyjnych (w listopadzie 2007 spam był odpowiedzialny za generowanie ponad 80% całego ruchu SMTP). W przypadku usługi list przechowywany jest na serwerach usługodawcy przez kilkaset milisekund, a w przypadku umieszczenia w kwarantannie – składowany jest on w szyfrowanej bazie danych, dedykowanej dla każdego klienta. Korzystając z usługi firma otrzymuje dwa niezależne serwery (główny i zapasowy), pozwalające na zachowanie ciągłości pracy przy jednoczesnej możliwości okresowej konserwacji i wyłączania jej serwerów. Wiadomości e-mail niedostarczone do serwerów pocztowych przedsiębiorstwa są kolejkowane na serwerach usługodawcy do momentu uruchomienia serwerów docelowych.

Omawiane rozwiązania wielosilnikowe umożliwiają zapobieganie wyciekowi informacji drogą poczty elektronicznej, realizują sprawdzanie poprawności protokołu SMTP, zapobiegają atakom dedykowanym dla konkretnych serwerów (ang. *identity spoofing*), są wyposażone w wiele metod antyspamowych (m. in. SPF, RBL i DNSRBL, HashCash, SURBL, Verifier, GreyListing, White & black IP lists, White & black address lists, Bayesian, Heuristic). Poza tym realizują również proces automatycznego sprawdzania istnienia nadawcy poprzez próbę połączenia lub sprawdzenie DNS przy dowolnym zagłębieniu. W przypadku większości rozwiązań wielosilnikowych poszukiwanie wirusów przeprowadzane jest dla treści wiadomości e-mail, zagnieżdżonych elementów MIME. Ponadto wykrywanie spamu jest realizowane przy pomocy modułów odpowiadających za sprawdzenie czy konto pocztowe, z którego przysłano wiadomość istnieje fizycznie na serwerze nadawcy. Sprawdzenie polega na wysłaniu wiadomości testowej do nadawcy. Rozwiązanie wielosilnikowe umożliwia zautomatyzowanie przeciwdziałania bombardowaniu e-mailami. Atakujący, który próbuje w powyższy sposób spowolnić lub unieruchomić pracę serwera poczty, zostaje odcięty na określony czas lub też zablokowany trwale. Poza tym można spotkać się z przydatnymi opcjami umożliwiającymi konfigurację identyfikującą i filtrującą załączniki według określonych rozszerzeń oraz zastosowanie konkretnych ustawień dla wskazanych użytkowników, np. zablokowanie wysyłania na zewnątrz plików z rozszerzeniem **.xls* określonym pracownikom lub też umożliwienie wysyłania plików o określonej wielkości w załącznikach. Interesującym elementem, który można dostrzec w rozwiązaniach wielosilnikowych, blokujących wiadomości z niechcianą zawartością jest możliwość binarnej analizy obrazów (np. filtrowanie przemycanych obrazków o treści pornograficznej przy zastosowaniu zaawansowanej technologii, której producentem jest firma LTU Technologies SA). W procesie analizy grafiki pierwszym etapem jest segmentacja obrazu. Technologia LTU wykorzystuje nieparametryczne, multiskalowalne podejście, dzięki czemu dzieli obraz na odpowiednie, wizualnie stabilne części. Dane wejściowe są analizowane pod kątem poszczególnych obszarów pikseli. Następnym etapem jest indeksowanie obrazu. Technologia LTU przypisuje podzielonemu obrazowi unikalny identyfikator nazwany podpisem (albo *zawartością DNA*). Zawartość DNA to zoptymalizowana kombinacja unikalnych cech tj. koloru, tekstury, kształtu, konfiguracji



Rysunek 4. *Q4 2007 outbreaks timeline*

w przestrzeni. Na końcu procesu obraz jest reprezentowany przez wektor liczbowy (zawartość DNA), w którym zakodowane są wszystkie szczegóły obrazu. W następnym procesie, tzw. klasyfikacji, zawartość DNA jest rozpoznawana przez moduły eksperckie zgodnie z ich bazą wiedzy. Moduły te wykorzystują najbardziej zaawansowane techniki rozpoznawania wzorców, takie jak: sieci neuronowe, funkcje z bazą radialną, estymację Bayesa czy maszynę wektorów wspierających. System ten przewyższa dotychczasowe techniki klasyfikacji obrazów ze względu na jego elastyczność i zdolność interaktywnego *uczenia się* od użytkownika, przy stałym poszerzaniu swojej bazy wiedzy. Warto zaznaczyć, że cały proces rozpoznawania obrazu, począwszy od jego segmentacji do określenia zawartości, dokonuje się w czasie rzeczywistym. Oprócz tego istnieje również możliwość szyfrowania newralgicznych informacji przesyłanych w wiadomościach e-mail oraz skanowania pod względem słów kluczowych. Poza tym, jak to bywa w przypadku oprogramowania antywirusowego oraz antyspamowego, istnieje możliwość tworzenia analiz w postaci wykresów i danych liczbowych w przedziałach czasowych określonych przez administratora. Standardowo technologie antywirusowe powinny być dostarczane przez kilka silników antywirusowych równocześnie, natywnie zintegrowane z oprogramowaniem. Wiadomości są skanowane przez wszystkie silniki jednocześnie. Każdy z silników zwraca informacje: *True* – oznajmiającą odnalezienie wirusa lub *False* – oznaczającą brak wirusa. W momencie, gdy przynajmniej jeden z silników rozpozna wirusa podejmowane są działania zgodne z utworzoną przez administratora polityką.

Aktualizacje oprogramowania umieszczane są zwykle na serwerze FTP, a informacje o nich można znaleźć na stronie WWW producenta. Klient sam pobiera poprawkę oraz instaluje ją u siebie na serwerze. W momencie pojawienia się problemów podczas aktualizacji oprogramowania, powinniśmy mieć możliwość skorzystania z pomocy telefonicznej, jak i bezpośredniej, oferowanej przez inżynierów producenta. Kolejna kwestia dotyczy sieci izolowanych. W przypadku tychże sieci większość oferowanych na rynku rozwiązań nie będzie w pełni realizować swojego zadania, ponieważ gros testów antyspamowych wymaga dostępu do sieci Internet (np. wymagana jest możliwość realizowania połączeń wychodzących na trzech, specyficznych dla testów, portach).

Archiwizacja i składowanie (ISO 27001)

W celu zapewnienia bezpieczeństwa (o czym mówi norma ISO 27001) należy archiwizować wszystkie wychodzące i przychodzące wiadomości e-mail. Kompleksowe rozwiązania do ochrony serwerów pocztowych umożliwiają archiwizację całej przesyłanej i odbieranej korespondencji lub tylko jej wybranych fragmentów. Poczta powinna być przechowywana w co najmniej dwóch różnych miejscach. Powinniśmy mieć także możliwość składowania danych na dedykowanych nośnikach (choćby na taśmach) poprzez np. IBM Tivoli Storage Manager, a także możliwość nagrywania paczek ok. 3,5 GB danych na DVD. Norma bezpieczeństwa ISO 27001 zaleca przechowywanie kopii poczty elektronicznej przez kilka lat. Często tylko wtedy ustalenia poczynione drogą elektroniczną można traktować jako wiążące.

Zastosowanie omawianych rozwiązań jest pomocne przy wdrażaniu polityki ochrony informacji zgodnej z normą ISO/IEC 27001.

Bezpieczeństwo informacji w świetle prawa

Bezpieczeństwo informacji w Polsce jest postrzegane głównie jako ochrona informacji niejawnych oraz danych osobowych. W każdej firmie znajdują się dane, które powinny podlegać ochronie. Ich wyciek lub utrata może pociągnąć za sobą szereg niekorzystnych skutków dla organizacji. Każda firma dbająca o bezpieczeństwo danych osobowych oraz informacji niejawnych powinna posiadać politykę bezpieczeństwa, jak również system zarządzania bezpieczeństwem informacji. Zagadnienia dotyczące bezpieczeństwa danych osobowych reguluje ustawa z dnia 29 sierpnia 1997 r. o ochronie danych osobowych. Natomiast kolejnym aspektem związanym z bezpieczeństwem, a zarazem ochroną osób prywatnych przed komercyjnym spamem, jest ustawa z dnia 18 lipca 2002 r. o świadczeniu usług drogą elektroniczną. Zgodnie z art. 10 tejże ustawy zakazane jest przesyłanie niezamówionej informacji handlowej skierowanej do oznaczonego odbiorcy za pomocą środków komunikacji elektronicznej, w szczególności poczty elektronicznej. Ustawa uznaje tę czynność za czyn nieuczciwej konkurencji. W myśl ustawy z dnia 2 marca 2000 r. o ochronie niektórych praw konsumentów oraz o odpowiedzialności za szkodę wyrządzoną przez produkt niebezpieczny, posłużenie się pocztą elektroniczną w celu złożenia propozycji zawarcia umowy może nastąpić wyłącznie za uprzednią zgodą konsumenta. Jednak w codziennej pracy mamy do czynienia ze spamem, a prawo nadal jest łamane. Dlatego tak ważne jest uświadomienie przedstawicielom kadry zarządzającej, jak również administratorom, jak ważne jest stosowanie optymalnych metod zabezpieczeń przed szkodliwym działaniem spamerów.

Podsumowanie

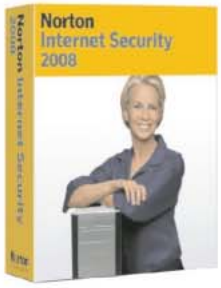
Rozwiązania omówione w artykule wskazują na wiele zalet systemów kompleksowych, posiadających szerokie możliwości konfiguracji oraz zapewniających przejrzysty, przystępny interfejs. W procesie zapewniania bezpieczeństwa firmy należy stawiać na systemy, które umożliwiają stosowanie technologii wielu producentów, wpływając na zmniejszenie ryzyka wystąpienia infekcji systemów komputerowych i jednocześnie na zredukowanie do zera prawdopodobieństwa nierozpoznania nowych zagrożeń. Każda firma, która jest świadoma wagi informacji w niej przetwarzanych, powinna stawiać na stosowanie rozwiązań gwarantujących utrzymanie bezpieczeństwa informacji na wysokim poziomie.

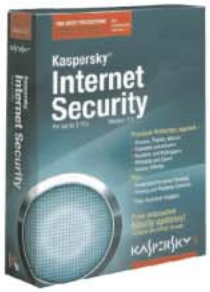
Piotr Cichocki
Z wykształcenia inżynier. Jest cenionym dziennikarzem zajmującym się tematyką bezpieczeństwa systemów teleinformatycznych. Obecnie pracuje jako specjalista w Wydziale Informatyki Urzędu Lotnictwa Cywilnego w Warszawie. Na swoim koncie ma wiele osiągnięć, do których można zaliczyć współpracę z Migut Media S.A., E-Security Magazine, CM LIM Sp. z o. o., a także z publicznymi oraz niepublicznymi dostawcami usług rynku pracy. Jego hobby to bezpieczeństwo systemów informatycznych, testy urządzeń typu appliance, muzyka, kompozycje, instrumenty klawiszowe, realizacja dźwięku, pływanie.
Kontakt z autorem: *cichocki.piotr@gmail.com*

TTS Company

Największy wybór oprogramowania w Polsce !

... w ofercie produkty ponad 200 producentów ...





www.OprogramowanieKomputerowe.pl

Altova
Acronis
A plus C
Attest Systems
Cloudmark
COMDOM Software
DameWare Development
ElcomSoft
Entensys
ES-Computing
ESET
Famatech
GASP
Ghisler
Ipswitch
Kerio
Kaspersky Lab
Krool Ontrack
Lavalys Software
LAVASOFT
McAfee
No Magic
PGP
RealVNC
SmartLine
Shavlik Technologies
SolarWinds.Net
Stellar Information Systems
System Tools
Tsafrin Computing
Trend Micro
Trolltech
...

Sprzedaż ▸▸ Dystrybucja ▸▸ Import na zamówienie

TTS Company Sp. z o.o. ul.Słowicza 53 02-170 Warszawa tel./fax: (022) 868 40 42 e-mail: tts@tts.com.pl www.tts.com.pl

FILIP DEMIANIUK

Wróg wewnątrz firmy

Stopień trudności

Przed plagą ataków zewnętrznych chroni nas wiele technologii, które – ciągle uaktualniane i doskonalone – są coraz skuteczniejsze. Jednak, jak pokazują statystyki, największe niebezpieczeństwo czyha wewnątrz firmy.

Gigabajty danych przechowywanych na mobilnych urządzeniach codziennie wyciekają z firm na zewnątrz, wiele cennych danych wysyłanych jest w zwykłych mailach przez osoby uprawnione do ich wykorzystywania, jeszcze inne są przekazywane w rozmowach telefonicznych. Jak się okazuje, najczęściej dzieje się tak w wyniku bezmyślności lub niefrasobliwości ludzi odpowiedzialnych za te informacje. Czy istnieją sposoby ograniczenia tego zjawiska? Jak można profesjonalnie chronić zasoby przed tego typu dywersją wewnętrzną? Sposobem na to mogą być systemy *Data Leak Prevention* (Zapobieganie wyciekom danych – DLP).

Jak wszystko, co tyczy się bezpieczeństwa, również stosowanie technologii *Data Leak Prevention* musi – a przynajmniej powinno – wynikać z potrzeby firmy, w której jest wdrożone.

W idealnym przypadku dokonaliśmy już oceny zasobów firmy, zarówno pod względem ich wartości, jak i ryzyk, na jakie są one narażone. Wyniki naszych analiz zostały oczywiście udokumentowane, odpowiednie środki zaradcze określone i powstały nasze korporacyjne Polityki Bezpieczeństwa. Jeśli tak się stało, to wiemy co, dlaczego i jak bardzo chcemy chronić. Przy okazji tego ćwiczenia po raz kolejny zdajemy sobie sprawę, że nigdy nie będziemy w stanie zapewnić 100% bezpieczeństwa naszych zasobów. To, o czym mówią *polityczni* – czyli ludzie, którzy na co dzień opiekują się korporacyjnymi politykami bezpieczeństwa – niestety pokrywa się z tym, co wyobraźnia podpowiada doświadczonym administratorom systemów związanych z IT security – wszystkiego zabezpieczyć się nie da. Nie da się w tej rzeczywistości, w jakiej pracujemy – choćby dlatego, że wprowadzając kolejne systemy zabezpieczeń, znacząco komplikujemy życie pracownikom firmy. Zabezpieczenie wszystkiego mogłoby doprowadzić do sytuacji, w której nie da się normalnie pracować, a uzyskanie dostępu do czegokolwiek trwa tyle, ile przejazd samochodem przez centrum każdego większego miasta w godzinach szczytu. Wtedy okazać by się mogło, że firma przestała zarabiać, straciła pracowników, a w konsekwencji i my razem z naszym bezpieczeństwem przestaliśmy być jej potrzebni.

Z drugiej zaś strony, czy na ochronę systemów wartych dla nas 10 000 zł warto wydawać 30 000 zł? A może dane dotyczące naszej firmy warte są więcej?

Założę się, że w przytłaczającej większości przypadków polityki bezpieczeństwa będą dotyczyły między innymi ochrony danych. Istnieje niemalże niezliczona liczba powodów, dla których dane warto chronić.

Skłonić może nas do tego chęć ochrony własności, takiej jak kody źródłowe aplikacji, nad którymi pracują nasi programiści. Schematy techniczne albo formuły chemiczne produktów przez nas sprzedawanych też wydają się wystarczająco cenne, żeby zawracać sobie

Z ARTYKUŁU DOWIESZ SIĘ

dlaczego warto stosować dobre polityki bezpieczeństwa,

dlaczego warto chronić firmowe dane,

co to jest Data Leakage Prevention,

jakie są wady i zalety dwóch głównych typów DLP,

co musisz wiedzieć, zanim zdecydujesz się na wybór konkretnej technologii.

CO POWINIENEŚ WIEDZIEĆ

powinieneś mieć teoretyczną wiedzę na temat podstaw bezpieczeństwa IT,

powinieneś mieć otwarty umysł.

głowę ich ochroną. W każdym przypadku powinniśmy również być zainteresowani ochroną danych osobowych naszych pracowników i klientów. Nie tylko wymaga tego nasze dobre imię, ale i prawo (GIODO, SOX itd.). Nie powinniśmy też zapominać o wszelkich dokumentach finansowych, umowach, kontraktach, przetargach i wszystkich innych informacjach, które w niepowołane ręce dostać się nie mogą.

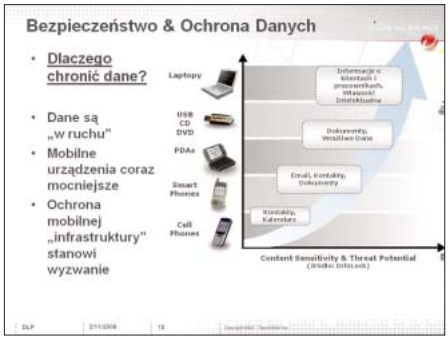
Żeby nie było nam zbyt łatwo – jak podaje Gartner, za oceanem 47% korporacyjnych danych jest przechowywanych na urządzeniach mobilnych, a w okresie ostatnich dwóch lat 350 000 tych urządzeń zostało skradzionych lub zgubionych (trend widoczny również na rodzimym podwórku). W Polsce też ostatnio głośno o skradzionych i zniszczonych notebookach oraz urządzeniach mobilnych. Czy to przypadek?

Zatem już ustaliliśmy – dane trzeba chronić, tylko jak?

Ochrona danych z DLP

Data Leakage Prevention (stosowana jest również nazwa *Data Leakage Protection*), w skrócie DLP, to praktyka bezpieczeństwa mająca na celu wykrywanie i zapobieganie *wyciekowi* danych poufnych poza fizyczne i logiczne granice przedsiębiorstwa, czy też poza obszary przedsiębiorstwa, stanowiące granicę dla danych poufnych. W momencie, gdy jakiś zbiór danych nie powinien wyciekać z firmy jako takiej, inny zbiór nie powinien wydostać się poza grupę prezesów i dyrektorów, a do jeszcze innego, poza pracownikami działu finansowego, dostępu nie powinien mieć NIKT! DLP ma za zadanie chronić przed wyciekiem danych niezależnie od jego przyczyny i strat, jakie może spowodować. Dane mogą opuścić firmę lub jej krytyczne systemy zarówno w wyniku ataku, jak i zwykłej nieświadomości czy nawet nieuwagi użytkowników. Z danych *Ponemon Institute Study* za rok 2006 wynika, iż 78% wycieków jest powodowanych przez pełnoprawnych, acz nieświadomych swoich działań, użytkowników systemów korporacyjnych. Dane najczęściej wydostają się z firmy poprzez zwyczajne wektory normalnej z pozoru aktywności, czyli *e-mail*, komunikatory internetowe, nośniki USB, a w końcu rozmowy telefoniczne.

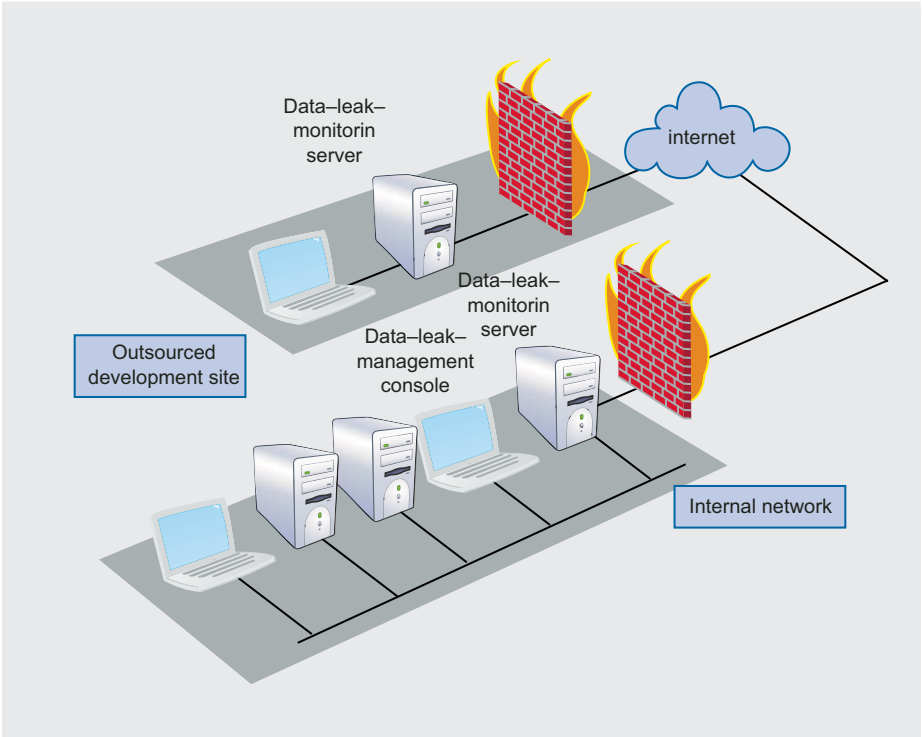
Zatem jakie główne technologie DLP zabezpieczają dane? Odpowiedzi na to pytanie jest wiele, w zależności od przyjętego scenariusza wycieku, a nade wszystko od tego, co i jak bardzo chcemy chronić. Zasadniczo istnieją dwa rodzaje DLP.



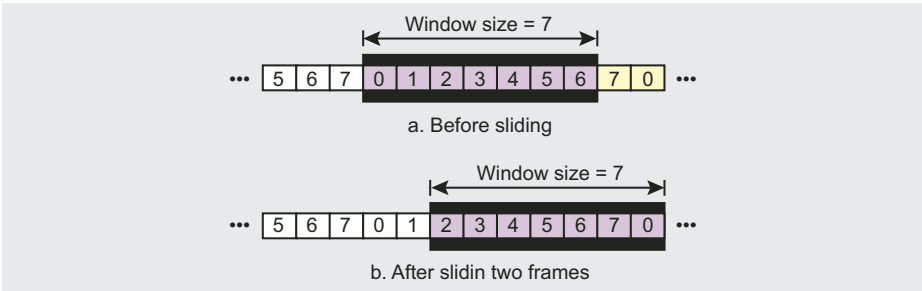
Rysunek 1. *Bezpieczeństwo i ochrona danych*

Network DLP (Gateway DLP)

Koncepcja sieciowego DLP (Rysunek 2) zakłada instalację systemu filtrującego dane w okolicach styku firmy z Internetem lub na brzegu chronionej podsieci. Najczęściej system jest dedykowanym urządzeniem, skanującym ruch sieciowy opuszczający firmę pod kątem nieautoryzowanych transmisji danych. Zaletą tej techniki jest stosunkowo proste wdrożenie, wynikające z braku konieczności instalowania agenta na chronionych stacjach roboczych. Analiza danych transmitowanych przez sieć w czasie rzeczywistym szczególnie obecnie – przy coraz większych prędkościach transmisji – nie jest rzeczą łatwą. Ze względów praktycznych tego typu rozwiązania sieciowe zawierają również komponenty pozwalające na analizę danych spoczywających na serwerach



Rysunek 2. *Network DLP*



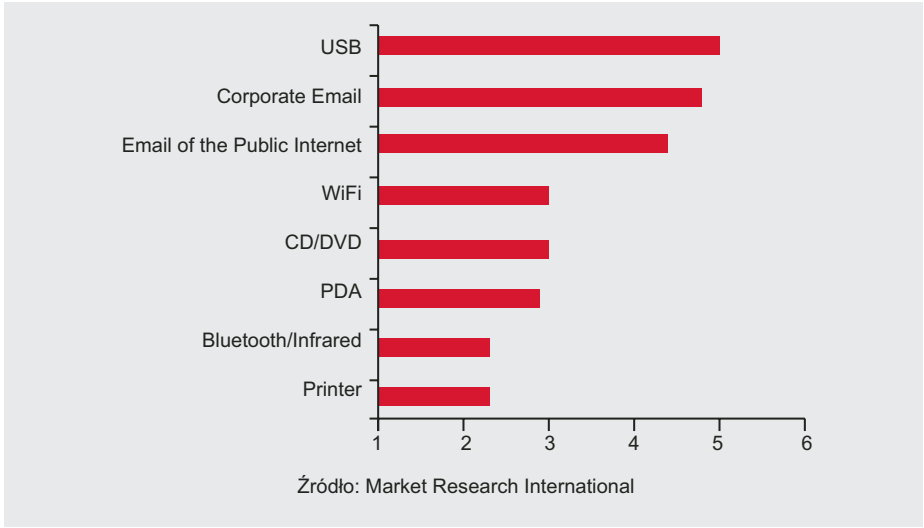
Rysunek 3. *Znakowanie danych metodą Sliding Window*

zasobowych, wskazanych uprzednio przez administratora. Analiza taka polega na przesłaniu dokumentów z chronionego repozytorium na urządzenie DLP, gdzie są one sprawdzane i znakowane. Dzieje się to w trybie *off-line*, więc nie jest aż tak krytyczne czasowo i pozwala na bardziej precyzyjne oznakowanie (*fingerprint*) dokumentów. Najbardziej popularną techniką znakowania danych wykorzystywaną w tego typu instalacjach jest *Sliding Window* (Rysunek 3).

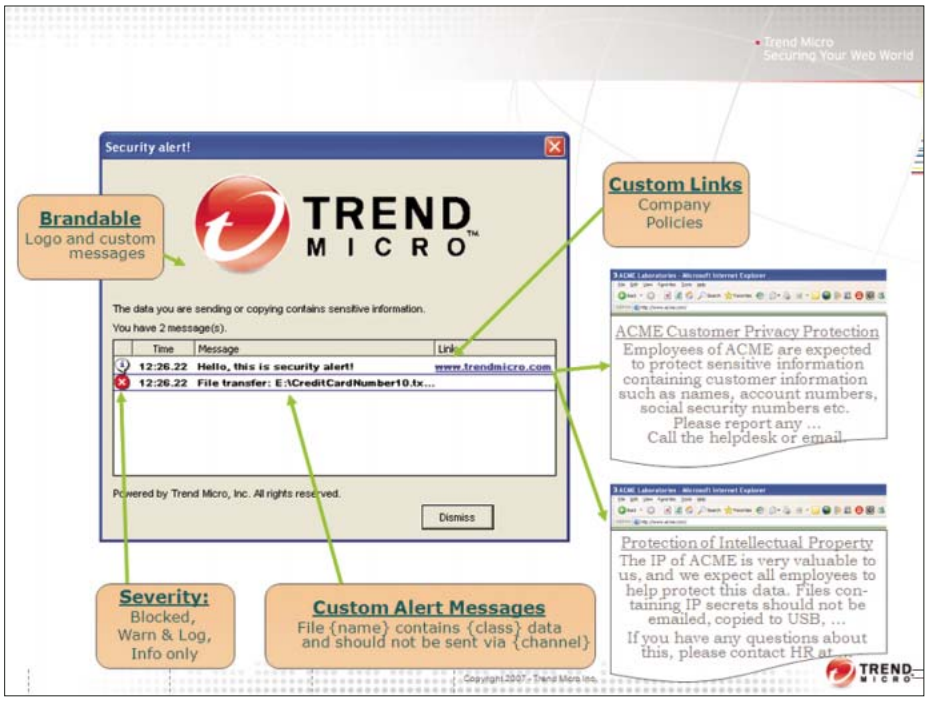
Przy wykorzystaniu tej techniki okno o określonej wielkości mieszczące dane (w przypadku DLP na przykład tekst) przesuwa się po dokumencie, pozwalając stworzyć bardzo dokładną – choć i znacznych rozmiarów – sygnaturę. Jej dokładność polega na tym, że jest nam znana kolejność znaków w całym analizowanym dokumencie, w następujących po sobie sekwencjach przesuwania okna o jeden krok do przodu. Niestety, problemem staje się tutaj wielkość samej sygnatury, która rośnie wraz z wielkością znakowanego dokumentu, osiągając niekiedy wręcz monstrualne wielkości. Producenci tego typu rozwiązań bronią się przed tą niedogodnością, znakując dokumenty mniej dokładnie – przez przesuwanie okna nie co jeden, a o więcej znaków, co pozwala zredukować wielkość *fingerprintu*. Urządzenie skanujące przechowuje sygnatury przeskanowanych dokumentów i przy ich udziale analizuje ruch sieciowy wychodzący z chronionego segmentu.

Reasumując, sieciowe rozwiązania DLP mają swoje zalety, takie jak prosta instalacja i możliwość wykorzystania dokładnej – nawet w przypadku *oszczędnego* skoku okna – techniki *sliding window* do tworzenia sygnatur. Czy zatem są to rozwiązania idealne? To zależy od tego, jak dokładnie chcemy nasze dane chronić. Dla bardziej wymagających instytucji poziom ochrony dostarczany przez Network DLP będzie zapewne o wiele za niski.

Sieciowe rozwiązania DLP nie poradzą sobie z przesyłaniem danych przez szyfrowane protokoły webowe, jak HTTPS – czyli na przykład przez prywatne konta *Webmail* pracowników czy ich komunikatory internetowe. Nie poradzą sobie również z transferem danych z komputerów pracowników na klucze USB czy zrzutami ekranowymi (*Print Screen*) i wysyłaniem ich jako obrazków zamiast dokumentów. Co więcej, wiedząc, że większość sprzedawanych obecnie komputerów to komputery przenośne, warto się zastanowić, co ochroni dane firmowe na laptopach pracowników, którzy wynieśli je z miejsca pracy i używają ich w domach albo korzystają z Internetu poprzez publiczne hot spoty? W takich przypadkach sieciowe rozwiązania DLP są bezradne, ale nie wszystko stracone!



Rysunek 4. *Główne źródła wycieku danych w firmach*



Rysunek 5. *Przykład okna powiadomienia o naruszeniu polityk bezpieczeństwa*

Host Based DLP

Czego zatem boją się dzisiejsze firmy? Według *Market Research International* firmy zainteresowane rozwiązaniami DLP najbardziej obawiają się wycieku danych poprzez porty USB, korporacyjną pocztę *e-mail* i przez prywatne konta *e-mail* pracowników (Rysunek 4).

Do ochrony przed dwoma z trzech głównych, jak i przed wszystkimi pozostałymi zagrożeniami, niezbędne jest rozwiązanie działające na komputerach użytkowników – *host based DLP*.

W odróżnieniu od rozwiązań sieciowych, *host-based DLP* bazuje na agentach zainstalowanych i działających na poszczególnych stacjach roboczych

i serwerach. Ich zadaniem jest monitorowanie wszystkiego, co dzieje się z chronionymi informacjami – zupełnie niezależnie od tego, gdzie (fizycznie) operacje na danych mają miejsce. Rozwiązania te są równie skuteczne w sieciach firmowych, w domach pracowników i w każdym innym miejscu, gdzie korzystają oni z firmowego sprzętu. Co więcej, mamy możliwość wdrożenia różnych polityk w zależności od tego, czy chroniony komputer znajduje się w naszej sieci, czy też nie (*on i off-network policies*).

Wyobraźmy sobie, że Kasia, asystentka prezesa naszej kochanej firmy, nie jest zadowolona z pracy u nas. Chcąc zmienić pracodawcę, zaprzyjaźnia się z pracownikiem konkurencji i postanawia wkupić się w jego łaski, dostarczając mu dokument dotyczący szczegółów naszego kontraktu z największym klientem. Ponieważ nasza firma, producent systemów i dostawca usług IT, korzysta z rozwiązań DLP, próba wysłania rzeczonego dokumentu przez firmowe konto pocztowe Kasi została zablokowana. Nasza przykładowa bohaterka postanawia wynieść poufne dane na kluczu USB. Niestety, podczas próby wykonania tej operacji, Kasia zostaje pouczona, że takie działanie jest wbrew politykom bezpieczeństwa firmy i zostało zarejestrowane w celu przeprowadzenie analizy zjawiska przez stosowne służby (czyli nas). Co więcej, Kasia została również poproszona o podanie uzasadnienia dla swojego działania (Rysunek 5).

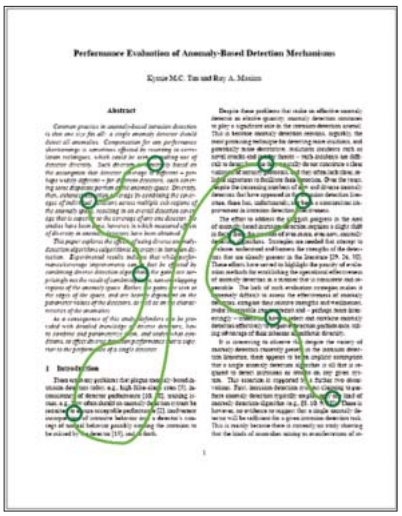
W takiej sytuacji Kasia postanowiła uciec się do fortelu i zaniosła swojego notebooka wraz z naszym dokumentem do domu. Tam przeredagowała jego część i postanowiła nie korzystać więcej z usług naszej firmy w celu jego przesłania. Zamiast naszych serwerów pocztowych skorzystała z prywatnego konta *web mail* (Rysunek 6).

Czy dowiedzieliśmy się, że takie zdarzenie miało miejsce? Tak, gdyż agent, znajdujący się na chronionym komputerze Kasi, cały czas trzymał rękę na pulsie, pilnując naszych danych nawet po ich modyfikacji. Zablokował transmisję, my dostaliśmy raporcik, a Kasia dłużej nie pracuje w naszej firmie.

Skąd agent wiedział, że wysyłany plik był tajnym dokumentem firmowym, a nie prywatnym listem miłosnym? Rozwiązania *host-based*, tak jak i rozwiązania sieciowe, korzystają z technik znakowania danych – więc są w stanie zidentyfikować, czy sprawdzany tekst powinien być chroniony, czy też nie. Oczywiście, w tego typu instalacji sygnatury danych muszą być przechowywane na stacjach roboczych, gdyż w ten sposób następuje weryfikacja zdarzeń pod kątem ich zgodności z przyjętymi politykami. Tego typu podejście mogłoby być problematyczne, gdyby chcieć wykorzystać techniki *sliding window* do tworzenia sygnatur dokumentów, ponieważ ich wielkość znacząco obciążała by chronione komputery. Na szczęście są jednak inne, niemal równie dokładne, techniki uzyskiwania *fingerprintów*. Jednym z przykładów jest *Data DNA* (Rysunek 7).

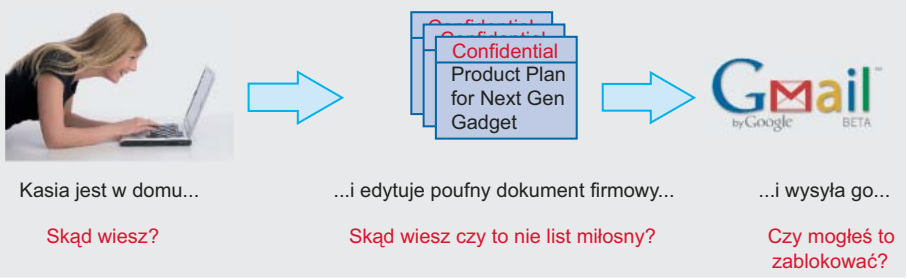
Technika ta oparta jest na algorytmie, który najpierw poszukuje znaków szczególnych dla danego dokumentu (niezależnie od języka, w jakim ten jest napisany – np. polski, chiński lub C++ itd.), a później określa ich wzajemne położenie względem siebie. Pozwala to na stworzenie bardzo małych (poniżej 1kb) sygnatur, których transfer do agentów nie stanowi żadnego obciążenia dla sieci i samych komputerów. Oczywiście samo tworzenie sygnatur – tak jak w przypadku rozwiązań sieciowych – ma miejsce na dedykowanym urządzeniu rezydującym w sieci. Administrator wskazuje, które repozytoria dokumentów i jak często mają być skanowane, a znajdujące się tam dokumenty – znakowane. Wszystkie aktualizacje na bieżąco trafiają do wszystkich agentów zainstalowanych na firmowych komputerach, natomiast sam agent, rozpoznając chronione dane, może wdrożyć odpowiednią politykę zachowania. Plusem takiego mechanizmu jest jego odporność na błędy, które mogłyby powstać w wyniku edycji dokumentów źródłowych, na przykład poprzez usunięcie ich części, zamianę kolejności, czy też doklejenie nowej treści i próbę zrobienia czegoś niewłaściwego z tak zmodyfikowanymi danymi.

Wracając do przerabianego scenariusza – czy działanie użytkownika dało się zatrzymać? Jak najbardziej. Zastosowanie agenta daje nam możliwość blokowania transmisji szyfrowanej, zarówno poprzez klienta poczty, jak i komunikator internetowy. Co więcej, możemy wybrać sposób interakcji z użytkownikiem. Możemy zablokować i zalogować tę próbę. Możemy również dać szansę użytkownikowi na wpisanie uzasadnienia swojego działania, możemy też poinformować go, dlaczego takie działanie jest zablokowane i dlaczego wynika to z posiadanej przez nas polityki

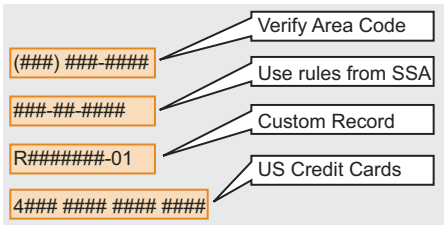

Performance Evaluation of Anomaly-Based Detection Mechanisms

Abstract

Rysunek 7. *Działania algorytmów Data DNA*



Rysunek 6. *Przykład wycieku danych*



Rysunek 8. *Filtrowanie danych przy*

bezpieczeństwa. A może chcemy dać użytkownikowi możliwość przeprowadzenia tego działania wraz z uzasadnieniem? Agent pozwala nam na taką elastyczność, daje możliwość ochrony, edukuje oraz pozwala uwiarygodnić swoje działanie.

A co by się stało, gdyby Grzegorz, nasz przykładowy specjalista ds. finansów chciał wynieść z firmy dane dotyczące naszych operacji finansowych? Jeśli wiemy, jaki format mają numery rachunków bankowych (a wiemy to doskonale) albo jaka jest struktura dowolnego typu danych, które naszym zdaniem mają pozostać poufne, to agent zatrzyma wszystko, co spełni zdefiniowane przez nas kryteria.

No dobrze, ale Grzegorz to bystry gość – używa komputera codziennie i od lat. Wie, że procesy działające w systemie można wyłączyć, a wtedy – hulaj dusza piekła nie ma! Czy jeśli na komputerze działa agent, to użytkownik może po prostu go wyłączyć i bezkarnie korzystać z komputera? Nie do końca. Dobre produkty DLP wykorzystują technologie *stealth*, czyli nic innego jak mechanizmy typu *rootkit*, ukrywające pliki i serwisy agenta przed okiem użytkownika komputera. Tylko administrator systemu DLP może podjąć decyzję, czy mają być one widoczne z poziomu systemu, czy też nie. Wygląda więc na to, że tak łatwo Grzegorzowi nie pójdzie, a my będziemy wiedzieli, że podejmuje dziwne próby przesłania danych i weźmiemy go pod lupę!

A gdyby nasi software developerzy chcieli po cichu podzielić się z kimś naszym kodem źródłowym? A może nasza koleżanka z działu kadr chciałaby się z kimś podzielić informacjami o naszych pensjach? Oczywiście możemy zastosować techniki z powyższych przykładów, ale ponadto zostaje nam w odwodzie wykorzystanie filtra słów kluczowych, informacji *meta-data* oraz wyrażeń regularnych.

Dla słów kluczowych możemy ustawić filtry tak, żeby za każdym słowem z naszej podejrzanej listy, które pojawia się w sprawdzanym dokumencie, liczba przyznawanych punktów ryzyka zwiększała się, wpływając na decyzję o blokadzie transferu.

Wykorzystując informacje *meta-data* możemy blokować pliki o danych właściwościach, co nie musi mieć wyraźnego związku z ich treścią. Możemy potraktować w szczególny sposób pliki o interesujących nas atrybutach, pliki które powstały nie wcześniej niż np. w ostatni wtorek albo takie, których twórcą jest nasz główny księgowy Kamil.

W końcu, przy pomocy hostowego DLP możemy też kontrolować coś, czemu nie jesteśmy w stanie przeciwdziałać z poziomu sieci. Mamy możliwość zablokowania dostępu do kluczy USB, zarówno całkowicie, jak i tylko w przypadku transferu chronionych dokumentów. Dokładnie w ten sam sposób możemy poradzić sobie z problemem nieautoryzowanego nagrania poufnych danych na nośniki CD czy DVD, ich wydrukowania i wyniesienia z firmy w wersji papierowej. Możemy też uniemożliwić tworzenie zrzutów ekranowych i przekazywanie ich na zewnątrz jako pliki graficzne.

A ile nas to wszystko kosztuje? W końcu mówimy o oprogramowaniu, które rezyduje na firmowych komputerach i jest aktywne przez cały czas ich pracy. Tutaj na szczęście, w zależności od rozwiązania, koszty mierzone obciążeniem naszych komputerów mogą być bardzo małe – nawet tak niewielkie, jak niecałe 9kb pamięci operacyjnej i około 2,5% mocy procesora.

A co z tymi, którzy są zdolniejsi niż przeciętny użytkownik, czy nawet administrator? Oni na pewno znajdą sposób na obejście zabezpieczeń.

Programming	Legal Dictionary	Medical Terms
atof(	MALICE	Abdominalgia
atoi(	ADJOURNMENT	parathyroid gland
atol(	DISMISSAL	Vascular
else if	AFFIDAVIT	Hypoproconvertin emia
#en dif	ALIMONY	Polyonychia
errn o .h	CURIAE	Gangrene
java.applet	BIFURCATION	Osteomyelitis
java.awt	TRUST	spinal curvature
java.beans	CAPITAL	Tumor
java.io	GAIN	Osteomyleitis
java.lang	CAPITAL	chylomicronemia

Rysunek 9. *Filtrowanie danych przy pomocy słów kluczowych*

Podsumowanie

Sieciowe DLP, choć proste w implementacji, nie jest specjalnie skuteczne. *Host Based DLP* wydaje się być lepszym rozwiązaniem, ale okupionym koniecznością wdrożenia. Niestety, musimy też wiedzieć, że uzdolniony specjalista tak czy inaczej znajdzie w końcu sposób na obejście tych zabezpieczeń. Jeśli będziemy uważnie przeglądać logi i alerty, mamy szanse zobaczyć próby oszukania systemu i im przeciwdziałać, ale czy to wystarczające narzędzie? Czy warto się w to bawić?

Myślę, że tak, jeśli podejdziemy do DLP jako jednego z narzędzi, będącego w stanie pomóc nam osiągnąć zgodność z tym, co zapisaliśmy w politykach bezpieczeństwa lub tym, co narzucają nam regulacje prawne. Tak, jeśli nie zaufamy ślepo reklamom i nie uwierzymy, że samo DLP rozwiąże nasze problemy. Tak, jeśli potraktujemy DLP jako jeden z elementów naszej strategii bezpieczeństwa i wykorzystamy je w połączeniu z wiedzą specjalistów i właściwym wykorzystaniem innych technik ochrony, jak na przykład *Content Filteringiem*. Musimy pamiętać, że za 78% przypadków wycieku danych stoją nieuważni lub nie wyedukowani pracownicy, a nie genialni hakerzy. Czy jesteśmy w stanie zatrzymać tych pierwszych? Na pewno tak. Czy jesteśmy w stanie zatrzymać tych ostatnich? Możemy próbować, ale nie powinniśmy wierzyć w skuteczność DLP w tym zakresie bardziej, niż w skuteczność naszych działów *IT Security*. Rozwiązania DLP służą temu, żeby uporać się z wymienionymi wyżej 78%.

Filip Demianiuk
Technical Channel Manager w firmie Trend Micro, będącej czołowym dostawcą zabezpieczeń internetowych i zapewniającej przedsiębiorstwom oraz użytkownikom indywidualnym bezpieczeństwo wymiany informacji. Od ponad 10 lat współpracuje z największymi firmami w Polsce, zarządzając zespołami IT i wdrożeniami złożonych projektów z zakresu bezpieczeństwa infrastruktury teleinformatycznej oraz konsultując tworzenie i utrzymanie największych systemów IT w kraju.
Kontakt z autorem: *Filip_Demianiuk@trendmicro.com*

Prenumerata Pro

CCNS

Działalność firmy skoncentrowana jest wokół hasła zapewnienia pełnego bezpieczeństwa funkcjonowania Klienta w realiach współczesnej gospodarki. Jako Expert Partner firmy WatchGuard Inc. oferujemy kompleksowe rozwiązania bezpieczeństwa sieci i systemów informatycznych obejmujące nowoczesne urządzenia typu Unified Threat Management, niezawodny serwis i szeroki wachlarz szkoleń.

www.ccns.pl

Media Systems

Firma Media Systems oferuje Państwu usługi oparte o rozwiązania systemu CashBill.pl i są to: organizacja kampanii marketingowych opartych o konkursy SMS'owe, organizacja kampanii reklamowych serwisów internetowych, system płatności elektronicznych poprzez karty płatnicze i przelewy. Prócz wymienionych usług oferujemy także budowę stron www, sprzedaż domen oraz pozycjonowanie w wyszukiwarkach. Zapraszamy do współpracy.

TTS Company Sp. z o.o.

Oprogramowanie komputerowe - sprzedaż, dystrybucja oraz import na zamówienie. W ofercie programy autorstwa ponad dwustu firm z całego świata. Chcesz kupić oprogramowanie i nie możesz znaleźć polskiego dostawcy? Skontaktuj się z nami? sprowadzimy nawet pojedyncze licencje.

www.OprogramowanieKomputerowe.pl

kontakt do nas:
katarzyna.juszczynska@software.com.pl,
robert.gontarski@software.com.pl
tel.: 22 427 36 77

Sokra-NET

Działa od roku 2002, specjalizuje się w szeroko pojętym bezpieczeństwie informacji. Posiada wykwalifikowany specjalnie do tych celów zespół inżynierów którzy przy współpracy z naszymi klientami maksymalizują bezpieczeństwo danych, audytując i dobezpieczając. Wykonujemy testy penetracyjne, analizy kodów źródłowych, testy wydajnościowe aplikacji i ich środowisk teleinformatycznych. Wdrażamy polityki bezpieczeństwa. Wspomagamy naszych partnerów merytorycznie.

www.sokra.net

Enigma SOI

Głównym przedmiotem działalności firmy jest produkcja, wdrażanie i sprzedaż systemów służących do ochrony informacji.

- elektroniczna skrzynka podawcza
- centra certyfikacji kluczy
- podpis elektroniczny i szyfrowanie na serwerach i stacjach klienckich
- zabezpieczanie stacji lokalnych
- karty elektroniczne i czytniki

www.enigma.com.pl

Petrosoft

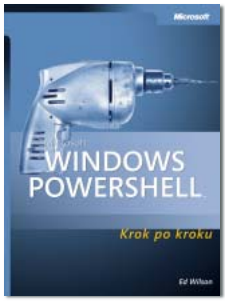
Partner Microsoft Bussiness Solutions Dynamics GP. Budowa sklepów internetowych, serwisów WWW, prezentacji multimedialnych. Budowa sieci LAN, WAN, telekomunikacyjnych. Telefonia stacjonarna, VoiceIP. Usługi outsorcingowe dla dużych firm z zakresu informatyki i telekomunikacji. Oprogramowanie na zamówienia. Dostawa serwerów, sprzętu, oprogramowania.

38-200 Jasło, ul. 3 Maja 101
Biuro: (13) 44 66 666
biuro@petrosoft.pl

Kei.pl

Kei.pl działa na rynku usług hostingowych od 2000 roku. Do naszych zadowolonych Klientów z dumą możemy zaliczyć wiele przedsiębiorstw sektora MSP, instytucji oraz osób prywatnych. W ofercie Kei.pl znajdują się pakiety hostingowe, a także usługi dla wymagających Użytkowników – platformy *e-Biznes oraz serwery fizyczne.*

http://www.kei.pl



Autorzy: Ed Wilson
Wydawca: Promise
Rok wydania w Polsce: 2007
Liczba stron: 282
Recenzent: Łukasz Gładysz

Microsoft Windows Powershell, krok po kroku

Narzędzie Powershell, przez większość użytkowników i administratorów znane jako *Monad*, staję się coraz to ciekawszą alternatywą dla *skryptowania* pod Windows. Już wprowadzenie do książki utwierdza nas w tym przekonaniu, czytamy w nim bowiem, iż zadania trudne lub bardzo pracochłonne w realizacji przy wykorzystaniu innych języków skryptowych (np. *VBScript*), są dużo łatwiejsze do wykonania właśnie w omawianym na łamach książki środowisku.

Autor przedstawia szeroką paletę odbiorców, do których kierowana jest książka. W tym momencie mógłbym się pokusić o stwierdzenie, iż jest ona adresowana praktycznie do każdego odbiorcy zainteresowanego zagadnieniem elastycznego i potężnego narzędzia do tworzenia skryptów. Obowiązkowo powinni tu chociaż zajrzeć administratorzy sieciowi Windows i wszyscy, którzy do swej pracy w Windows potrzebują elementów automatyzacji działań.

Treść jest zorganizowana w taki sposób, że stopień zaawansowania Czytelnika nie jest aż tak ważny. Oczywiście, mimo wyczerpujących wskazówek i wyjaśniania poszczególnych kwestii krok po kroku, aż do ich wykonania (jak zresztą sugeruje podtytuł książki), potrzebna jest choć podstawowa wiedza w zakresie działania systemu Windows i ogólne obeznanie z komputerem.

W myśl zasady: *Czego nie ma w Internecie – to nie istnieje* w poszukiwaniu informacji najpierw zagłębiam się w sieć globalną. Tak było i w przypadku Windows Powershell – niestety, tym razem się zawiodłem, ponieważ polskie witryny nie dają pełnego przeglądu funkcji tego narzędzia. Mówiąc o tylko polskich stronach kieruję się, skądinąd przykrym, doświadczeniem braku biegłej znajomości języka angielskiego wśród polskich administratorów i pracowników technicznych. W tym momencie książka pana Eda Wilsona jest dla nich idealnym rozwiązaniem, ponieważ otrzymują wyczerpujące omówienie tematu i to w ojczystym języku. Oczywiście daleki jestem od stwierdzenia, że osoby znające język angielski mogą podarować sobie tę pozycję. Sam wielokrotnie przekonałem się, jak ważne i nieocenione w trakcie wytężonej pracy jest posiadanie źródła z usystematyzowaną i logicznie ułożoną wiedzą. Szczególnie pomocne w czasie poszukiwania konkretnej informacji – obok spisu treści i indeksu – okazują się podsumowania poszczególnych działów (jako zestawienie najważniejszych informacji z odpowiedniej części książki).

Książka zaopatrzona jest w stosunkowo obszerne dodatki, a mianowicie: Polecenia *cmdlet* zainstalowane w Windows Powershell, Nazewnictwo poleceń *cmdlet* i Tłumaczenie języka VBScript na Windows Powershell. Szczególnie ciekawa jest ostatnia pozycja. Nie da się ukryć, że książka napisana została praktycznie identycznym stylem, jak wcześniejsze pozycje autora na temat tworzenia skryptów w języku VBScript. Przez pierwsze rozdziały odnieść wrażenie, że mamy do czynienia z ich kontynuacją, swego rodzaju uzupełnieniem o coś nowszego i lepszego. Ostatni dodatek, zestawiający odpowiedniki funkcji VBScript i Powershell (wraz z opisami), jest niczym instrukcja, jak przejść ze starej technologii na nową.

Spoglądając krótko na techniczne wykonanie książki, nie bardzo jest się do czego przyczepić. Układ oraz szata graficzna są podyktowane wymaganiami serii, do jakiej przynależy nasza pozycja. Czytelne bloki tekstu, przejrzysta czcionka, wyróżnione wskazówki i dane newralgiczne. Osobiście brakuje mi twardej okładki, która chroniłaby przed nadmiernie szybkim zużyciem, ponieważ – jak wielu moich kolegów – nie jestem typem człowieka trzymającego książki na półce, tylko ciągle są one *w obiegu*, ale to już kwestia upodobań.

Mimo, iż do tej pory sypały się same superlatywy, można dopatrzeć się kilku niedociągnięć. Pierwszą rzeczą, która mnie osobiście (moją żonę także, więc nie jest to ocena odosobniona) w pewnym momencie zaczęła denerwować, był zabieg ułatwiający początkującemu czytelnikowi przebrnięcie ze zrozumieniem przez lekturę. Mam tu na myśli zamieszczone niemal przed każdym zadaniem polecenia, nakazujące otworzyć Windows Powershell. O ile zrozumiałe jest to – załóżmy – przez pierwsze dwa lub trzy rozdziały, o tyle nie potrafię znaleźć usprawiedliwienia, dlaczego tak jest przez całą książkę. Jeżeli Czytelnik zapoznający się z czwartym rozdziałem nie wie, że aby wykonać skrypt, najpierw musi

uruchomić środowisko, to albo niech wróci do początku książki, albo odłoży lekturę do czasu uzupełnienia swoich wiadomości z zakresu podstaw informatyki.

Podsumowując, publikacja Microsoft Windows Powershell jest zdecydowanie obowiązkową lekturą dla grona nie tylko zaawansowanych użytkowników komercyjnych i administratorów. Co bardziej postępowi nauczyciele technologii informacyjnej w liceach czy technikach, mogliby z powodzeniem wprowadzić tematykę Powershell w klasach profilowanych lub na kołach zainteresowań, a książkę Eda Wilsona wskazać jako źródło wiedzy. O wysokim poziomie merytorycznym publikacji świadczy choćby fakt, iż na stronie WWW Microsoftu zamieszczone zostały jej fragmenty jako wprowadzenie do technologii.

Microsoft Windows Workflow Foundation. Krok po kroku

Po ukazaniu się platformy .NET w wersji 3.0 w krótkim czasie pojawiło się kilka nowych technologii, a mianowicie: *Windows Presentation Foundation* (*WPF*), *Windows Communication Foundation* (*WCF*), *Windows CardSpace* i *Windows Workflow Foundation* (*WF*).

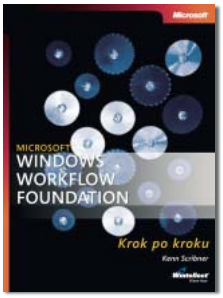
Workflow Foundation pozwala deweloperom budować aplikacje korzystające z koncepcji przepływu danych. Kenn Scribner nie zawodzi Czytelnika, przedstawiając samą koncepcje technologii, a następnie wprowadzając coraz bardziej złożone pojęcia. Jeżeli tylko tworzysz aplikacje, które reprezentują jakiś rodzaj przepływu danych, będziesz naprawdę zadowolony z *Workflow Foundation*.

Ta książka naprawdę zachęca do czytania. Nie otrzymujemy1000 – stronicowego tomiszcza, w którym musimy przebijać się przez dziesiątki nużących tabel czy specyfikacji funkcji i metod, ale pozycję przyjazną Czytelnikowi. Rzadko zdarza się że podczas lektury pozycji technicznej tak szybko i płynnie przechodzi się do kolejnych rozdziałów. Pierwsza część książki zawiera wiedzę, której poznanie jest niezbędne dla zrozumienia i stosowania *Workflow Foundation*, natomiast kolejne rozdziały rozszerzają podstawowy zakres informacji, prezentując zestaw bardziej zaawansowanych technik. Należy zaznaczyć, że podtytuł *krok po kroku* już na początku naszej znajomości z książką trafnie uspokaja Czytelnika, zapewniając, że stosowana terminologia oraz wymagany zakres wiedzy nie będzie sprawiać problemów początkującym użytkownikom – wręcz przeciwnie, poszczególne pojęcia wyjaśniane są w sposób prosty i zrozumiały. Dzięki temu wiadomo, jak wprowadzić nowo poznaną technologię do pisanych aplikacji.

Jeżeli nigdy wcześniej nie używałeś *Workflow Foundation*, będziesz naprawdę zadowolony, zapoznając się za pośrednictwem Kenna Scribnera z podstawami tej technologii. Godnym polecenia jest sposób przedstawienia materiału. Autor nie specyfikuje monotonnej listy komend i poleceń wraz z poszczególnymi parametrami, ale wprowadza przykład realnej aplikacji opartej na *Workflow Foundation* i na nim wyjaśnia rolę kolejnych stosowanych poleceń czy funkcji. Czyni to naukę zdecydowanie bardziej *życiową*. Na przykładach tych bardzo łatwo można zbudować narzędzie służącą do śledzenia błędów czy też aplikację e–commerce, która używa *Workflow Foundation* do obsługi procesów płatności. Korzystając z omawianej w książce technologii, nie musimy używać drogich, komercyjnych rozwiązań a ogranicza nas nie grubość portfela, lecz tylko wyobraźnia.

Wraz z kolejnymi rozdziałami autor wprowadza nas w coraz to nowe tajniki *Workflow Foundation*, dzięki czemu nawet początkujący użytkownicy – przy pomocy zamieszczonych w książce oraz na dołączonej płycie CD przykładów i kodów źródłowych – będą mogli tworzyć własne, przydatne im aplikacje.

Autor poruszył w książce wszystkie istotne kwestie dotyczące *Workflow Foundation*, przedstawiając je w sposób zrozumiały i jasny. Lektura tej pozycji sprawiła mi sporą satysfakcję, ponieważ wraz z każdym kolejnym rozdziałem rozumiałem coraz więcej na temat opisywanej technologii. Podczas czytania nie miałem żadnych negatywnych odczuć odnośnie któregokolwiek z rozdziałów. Książka dostarcza wszystkiego, czego początkujący użytkownik *Workflow Foundation* potrzebuje do uruchomienia swoich własnych aplikacji. Gorąco polecam pozycję Kenna Scribnera każdemu, kto nigdy nie używał *Workflow Foundation* i chce zobaczyć, jak ta technologia może wzbogacić jego aplikacje.



Autorzy: Kenn Scribner
Wydawca: Promise
Rok wydania w Polsce: 2007
Liczba stron: 474
Recenzent: Marcin Nawrocki

Kryptografia to moja pasja

Jacek Pokraśniewicz, od 1995 roku jest prezesem zarządu firmy ENIGMA Systemy Ochrony Informacji Sp. z o.o., z wykształcenia jest informatykiem.

hakin9: Czy jesteście w jakiś sposób połączeni z niemiecką enigmą?
Jacek Pokraśniewicz: *Połączeni* to chyba nie do końca odpowiednie słowo, ale pewien związek rzeczywiście istnieje. Otóż nasza firma została założona przez ludzi pasjonujących się współczesną kryptografią i wierzących w możliwości praktycznego zastosowania osiągnięć tej nauki w polskich firmach i instytucjach cywilnych (teraz to oczywiste, ale w latach 90, gdy powstawała firma, oczywiste to wcale nie było). A dlaczego *ENIGMA* ? Aby jakoś nawiązać, uhonorować niedoceniane wtedy osiągnięcia polskich kryptologów w przełamaniu szyfru niemieckiej maszyny szyfrującej *ENIGMA*.

h9:Czy używacie tego samego szyfru do szyfrowania danych, który zastosowano w niemieckiej enigmie?
J.P.: Nie, używamy szyfrów współczesnych – nowocześniejszych, bezpieczniejszych i przystosowanych do dostępnych aktualnie technologii przetwarzania informacji.

h9: Czy pierwsze Pana spotkanie z kryptografią było przypadkowe?
J.P.: Tak, można powiedzieć, że moje pierwsze spotkanie z kryptografią było przypadkowe. Jako student ostatnich lat Wydziału Elektroniki Politechniki Warszawskiej szukałem pracy na wakacje – i przypadkiem trafiłem na dr inż. Krzysztofa Gaja (obecnie jest profesorem w USA), kierującego grantem KBN, którego celem było stworzenie oprogramowania wykorzystującego metody kryptograficzne do ochrony informacji. Pracując przy grancie – i implementując od podstaw algorytmy kryptograficzne, poznałem *co to jest* kryptografia (bo trudno mówić, że poznałem kryptografię). Kilka lat później (w 1993 r.) założyliśmy wspólnie firmę (*ENIGMA SOI*), która wyrosła właśnie w wiary w to, że *takie fajne rzeczy* na pewno się komuś przydadzą i ktoś będzie chciał to kupić.

h9: Jaki algorytm kryptograficzny według Pana jest najlepszy?
J.P.: Chyba nie ma czegoś takiego jak *najlepszy* algorytm. Przede wszystkim mamy dwie zupełnie odrębne kategorie algorytmów kryptograficznych: algorytmy symetryczne i algorytmy klucza publicznego. Najbardziej znanym obecnie algorytmem symetrycznym jest algorytm AES (*Advanced Encryption Standard*) i w pewnym sensie można by powiedzieć, że jest *najlepszy* w tej

klasie – co oznacza, że gdybym miał komuś rekomendować zastosowanie jakiegoś algorytmu symetrycznego do zapewnienia poufności, to polecałbym AES. Choć mogę sobie wyobrazić sytuacje, że w pewnych zastosowaniach inny algorytm byłby *lepszy* (tzn. np. szybszy – przy zachowaniu wymaganego poziomu bezpieczeństwa).

W klasie algorytmów klucza publicznego niewątpliwym królem jest od wielu lat algorytm RSA. Wydaje się jednak, iż poziom bezpieczeństwa tego algorytmu przy powszechnie stosowanych długościach kluczy (1024 bity) nie jest zbyt wysoki. Oczywiście można *łatwo* poprawić bezpieczeństwo tego algorytmu wydłużając klucze – do 2048 lub 3072 bitów. Tyle tylko, że czas generowania podpisu elektronicznego lub deszyfrowania wiadomości przy

wydłużeniu klucza rośnie proporcjonalnie do 3. potęgi długości klucza – a więc wydłużenie klucza 2 razy (do 2048 bitów) powoduje wzrost czasu operacji 8 razy, a trzykrotne wydłużenie klucza (do 3072 bitów) skutkuje 27-krotnym wzrostem czasu operacji. To są wysokie koszty. Dlatego wydaje mi się, że następcą RSA będą algorytmy oparte na tzw. *krzywych eliptycznych* (*EC*). Są to algorytmy posiadające takie same właściwości funkcjonalne, jak RSA (tzn. też mogą być stosowane do generowania podpisów elektronicznych), natomiast narzędzia matematyczne używane w tej klasie algorytmów są zupełnie inne. Algorytmy owe mają ponadto tę ciekawą cechę, że długości kluczy są dużo mniejsze niż w RSA (przy zachowaniu tego samego poziomu bezpieczeństwa) i – co ważniejsze – koszt przetwarzania algorytmów EC przy zwiększaniu poziomu bezpieczeństwa jest dużo niższy, niż w RSA. To powoduje, że choć dla długości klucza algorytmu RSA (1024 bity) i odpowiadającej jej długości klucza algorytmu EC (160 bitów) trudno powiedzieć, żeby algorytmy EC umożliwiały szybsze przetwarzanie podpisów/szyfrowań, to już przy długości klucza RSA równej 3072 bity i odpowiedniej długości klucza algorytmu EC (256 bitów) przewaga staje się już widoczna – nawet 10 razy na korzyść EC. Dlatego chyba mogę powiedzieć, że za *lepsze* – w tym sensie, że bardziej przyszłościowe – uważam algorytmy z rodziny krzywych eliptycznych.

h9: Czy uważa Pan, że maile powinny być szyfrowane?
J.P.: Tak uważam. Choć wiem, że w praktyce to najczęściej nie jest robione. Przypuszczam, że przyczyna leży w stopniu skomplikowania i niewygody służących do tego narzędzi.

h9: Jakie programy szyfrujące mógłby Pan polecić czytelnikom? Dlaczego właśnie one?
J.P.: To zależy, do czego takie programy miałyby służyć. Z narzędziami kryptograficznymi jest trochę tak, jak powiedzmy z narzędziami ogrodniczymi – nie da się powiedzieć, co jest najlepsze – ktoś ma pole i potrzebuje traktora, a ktoś inny potrzebuje grabi, bo ma przydomowy ogródek. Sądzę, że sukces w kontaktach z klientami polega przede wszystkim na tym, żeby na początku poznać rzeczywiste potrzeby konkretnego klienta, a potem dobierać narzędzia – w tym narzędzia kryptograficzne.

h9: Proszę przedstawić osiągnięcia Waszej firmy w dziedzinie kryptografii.
J.P.: Byliśmy jedną z pierwszych firm produkujących i wdrażających oprogramowanie kryptograficzne w Polsce (od 1993 r.). Jesteśmy jedną z niewielu firm w Polsce (a może jedyną), która produkuje oprogramowanie kryptograficzne i utrzymuje się z tej działalności. Jesteśmy jedną z dwóch polskich firm posiadających własne oprogramowanie w pełni obsługujące system PKI (*Centrum Certyfikacji Kluczy*, punkty rejestracji, serwer datowania,

OCSP itd.), oferowane i wdrażane w firmach i instytucjach całej Polski – m.in. Policji, MSWiA, Narodowym Banku Polskim. Nasze oprogramowanie PKI jako jedyne posiada certyfikat ABW uprawniający do przetwarzania informacji niejawnych (do klauzuli *poufne* włącznie).

h9: Jak długi jest cykl wdrożenia do produkcji nowego modelu szyfratora?
J.P.: O to trzeba by było zapytać firmy produkujące sprzęt kryptograficzny (jako, że my produkujemy oprogramowanie) – na przykład firmę COMP S.A., która jest naszym właścicielem. Z tego, co widzimy, zajmuje to jednak co najmniej kilka lat.

Jeśli chodzi o wdrożenie nowego oprogramowania z mechanizmami kryptograficznymi – jest to czas liczony zazwyczaj w miesiącach, czasami w pojedynczych latach – jak np. czas wyprodukowania oprogramowania do obsługi całego systemu PKI.

h9: Współcześnie kryptografia opiera się w dużej mierze o teorię liczb pierwszych. Gdyby, w związku z budową nowych typów komputerów, np. kwantowych czy opartych o biotechnologię, znacznie zwiększyła się moc obliczeniowa komputerów – jak wpłynęłoby to na siłę powszechnie stosowanych algorytmów kryptograficznych, podpisu elektronicznego?
J.P.: W przypadku zbudowania komputera kwantowego współczesne algorytmy klucza publicznego, używane do szyfrowania i do składania podpisów elektronicznych (jak RSA), należałoby wyrzucić do kosza. Trzeba by było wynaleźć zupełnie nowe algorytmy.

h9: Hasła to podstawowy sposób uwierzytelniania i autoryzacji użytkownika, mający wiele wad. W związku z tym, co na tym obszarze będzie stosowane w przyszłości? W jakim kierunku będzie rozwijać się współczesna kryptologia?
J.P.: Sądzę, że będą się rozwijać metody biometryczne, choć i one – zwłaszcza w dzisiejszym, jeszcze niedoskonałym wykonaniu – mają wiele wad.
Myślę też, że nie odejdą w zapomnienie metody dostępu typu *coś wiedzieć i coś mieć* – czyli np. uwierzytelnianie kartami elektronicznymi. Tu wprawdzie też występują hasła (PINy) – ale, z uwagi na brak możliwości *zgadywania* (blokowanie karty po kilku próbach zgadnięcia PINu), nie muszą one być długie i skomplikowane. Uważam także, że rozwijać się będą metody uwierzytelniania wykorzystujące telefony komórkowe.

h9: Jakie jest podejście do bezpieczeństwa informacji w firmie, która chroni swoje zasoby własnym algorytmem kryptograficznym bądź innym o nieopublikowanym algorytmie działania?
J.P.: Sądzę, że jest to nierozsądne. Chyba, że *firma* ta jest instytucją państwową przetwarzającą informacje niejawne i taki tajny algorytm kupiła wraz z certyfikowanym urządzeniem. W normalnym jednak przypadku nie jest to celowe, ponieważ badanie własności algorytmu kryptograficznego wymaga bardzo dużego zaangażowania intelektualnego i finansowego. Trudno sobie wyobrazić, żeby jakąś firmę było stać na to, aby sfinansować badania nad jej własnym algorytmem, tak aby ten algorytm był sprawdzony w stopniu choć trochę zbliżonym do znanych algorytmów wybranych w międzynarodowych konkursach (jak AES). Historia kryptografii uczy, że bardzo wiele (można by wręcz powiedzieć, że znaczna większość) algorytmów, które zostały wymyślone i *wyglądały* bardzo porządnie, została złamana. Niektóre po miesiącach lub latach – a inne np. na tej samej konferencji naukowej, na której je opublikowano.

h9: Czy – z punktu widzenia bezpieczeństwa informacji o wysokim priorytecie, lepiej jest stosować szyfrowanie kilkukrotne, ale prostymi algorytmami, czy też może szyfrowanie przez jeden silny i złożony algorytm kryptograficzny? Dla przykładu, jakie zabezpieczenie stacji roboczych proponowaliby Państwo wdrożyć w firmie, gdzie wykorzystuje się dokumentację projektową, aby pracownicy nie mogli np. kopiować informacji na pendrive'y?
J.P.: Zdecydowanie polecałbym użycie jednego, dobrego algorytmu zamiast kilku wątpliwych. Przy czym *dobry* nie musi zawsze oznaczać *złożony*.

h9: Czy obecnie zaznacza się duże zainteresowanie usługami, produktami i szkoleniami Enigmy? Czy jest to związane z zapotrzebowaniem na tę dziedzinę nauki?
J.P.: Obserwujemy coraz większe zainteresowanie ofertą ENIGMY. Wiąże się to na pewno ze wzrostem świadomości użytkowników w zakresie bezpieczeństwa informacji.

h9: Jak wygląda u Was kwestia zatrudniania nowych pracowników, czy osoby od razu po studiach mają szanse znaleźć u Was pracę?
J.P.: Tak, od początku swojej działalności bardzo chętnie zatrudniamy nawet studentów ostatnich lat. Może dlatego, że wywodzimy się ze środowiska akademickiego, nie boimy się zatrudniać osób, które – posiadając niezbędną wiedzę teoretyczną uzyskaną na uczelni – praktyki muszą się nauczyć już w trakcie pracy.

h9: Jakie trendy panują obecnie na rynku i jak firma na nie reaguje? Jakie mają Państwo plany na najbliższy rok?
J.P.: Mogę odpowiedzieć tylko ogólnie – zamierzamy wprowadzać na rynek nowe produkty – i uzyskać jeszcze wyższe przychody z działalności, niż w roku ubiegłym. Jak co roku, zamierzamy też zorganizować *Krajową Konferencję Zastosowań Kryptografii i Ochrony Informacji ENIGMA 2008*, z międzynarodowym tutorialem Quo Vadis Cryptography. Na tutorial Quo Vadis Cryptography zapraszamy co roku 3-4 osoby spośród najbardziej znanych na świecie kryptologów. Jest to niepowtarzalna okazja poznania tych osób, posłuchania, co mają do powiedzenia na aktualne tematy itd. W zeszłym roku mieliśmy uczestników tutorialu, którzy specjalnie w tym celu przylecieli z zagranicy (w tym jedna osoba z Korei).

h9: Dziękuję za rozmowę.
J.P.: Dziękuję.

Formularz zamówienia prenumeraty korporacyjnej

Prenumerata korporacyjna pozwoli Ci na dowolne powielenie i rozpowszechnianie pism w obrębie Twojej firmy

Zamów, to bardzo proste: wypełnij formularz i dokonaj płatności. Prenumerata korporacyjna to tani i praktyczny produkt stworzony dla nowoczesnej firmy.
W ramach prenumeraty korporacyjnej otrzymają Państwo pismo w wersji elektronicznej w postaci plików *.pdf z możliwością ich dowolnego rozpowszechniania i powielania w obrębie danej firmy. Dodatkowo dwa egzemplarze w wersji tradycyjnej, drukowanej wysyłane pocztą.

Skorzystaj z naszej oferty i zamów już dzisiaj

Prosimy wypełnić czytelnie i przesłać faksem na numer: **(22) 244 24 59** lub listownie na adres:
Software-Wydawnictwo Sp. z o.o., Bokserska 1, 02-682 Warszawa,
Przyjmujemy też zamówienia telefoniczne: **(22) 427 36 53** oraz mailem: **pren@software.com.pl**

Dane firmy zamawiającej	
Imię i nazwisko	Stanowisko
ID kontrahenta* * jeżeli jesteś już klientem firmy Software-Wydawnictwo Sp. z o.o. – wystarczy, że podasz swój numer ID kontrahenta; jeżeli nie posiadasz takiego numeru, podaj swe dane teleadresowe	
Nazwa firmy	
Dokładny adres	
Telefon (wraz z numerem kierunkowym)	Faks (wraz z numerem kierunkowym)
Adres e-mail	Numer NIP firmy

Tytuł	Ilość numerów w roku	Ilość zamawianych prenumerat	Od numeru pisma lub miesiąca	Opłata w zł z VAT (egz)	Wartość w zł z VAT
Software Developer's Journal Miesięcznik profesjonalnych programistów	**12**			900	
Dodatkowy drukowany egzemplarz	**12**			80	
SDJ Extra! Numery tematyczne dla programistów	**6**			900	
Dodatkowy drukowany egzemplarz	**6**			80	
Linux+DVD Miesięcznik o systemie Linux	**12**			900	
Dodatkowy drukowany egzemplarz	**12**			80	
PHP Solutions Dwumiesięcznik o zastosowaniach języka PHP	**6**			900	
Dodatkowy drukowany egzemplarz	**6**			80	
Hakin9 Miesięcznik o bezpieczeństwie i hakingu	**12**			900	
Dodatkowy drukowany egzemplarz	**12**			80	
				W sumie (liczba prenumerat x cena)	

UWAGA: Nadesłanie zamówienia jest jednocześnie zobowiązaniem do zapłaty.

Numer konta: NORDEA BANK POLSKA 46 1440 1299 0000 0000 0391 8238

hack.zone.to

Pamięta ktoś jeszcze ten serwis Grzegorza 'dziuksa' Sterniczuka? Archiwum polskiego hackingu ówczesnych czasów 1997/8- (cracki, seriale, narzędzia, hasła do stron XXX), gdzie mieszkał także /hrabia – jeden z mentorów ówczesnej polskiej sceny phreakingu – autor strony www.tpsasux.com.

Ach... te strony działały i na IE i w lynx'ie bez żadnego ALE. Tak, tak – jakieś dziesięć lat temu elita polskiego Internetu posiadała domeny swoich nielicznych serwerów w do dziś istniejącym serwisie *eu.org*, a inni ratowali się darmowymi kontami WWW i e-mail na *free.com.pl*, *polbox.com*, *geocities.com* oraz *kki.net.pl* (także do dziś on-line), by zaistnieć w Sieci. Na *hack.zone*.to można było znaleźć dosłownie wszystko, co mogło wtedy wydawać się hi-techem polskiej hack sceny. Dziś mamy artykuły, blogi, fora, a na nich ustandaryzowane układy formatów i przeglądarki obsługujące gry zawieszające starsze x86, a kiedyś były tylko pliki FAQ zapisane tylko i wyłącznie w .txt, stylizowane wymyślnie ułożonymi znakami ASCII – tworzące odpowiednie schematy elektroniczne lub stanowiące *nagłówki* wprowadzające – i każdy wiedział, o co chodzi, bez potrzeby interpretacji kolorowych jotpegów. Umieszczone na prostych stronach HTML, na których skrypty Java były egzotyką, składały się na dzienny zestaw odwiedzin w poszukiwaniu nowości (brak RSSów zobowiązywał, ale na szczęście ciasteczka wyświetlały czerwone napisy *Nowość*). Lekturą obowiązkową był P0WERFAQ – czyli P0WER & Lcamtuf HACK FAQ ver 1.1 beta – wydany 11 października 1997 roku, dostępny do ściągnięcia ze strony Centrum Kształcenia Ustawicznego Politechniki Wrocławskiej – który był jednym z porządniejszych dokumentów o hacku. Warte polecenia były także zagraniczne publikacje amerykańskich grup *Legion Of the Apocalypse oraz Legion Of Doom*, tłumaczone na nasz ojczysty język. Budowa każdego FAQ była znana wszystkim. Swoisty wstęp stanowił disclaimer, zaczynający się zawsze od słów: *nie ponosimy żadnej odpowiedzialności za wykorzystanie zamieszczonych informacji czy materiałów do celów niezgodnych z prawem*, co w praktyce oznaczało – i do dziś oznacza – że wszelkie informacje umożliwiające dokonanie w ciągu pięciu minut włamu na serwer były tylko i wyłącznie informacjami, a cała odpowiedzialność spadała na osobę, która stosowała zawarty kod w praktyce. Po tak optymistycznej wiadomości ostrzegającej przechodziliśmy do spisu treści, a w nim znajdowało się wszystko, co wówczas stanowiło uciechę dla duszy: IRC, król poczty Sendmail (wysyłanie fake mejli, wchodzenie dzięki .rhosts), łamanie passwd (oczywiście dzięki John the Ripper, Cracker Jack czy Killer Crack), sposoby wyciągania konkretnych adresów URL z pierwszych stron napisanych w PHP (początki luk WWW), niezbędnik Linuksa (rozmieszczenie kluczowych plików, jak skutecznie czyścić logi, niezapomniane zabawy z komendą finger), zakładanie backdoorów przy pomocy /etc/services oraz /etc/inetd.conf (w swojej prostocie

Dziś mamy artykuły, blogi, fora, a na nich ustandaryzowane układy formatów i przeglądarki obsługujące gry zawieszające starsze x86, a kiedyś były tylko pliki FAQ zapisane tylko i wyłącznie w .txt, stylizowane wymyślnie ułożonymi znakami ASCII – tworzące odpowiednie schematy elektroniczne lub stanowiące nagłówki wprowadzające – i każdy wiedział, o co chodzi, bez potrzeby interpretacji kolorowych jotpegów

działa do dziś!), kody splojtów dla Uniksów, wywalanie Windowsa i wiele, wiele innych. Wtedy IRC – oprócz grup dyskusyjnych i mejlowych – stanowił jedyne źródło komunikacji (w dodatku w czasie rzeczywistym!). Nieujarzmione BitchX, Vampire, Irssi czy EPIC do dziś są pośrednikami czystych komend na rasowym chacie (klikać to se można dziś na Javowych klonach polskich portali do misi1989!). Komendy wędrowały do coraz to nowszych wersji Eggdropów (trzeba było zostawać po lekcjach, by to ustrojstwo dobrze skonfigurować), gdy jeszcze nie pojawiły się Voidy, Ameno, Blowy i Diversy, a wraz z nimi wojny IRC, przejmowanie kanałów (pozdrowienia dla pushera), łączenie botnetów (nie mylić z połączonymi komputerami zombie) i poszukiwanie nowszych, szybszych TeCeeLek (ukłon w stronę fahrena). Ci, którzy podpadali, dostawali Winnukem na port 139 (Win95) lub odpowiednio spreparowanym pingiem (Win98) i był spokój – czasami można było też załatwić czasowy k-line, stanowiący wyrok śmierci dla wielu serwerów uczelnianych. Kiedyś można było dostać bana za sam niewłaściwie napisany nick, wchodząc na #hackpl czy #hackingpl, gdzie wielu newbie na publicznym próbowało doprosić się o jakieś informacje o nauce włamywania. Dzisiaj IRCnet i Freenode nie posiadają już tak przepełnionych kanałów. W dobie kawiarenek internetowych w Sieci zaczęło pojawiać się coraz więcej osób, a słowo *lamer* przeżywało swoje lata świetności – tak jak dzisiaj n00b – w kręgu profesjonalnych graczy. Gumisie gnębiły NASK i TePsę, podmieniając wszystkie możliwe strony miastowych oddziałów w walce o tańsze połączenia internetowe. W noc sylwestrową 1996 dokonano pierwszego włamania na serwer NASK – zmiana strony WWW (przekształcono wówczas nazwę tej organizacji na *Niezwykle Aktywna Siatka Kretynów*; drugie włamanie polegało na umieszczeniu na stronie rysunku z kreskówki i napisu: *Gumisie wróciły*!, a pod linkiem *Zasoby sieciowe w Polsce* opublikowane zostały pliki passwd uzyskane z kilkunastu znanych serwerów w różnych miastach Polski). Kiedy 29 sierpnia 1999 roku (należy przypomnieć, że *przestępcy* komputerowi ścigani są w Polsce dopiero od 1 września 1998 r.) w godzinach wieczornych dokonali włamania na główny serwer *www.tpnet.pl*, podmieniając jego strony WWW, a informacja ta została ogłoszona w wieczornym wydaniu Wiadomości na kanale TVP1, serce od adrenaliny szybciej biło. Dzisiaj ta sama informacja wzbudza śmiech, gdy czas antenowy poświęca się jakiemuś gówniarzowi, który pomylił adres proxy z 127.0.0.1. Szczęśliwe 97' było okresem, kiedy polska scena hakerska przeżywała istny rozkwit. Obok FAQów pojawiły się systematycznie wydawane ZINy (HackPL – napisany w Pascalu, z dostępem na hasło, witający wpadającym w ucho refrenem z utworu Clawfinger – Biggest & The Best; Hackers MAG – wydawany w formacie HTML). Zaczęto pisać masowo skanery portów, narzędzia przeprowadzające ataki FLOOD oraz DoS. Z upływem czasu *hack.zone*.to niestety powoli umierało, a na Sieci zaczęły krążyć powielane informacje, nie wnoszące nic nowego. Nikt już nie chciał dzielić się tak ochoczo posiadaną wiedzą. Równolegle z hack działała scena phreak.zone.to. Odnosząca równie wymierne efekty, co koledzy po fachu. Fala nadeszła po publikacji tekstu spacemana o phreakingu w dodatku *Słowo o komputerach*. Kiedyś wystarczyło włożyć odwróconą łyżkę w automat na żetony. Wraz z pojawieniem się automatów na karty nie było to już takie proste, ale nawet po tym upgradzie niebieskie czy późniejsze srebrne URMETy nie stanowiły żadnych tajemnic. Odpowiednie numery, spreparowane EPROMy, zaprogramowane karty magnetyczne... wszystko dla wszystkich. Wystarczyło mieć gdzie dzwonić. Wówczas powstała nawet grupa Urmet Developers. Wraz z nadejściem ery kart chipowych serwis także powoli zaczął osłabiać swoje osiągi – aczkolwiek dawał sobie jeszcze jakiś czas radę – po czym został zamknięty. Lekka odnowa przyszła z phreak.it, lecz i ten został po dwóch latach wyłączony z obiegu. Niestety dzisiaj to wszystko możemy oglądać tylko w Muzeum internetowej sceny phreakerskiej 1997-2005 (na stronie http://phreaking.eu.org/). Po powolnym zanikaniu *hack.zone*.to powstały dwa konkurencyjne serwisy: *Hacking.PL* (wówczas jeszcze *zieloni* i posiadający coś do przekazania oprócz kilku newsów) oraz *Underground.org.pl* (aktualnie zamknięci we własnym umyśle), wspierający także scenę crack. Istniała jeszcze scena cardingu, lecz ze względu na zacofanie gospodarcze naszego kraju nie było nawet jak wykorzystać lokalnie zakupów. Wszystko odbywało się poprzez zagraniczne serwisy (chodziła plotka, że do 100$ i tak nie będzie opłacało się im ścigać, bo koszta śledztwa będą większe niż straty

Wszystko to składa się teraz tylko na zarchiwizowane pliki, nadające się do internetowych muzeów lub zabaw na VT100. Podobno historia jest matką nauki, a kto ignoruje fakty – jest tylko wskrzesicielem upiorów historii. Czy warto pamiętać? Kiedyś było inaczej, inaczej także będzie za kolejne dziesięć lat, kiedy przyjdzie wspominać, to co dzisiaj – jak stare, dobre czasy. Jednak najważniejsze jest, aby nie tylko pamiętać wszystkie błędy, jakie się popełniło – by nie być ich ponownym autorem – ale także wszystkie te rzeczy, które dodawały smaku i wskazały docelową drogę. Mimo historycznej wartości wszystkich tych zdarzeń, faktów i konkretów – wielu Czytelników tego tekstu przypomni sobie non dozę początków żelaznej netykiety z przesłaniem, z którą tak rzadko można się dzisiaj spotkać. Inni będą musieli używać wyszukiwarek, by rozszyfrować *neologizmy* – i dobrze, bo *nie pamiętasz już o smokach i rycerzach, ja zapomnieć nie zamierzam*. Greetz dla wszystkich, którym kiedyś było dane składać samemu podziękowania.

Patryk Krawaczyński jest studentem Informacji Naukowej na Uniwersytecie Mikołaja Kopernika w Toruniu. W wolnych chwilach stara się prowadzić serwis na temat administracji oraz podstawowych mechanizmów bezpieczeństwa systemu operacyjnego Linux – *www.nfsec.pl*. Jego zainteresowania wiążą się także z ogólnymi zjawiskami zachodzącymi w społeczeństwie, na które oddziałują Technologie Informacyjne.
Kontakt z autorem: *agresor@nfsec.pl*

Roczna prenumerata

tylko 219,-

hakin9 – jak się obronić to ukazujący się w dwudziestu czterech krajach Europy magazyn o bezpieczeństwie. hakin9 porusza sprawy związane z bezpieczeństwem systemów informatycznych – rozpatrywane zarówno od strony osoby naruszającej bezpieczeństwo, jak i osoby, która bezpieczeństwo zapewnia. Radzimy jak skutecznie zabezpieczyć system przed hakerami i wszelkimi innymi zagrożeniami, oprowadzamy Czytelników po tajnikach najpopularniejszych programów antywirusowych, systemów wykrywania włamań, narzędziach, których potrzebuje każdy administrator.

Kontakt

1. Telefon
+48 22 427 36 93
+48 22 427 36 79
+48 22 427 36 53

2. Fax
+48 22 244 24 59

2. Online
pren@software.com.pl

3. Adres
Bokserska 1
02-682 Warszawa
Polska

Zamówienie prenumeraty

Zadzwoń
+48 22 427 36 93
lub
zamów
mailowo!

Prosimy wypełniać czytelnie i przesyłać faksem na numer:
00 48 22 244 24 59
lub listownie na adres:
Software-Wydawnictwo Sp. z o. o.
ul. Bokserska 1
02-682 Warszawa
Polska
E-Mail: *pren@software.com.pl*
Przyjmujemy też zamównienia telefoniczne:
0048 22 427 36 93
0048 22 427 36 79
0048 22 427 36 53

Jeżeli chcesz zapłacić kartą kredytową,
wejdź na stronę naszego sklepu internetowego *www.buyitpress.com.*

Imię i nazwisko .....

Nazwa firmy.....

Dokładny adres .....

.....

Telefon .....

E–mail .....

ID kontrahenta .....

Numer NIP firmy .....

Fax (wraz z nr kierunkowym) .....

□ automatyczne przedłużenie prenumeraty

Prenumerujesz – zyskujesz

- oszczędność pieniędzy
- szybka dostawa
- prezenty
- bezpieczna płatność
ność on–line

Tytuł	Ilość numerów	Ilość zamawianych prenumerat	Od numeru pisma lub miesiąca	Cena
hakin9, jak się obronić (1 płyta CD) Miesięcznik o bezpieczeństwie i hakingu	**11**			**199*/ 219 PLN**

* cena prenumeraty rocznej dla osób prywatnych

ZA MIESIĄC

W następnym numerze między innymi:

Aktualne informacje o najbliższym numerze znajdziesz na naszej stronie www.hakin9.org/pl.

OBRONA

CONTINOUS DATA PROTECTION – NAJLEPSZA OCHRONA DANYCH NA NASZYM KOMPUTERZE

Artykuł Continuous Data Protection – najlepsza ochrona danych na naszym PC podejmuje tematykę ochrony danych na komputerach osobistych (laptopach i desktopach). Ochrona ta nie dotyczy zapobieganiu przed nieautoryzowanym dostępem, lecz przedstawia sposób zabezpieczenia tychże danych przed ich utratą. Uświadamia czytelnikom jak ważna jest ochrona przed ewentualną utratą (jak wiele możemy stracić tracą swojego laptopa, dysk itp.). Przedstawia jedną z najlepszych metod zabezpieczania danych i komentuje sposób działania. Pokazuje jak wiele dostajemy stosując oprogramowanie oparte na technice CDP oraz jak prosto można się zabezpieczyć przed utratą danych. Czytelnik przekonuje się na wielu przykładach jak silną i uniwersalną jest technika CDP w odniesieniu do zabezpieczenia danych przed ich utratą.

NIELEGALNE OPROGRAMOWANIE

To pracodawca ponosi odpowiedzialność za pirackie oprogramowanie używane w jego firmie, niezależnie od tego czy zlecił jego instalację. Pracodawca nie musi nawet wiedzieć, że jego pracownicy używają na służbowych komputerach nielegalnych programów. Rolą pracodawcy jest wprowadzenie takich procedur, żeby wyeliminować możliwość korzystania w zakładzie z pirackich kopii programów. Nie oznacza to jednak, że pracownik może, nie licząc się z konsekwencjami, instalować w firmie nielegalne programy.

NA CD

hakin9.live bootowalna dystrybucja Linuksa
Mnóstwo narzędzi – niezbędnik hakera
Tutoriale – praktyczne ćwiczenia zagadnień poruszanych w artykułach
Dodatkowa dokumentacja
Pełne wersje komercyjnych aplikacji

ATAK

HAKOWANIE APPLE

Apple od dawna jest producentem oprogramowania do odtwarzania multimediów. Jego produkty takie jak QuickTime czy iTunes zyskały sobie ogromną popularność wśród użytkowników, szczególnie, gdy posiadali oni inny produkt Apple – iPod'a. Jednak bezpieczeństwo użytkowników oprogramowania Apple zostało nadszarpnięte – odkryto lukę, która pozwala na przejęcie kontroli nad komputerem ofiary – użytkownika programu QuickTime.

TECHNIKI CYBERPRZESTĘPCÓW

W artykule zostana omówione i zaprezentowane aktualne technik działania Cyberprzestępców. W tym techniki omijania zapor ogniowych i wykradania ważnych informacji. Będzie to bardzo trafny temat z uwagi o przekaz bardzo ważnych informacji na temat działania cyberprzestępców i sposoby unikania takich sytuacji.

NA CD:

- hakin9.live – bootowalna dystrybucja Linuksa,
- mnóstwo narzędzi – niezbędnik hakera,
- tutoriale – praktyczne ćwiczenia zagadnień poruszanych w artykułach,
- dodatkowa dokumentacja,
- pełne wersje komercyjnych aplikacji.

Numer będzie w sprzedaży na początku czerwca 2008

Redakcja zastrzega sobie prawo zmiany zawartości pisma.

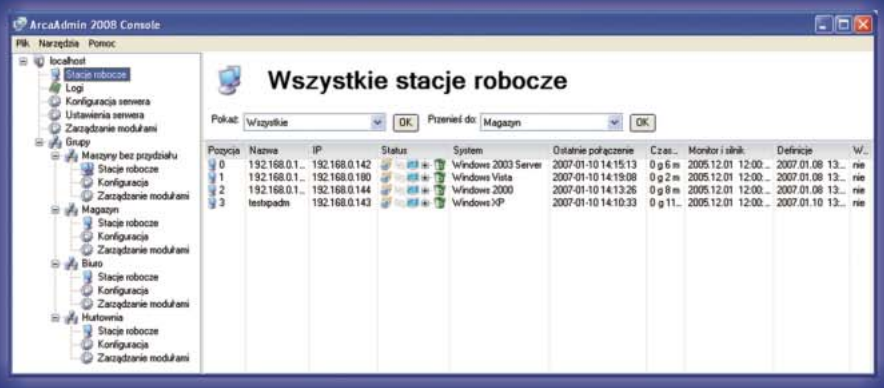
ArcaVir® 2008

BEZPIECZEŃSTWO STACJI ROBOCZYCH I SERWERÓW

Administrator umożliwia zdalną konfigurację programu dla grup stacji w sieci komputerowej, ich bieżące monitorowanie oraz tworzenie centralnych raportów. Możliwość zdalnej konfiguracji oprogramowania dla grup stacji w sieci komputerowej oraz bieżące monitorowanie ich stanu jest jednym z kluczowych wymogów stawianych przez administratorów dużych sieci.

Moduł ArcaAdmin

- narzędzia umożliwiające zarządzanie programem ArcaVir w sieci
- centralna baza danych zawierająca informacje o konfiguracji programu i zdarzeniach mających miejsce w systemie
- efektywne mechanizmy wymiany danych pomiędzy serwerem zarządzania a zarządzanymi stacjami
- centralna konsola zarządzania pozwalają na sprawne wdrożenie i zarządzanie programem ArcaVir nawet w bardzo dużej sieci



ARCABIT

www.arcabit.pl

Producent:
ArcaBit Sp. z o.o.
e-mail: biuro@arcabit.pl, www.arcabit.pl
tel. 022 532 69 00, fax 022 532 69 01
Wsparcie techniczne: 022 532 69 20

COMDOM ANTISPAM
for servers

Możliwości jest wiele,
Rozwiązanie tylko jedno!

COM DOM ANTISPAM

westcoast labs Checkmark

We are members of: ICSA labs

Generalny dystrybutor oprogramowania ComDom AntiSpam - Technologie Internetowe S.A.

www.ti.com.pl
comdom@ti.com.pl
34 361 15 14

www.comdomantispam.pl
kontakt@ti.com.pl
22 743 86 88

COM DOM GOLD PARTNER