Jan A. Januszkiewicz

Ochrona danych w lokalnej sieci komputerowej w ujęciu praktycznym

Słupsk 2004

Spis treści

Wstęp

Cel pracy

Od połowy lat osiemdziesiątych przeżywamy burzliwy rozwój komputeryzacji. W ostatnich latach dołączyła do niego równie dynamiczna ekspansja globalnej sieci komputerowej, zwanej Internetem. Coraz trudniej jest znaleźć komputer, który nie jest w jakiś sposób podłączony do Internetu w sposób stały lub chociaż czasowy, np. przez modem. Zalet takiego stanu rzeczy nie sposób przecenić. Internet stał się kanałem informacyjnym, bez którego w dzisiejszych czasach praktycznie nie sposób się obejść.

Jednocześnie jednak przyłączenie sieci lokalnej do sieci rozległej staje się źródłem zagrożeń dla lokalnego systemu. Internet jest również, niestety, pełen ludzi, którzy z ciekawości, bezmyślności, czy nawet świadomie i złośliwie starają się penetrować obce systemy aby uszkodzić je (osobiście lub przez napisane przez siebie wirusy) albo przejąć nad nimi kontrolę (np. przez podrzucenie do ich wnętrza programów zwanych koniami trojańskimi).

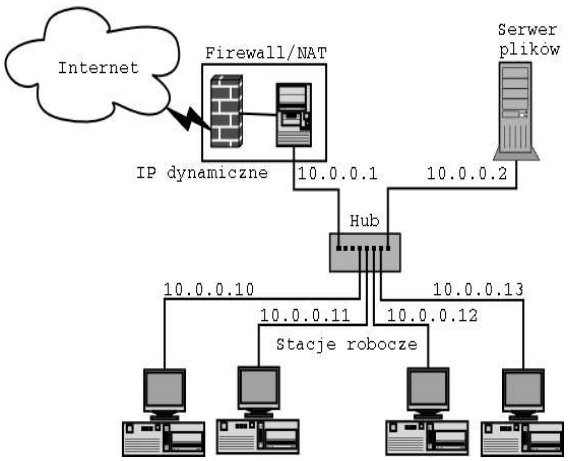
Na szczęście jednak wszechobecność komputerów oraz powszechny dostęp do Internetu przyniosły ze sobą jeszcze jeden skutek. Oto programiści, rozsiani po całym świecie, mogą obecnie tworzyć społeczność, której celem jest hobbistyczna praca nad wspólnymi projektami informatycznymi - między innymi nad tworzeniem darmowego oprogramowania komputerowego. Dzięki możliwości szybkiej wymiany informacji, ludzie o podobnych zainteresowaniach dzielą się pomysłami, pracą i wsparciem technicznym. W rezultacie, programiści są w stanie tworzyć programy nie ustępujące jakością produktom komercyjnym (często nawet lepsze) a przy tym dostępne nieodpłatnie.. W ten właśnie sposób powstało wiele darmowych programów, nierzadko o otwartym kodzie źródłowym, które mogą służyć ochronie danych komputerowych i podwyższaniu bezpieczeństwa systemu. Celem niniejszej pracy jest prezentacja kompleksowego systemu ochrony danych

w lokalnej sieci komputerowej małej firmy, wykorzystującego możliwości oprogramowania pochodzącego z różnych źródeł, a przy tym dostępnego nieodpłatnie.

Przedmiot i zakres pracy

Przedmiotem pracy jest opis zabezpieczeń, które autor wprowadził w sieci komputerowej działającej w firmie, gdzie administruje siecią. Zabezpieczenia te zostały oparte na sporządzonym uprzednio teoretycznym modelu bezpiecznego systemu komputerowego. Model ten był podstawą opracowania polityki bezpieczeństwa, a następnie do wdrożenia jej w praktyce. Jednym z istotnych czynników branych pod uwagę przy zabezpieczaniu sieci był koszt całej operacji, który miał być utrzymany na możliwie najniższym poziomie. Z tego powodu powstała konieczność zrezygnowania np. z ochrony antywirusowej każdej stacji roboczej, a zamiast tego zastosowanie rozwiązań alternatywnych, między innymi – uszczelnienia sieci i skoncentrowaniu się na ochronie wyłącznie plików istotnych dla firmy i składowania ich centralnie na serwerze. Jednocześnie tam, gdzie to możliwe, zdecydowano się wybrać Oprogramowanie Otwarte, jako dające szersze możliwości konfiguracyjne.

Schemat sieci omawianej w niniejszej pracy jest widoczny na Rys. 1.Opisywana biurowa sieć komputerowa składa się z 15 stacji roboczych PC, pracujących pod kontrolą systemu Windows NT, serwera plików i wydruku HP E800, pracującego pod kontrolą systemu Linux Mandrake, oraz dedykowanego firewalla, w postaci komputera PC pracującego pod kontrolą minidystrybucji systemu Linux o nazwie Freesco.



Rysunek 1 Schemat opisywanej sieci komputerowej.

Rozdział 1. Bezpieczeństwo sieci komputerowej

Idealnie bezpieczna sieć komputerowa to sieć, do której nikt nie miałby dostępu. Praktyka użytkowania sieci komputerowych wykazuje bowiem, że na końcu łańcucha potencjalnych zagrożeń zawsze znajduje się człowiek.

„To ludzie włamują się do systemów, podsłuchują, niszczą dane, wprowadzają wirusy, zaniedbują swoje obowiązki lub w sposób nieświadomy przyczyniają się do obniżenia poziomu bezpieczeństwa[1]"

Siłą napędową, czyli motywacją kierującą ludzi do działań niezgodnych z prawem również na polu informatyki, są te same czynniki, które wpływają na niegodne zachowanie się Homo sapiens. Włamując się do systemów komputerowych stosuje się inne narzędzia i środki, ale u podstaw leżą bardzo często nuda, chęć osiągnięcia korzyści materialnych, źle pojęta ambicja czy zawiść.

Oznacza to, że nigdy nie można osiągnąć *bezpiecznej* sieci komputerowej, ponieważ jest ona zawsze przeznaczona dla ludzi – bez nich jej istnienie nie miałoby sensu. Płynie z tego wniosek, że bezpieczeństwo sieci nigdy nie może być stanem stabilnym: zawsze jest tylko procesem, ogółem działań, których celem jest zminimalizowanie czynników niepożądanych w danej teraźniejszości. Działania te polegają na stałym monitorowaniu stanu sieci i udoskonalaniu jej ochrony w miarę, jak pojawiają się nowe metody działania na jej szkodę oraz, zgodnie z filozofią tarczy i miecza, z wykorzystaniem nowych sposobów zabezpieczania danych.

Definicja bezpieczeństwa danych opisuje najczęściej jego trzy podstawowe aspekty: dostępność, poufność i integralność (zwaną również *autentycznością* danych)[2]. Oznacza to, że dane powinny być chronione odpowiednio przed:

- zniszczeniem lub blokadą dostępu do nich,
- ujawnieniem danych, tj. dostaniem się ich w niepowołane ręce,

1 J. Stokłosa, T. Bilski, T. Pankowski: Bezpieczeństwo danych w systemach informatycznych, Wyd. Naukowe PWN, Poznań 2001, s. 19.

2 Ibidem.

- nieuprawnioną modyfikacją[3].

Niektórzy autorzy wyróżniają jeszcze jeden element bezpieczeństwa, tj. i*ntegralność systemu*. M. D. Bauer opisuje ją jako „miarę tego, czy system jest wykorzystywany w sposób zgodny z intencjami jego administratora (to znaczy, czy jest używany wyłącznie przez upoważnionych użytkowników, których przywileje nie przekraczają poziomu, jaki im nadano)[4]" Obejmuje to na przykład wysyłanie w czasie pracy, a więc korzystając z firmowego sprzętu, prywatnej korespondencji e-mail, przeglądanie w prywatnych celach stron internetowych, instalowanie własnego oprogramowania nie związanego z wykonywaną pracą, granie w gry komputerowe i temu podobne zachowania użytkowników.

W świetle powyższego można więc stwierdzić, że zadaniem administratora sieci jest więc zapobieganie wszelkim działaniom, które mogłyby przyczynić się do przerw lub całkowitej destrukcji pracy sieci, wydostawaniu się na zewnątrz danych należących do firmy, nieuprawnionej modyfikacji danych zgromadzonych na serwerze firmy, skasowaniu lub uszkodzeniu plików i programów z których firma korzysta oraz niewłaściwego wykorzystywania firmowych zasobów informatycznych.

3 B. Fisher: Przestępstwa komputerowe i ochrona informacji. Aspekty prawno-kryminalistyczne, Kantor Wydawniczy Zakamycze 2000, s. 146.

4 M. D. Bauer: Linux. Bezpieczeństwo serwerów, Wydawnictwo RM, Warszawa 2003, s. 4

1.1 Podział zagrożeń

Jeśli za linię podziału zagrożeń przyjąć linię graniczną, leżącą na styku sieci firmowej i Internetu, to najbardziej ogólny podział zagrożeń wyróżnia ich dwa zasadnicze typy:

- zagrożenia wewnętrzne, pochodzące od pracujących w sieci osób lub uruchamianych przez nie programów

oraz

- zagrożenia zewnętrzne, mające swoje źródło w innych sieciach, do których sieć firmowa jest przyłączona.

Z punktu widzenia ochrony danych, oba typy zagrożeń są równie istotne, ponieważ ich skutki mogą być do siebie podobne. Ponadto, zagrożenia zewnętrzne i wewnętrzne mogą mieć wspólne elementy i wzajemnie się przenikać.

1.1.1 Zagrożenia wewnętrzne

Przeprowadzone badania wykazują, że większość incydentów związanych z bezpieczeństwem ma swoje źródło wewnątrz organizacji[5]. Wynika to z faktu, że osoby znajdujące się po wewnętrznej stronie granicy mają ułatwione zadanie, ponieważ znają realia. Wiedzą jaką sieć ma strukturę, jak rozłożone są uprawnienia użytkowników, gdzie znajdują się interesujące dane, kto i na jakich warunkach ma do nich dostęp, jak często podejrzane zachowania mają szansę być ujawnione itd. Złoczyńcami „wewnętrznymi" nie muszą być bezpośredni pracownicy firmy. Często bowiem podobną do nich wiedzę mogą posiadać dostawcy, klienci, konsultanci czy osoby, które odbywały kiedyś w firmie praktykę. Dodatkowym czynnikiem stymulującym nieuprawnione działania po wewnętrznej stronie sieci może być specyficzna motywacja – np. frustracja pracownika albo byłego pracownika spowodowana wzajemnymi relacjami służbowymi.

Jednym z kryteriów, według którego można podzielić zagrożenia wewnętrzne, jest świadomy udział użytkownika, bądź jego niewiedza. Zagrożenia

5 J. Stokłosa T. Bilski, T. Pankowski: *op. cit.* s. 19.

świadome obejmują starania mające na celu uzyskanie nieautoryzowanego dostępu do całej sieci lub do części jej zasobów w celu wykorzystania zdobytych tą drogą informacji, zmodyfikowania ich lub do destabilizacji pracy sieci. Większość z nich sprowadza się do przechwycenia lub odgadnięcia hasła, które umożliwia atakującemu przedstawienie się w sieci jako inny użytkownik, z innymi, często większymi niż własne, uprawnieniami.

Zagrożenia nieświadome obejmują wprowadzenie do sieci i uruchomienie w niej wirusów, robaków albo koni trojańskich, czyli programów, których celem działania jest albo uszkadzanie plików znajdujących się na dyskach, kaskadowe rozmnażanie się, powodujące zajmowanie coraz większych zasobów systemu albo otwarcie drzwi do inwazji zewnętrznej (jak jest w przypadku, gdy zainstalowany koń trojański melduje się u swego twórcy lub, w gorszym wariancie, ogłasza swoją obecność publicznie np. na kanale IRC, czekając na przejęcie kontroli przez przypadkowego napastnika).

Warto zauważyć, że ostatnio szerzące się wirusy (I i II kwartał roku 2004) nierzadko łączą w sobie wszystkie wymienione wyżej cechy[6].

Na uwagę zasługuje mnogość możliwości, w jakie nieświadomy użytkownik może wprowadzić szkodliwe programy do systemu. Można bowiem:

- otrzymać je wewnątrz przesyłki e-mail,
- otrzymać je w przekazie plikowym realizowanym wewnątrz transmisji typu IRC, Tlen albo Gadu-Gadu[7],
- pobrać wirusa z serwera ftp lub ze strony internetowej. Wariantem tej możliwości jest niewinne otwarcie przez użytkownika "aktywnej" strony www, tj. skonstruowanej tak, aby pobieranie i uruchamianie pliku odbywało się automatycznie,

6 Np. wirus Korgo, opisywany w serwisach antywirusowych, m. in. na http://www.mks.com.pl

7 Oba te programy są najpopularniejszymi komunikatorami w Polsce. Jednocześnie jednak umożliwiają przesyłanie plików, co może umożliwić użytkownikowi wprowadzenie do sieci programu szkodliwego.

- wprowadzić wirusa do sieci na nośniku danych typu dyskietka lub płyta CD/DVD.

Administrator sieci powinien być świadomy wszystkich tych zagrożeń i podjąć kroki zapobiegające wystąpieniu każdej z wyżej wymienionych okoliczności.

1.1.2 Zagrożenia zewnętrzne

Zagrożenia zewnętrzne to takie, które przenikają do sieci lokalnej z innych sieci, do której jest ona podłączona. Najczęściej siecią zewnętrzną jest po prostu Internet.

Teoretycznie, potencjalni napastnicy zewnętrzni powinni dysponować znacznie mniejszą wiedzą o sieci komputerowej, która jest obiektem ich ataku niż osoby związane z firmą, o których mowa była w poprzednim podrozdziale. Należy jednak zastanowić się, czy zawsze tak jest. W skrajnych przypadkach rzeczywiście włamywacze nie wiedzą o obiekcie swojego ataku nic lub prawie nic[8]. Nie musi jednak tak być zawsze.

Według R. J. Hantona, włamywaczami mogą być: „Uczniowie szkół średnich lub studenci, który mają za dużo wolnego czasu, (...) członkowie grup kultowych [które] zachowują się jak religijne sekty, posługują się pseudonimami zamiast imion i nazwisk i komunikują się charakterystycznym żargonem [oraz nierzadko] obierają konkretne cele społeczne lub polityczne. Nie dziwi fakt, że w wyniku takich ataków w rządowych witrynach internetowych pojawiają się komunikaty i hasła tych grup." Osobną grupę mogą stanowić „szpiedzy korporacyjni [którzy w przeciwieństwie do pozostałych grup] starannie wybierają ofiary i często zbierają informację na ich temat na długo przed wykonaniem ataku.[9]" Ich celem może być chęć uzyskania informacji, które można potem sprzedać konkurencji albo uszkodzenie zasobów ofiary lub zniszczenie jej reputacji, aby zleceniodawca (w

8 O ile stosowane są metody ukrywania wewnętrznej struktury sieci, opisanej jako jeden z elementów polityki bezpieczeństwa w rozdziale „Polityka bezpieczeństwa"

9 R. J. Hontanon: Bezpieczeństwo systemu Linux, Wyd. Mikom, Warszawa 2002, s. 35.

domyśle: konkurent) mógł uzyskać przewagę. Wynika z tego, że zawsze istnieje szansa, że atakujący jest osobą dobrze przygotowaną do wrogich działań, a nawet jeśli tak nie jest – może być bardzo zdeterminowany i mieć silną motywację, a jednocześnie – zdobyć najpierw wszystkie niezbędne informacje o atakowanej sieci.

We wczesnej fazie istnienia Internetu, zagrożenie z jego strony wobec sieci firmowej było dość nikłe, przede wszystkim dlatego, że obecność typowego komputera w Internecie była ograniczona czasowo. Wynikało to z tego, że najczęstszą formą przyłączenia do Internetu był dostęp przez linię telefoniczną. Takie połączenie, realizowane przez modem, zestawiano tylko na moment niezbędny dla odbioru lub wysłania danych (np. poczty), po czym je zakańczano. Czas ten był zwykle zbyt krótki na to, aby np. próbować odgadnąć hasło metodą pełnego przeglądu[10], albo aby w ogóle zlokalizować jakiś komputer i rozpocząć względem niego działania agresywne.

Dziś jednak sytuacja uległa diametralnej zmianie. Po pierwsze, wiele sieci jest przyłączonych do Internetu na stałe. Wydłuża to praktycznie w nieskończoność czas, jaki ma do dyspozycji potencjalny włamywacz. Oprócz tego zwiększyła się wielokrotnie liczba komputerów, użytkowników Internetu, oraz ilość dość przystępnych narzędzi, którymi mogą się oni posługiwać w celu poszukiwania potencjalnych ofiar. Należy też zauważyć, że „ofiara" nie musi być też ostatecznym celem ataku: zdarza się bowiem, że napastnik stara się jedynie przejąć kontrolę nad jej komputerem tylko po to, aby wykorzystać go następnie jako środek do zaatakowania kogoś zupełnie innego:

„Być może szanse zrealizowania tego szczególnego scenariusza dla większości z nas są niewielkie. Ale, czy to się nie może zdarzyć? Okazuje się, że może. Technika ataków przeprowadzanych za pośrednictwem wielu hostów jest powszechna i wielokrotnie się sprawdzała. To samo dotyczy całych zakresów IP, (...)

10 Metoda polega na systematycznym próbowaniu uzyskania dostępu, poprzez podawanie kolejno wszystkich możliwych kombinacji liter i znaków. Istnieje też metoda słownikowa, czyli podawanie systemowi haseł pobieranych z pewnego zbioru, tzw. słownika, który zawiera słowa „typowe", często stosowane przez użytkowników jako hasła. Obie metody są względnie czasochłonne.

które krakerzy przeprowadzają w celu zidentyfikowania podatnych na ataki użytkowników tak w domowych, jak i w firmowych systemach. Istnieje więc duże prawdopodobieństwo, że serwer internetowy należący właśnie do hobbysty będzie regularnie skanowany przez liczne grono potencjalnych napastników w celu znalezienia słabych punktów zabezpieczeń. Nie można wykluczyć, że będzie on skanowany nawet bardziej intensywnie niż niejeden cel o większym znaczeniu.[11]"

Administrator, który sprawdza regularnie dziennik systemowy systemu Linux pełniącego rolę zapory sieciowej, widzi na co dzień, że taki komputer, przyłączony bezpośrednio do Internetu, jest w ciągu doby wielokrotnie skanowany[12] pod kątem dziur programowych i nie zabezpieczonych wejść do systemu.

Przykład takiego skanowania przytoczono poniżej:

```
Jun 26 15:15:57 - rejected - 212.235.41.13:3394 - 212.244.75.161:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3395 - 212.244.75.162:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3396 - 212.244.75.163:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3397 - 212.244.75.164:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3398 - 212.244.75.165:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3400 - 212.244.75.166:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3401 - 212.244.75.167:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3402 - 212.244.75.168:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3403 - 212.244.75.169:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3404 - 212.244.75.170:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3405 - 212.244.75.171:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3406 - 212.244.75.172:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3407 - 212.244.75.173:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3408 - 212.244.75.174:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3409 - 212.244.75.175:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3410 - 212.244.75.176:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3411 - 212.244.75.177:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3412 - 212.244.75.178:135
Jun 26 15:15:57 - rejected - 212.235.41.13:3413 - 212.244.75.179:135
```

11 M. D. Bauer: *op. cit.* s. 6.

12 Skanowanie oznacza tu wysłanie do komputera-obiektu specyficznych poleceń, które mają na celu ustalenie, czy ten komputer odpowie w pewien charakterystyczny sposób. Uzyskana w ten sposób informacja może być przyczółkiem dla przeprowadzenia odpowiedniego ataku.

```
Jun 26 15:15:57 - rejected - 212.235.41.13:3414 - 212.244.75.180:135
```

Jest to zapis dziennika systemowego z zaledwie jednej sekundy pracy zapory internetowej, chroniącej sieć będącą przedmiotem opisu niniejszej pracy. Z zapisu tego wynika, że zapora odrzuciła w ciągu tej *jednej* sekundy (komunikat „rejected") dokładnie dwadzieścia żądań dostępu, pochodzących z nieautoryzowanego komputera umiejscowionego gdzieś w Internecie. Komputer ten, o adresie 212.235.41.13, wykonywał konsekwentne skanowanie sieci w poszukiwaniu systemu podatnego na infekcję, co widać po tym, że starał się uzyskać dostęp do kolejnych komputerów sieci wewnętrznej, poczynając od numeru 212.244.75.161, na numerze 212.244.75.180 skończywszy – co widać w ostatniej kolumnie, wyróżnionej wytłuszczeniem. Nieznany komputer atakował przy tym port o numerze 135, co świadczy o tym, że sam najprawdopodobniej był zainfekowany wirusem MS-Blaster, poszukującym metodycznie nowych ofiar właśnie w taki sposób[13].

W dziennikach systemowych można też znaleźć ślady po usiłowaniach odgadnięcia haseł do systemu. Poniższy, również autentyczny fragment, wskazuje na próbę odgadnięcia hasła poprzez systematyczne próby zameldowania się w systemie pocztowym:

```
04-04-19,18:40:39,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:          abc
04-04-19,18:40:34,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:          abc
04-04-19,18:40:29,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:          abc
04-04-19,18:40:24,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:          abc
```

13 Wirus i jego zainteresowanie portem 135 jest opisane w serwisie antywirusowym firmy F-secure, http://www.f-secure.com/v-descs/msblast.shtml

```
04-04-19,18:40:20,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:           abc
04-04-19,18:40:15,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:           abc
04-04-19,18:40:10,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:           abc
04-04-19,18:40:06,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:           abc
04-04-19,18:40:01,Logon Failure:
         Reason:   Unknown user name or bad password
         User Name:           abc
```

Napastnik wybrał sobie nazwę użytkownika „abc" i starał się uzyskać dostęp do systemu, podając najprawdopodobniej losowe hasła. Zacytowany fragment obejmuje zdarzenia, które miały miejsce podczas jednej minuty pracy komputera-serwera pocztowego. Z pełnego zapisu wynikało jednak, że w tym konkretnym przypadku próby trwały całą dobę. Daje to obraz skali zagrożenia w systemach, które podłączone są do Internetu w sposób ciągły. Wobec takiej sytuacji, ochrona przed zagrożeniami z zewnątrz staje się obowiązkiem.

Oprócz prób siłowego przejęcia kontroli nad systemem, opisanych powyżej, istnieje jeszcze jedno bardzo powszechne zagrożenie. Są nim wirusy komputerowe. Jest to zagrożenie specyficzne o tyle, że twórcy wirusa przeważnie nie zależy na zaszkodzeniu jakiejś konkretnej ofierze. Wydaje się, że celem istnienia wielu wirusów jest samo rozmnażanie się w sieci, oraz atakowanie przypadkowych systemów. Istnieją wirusy, które starają się przemieścić na inne komputery po prostu losując ich adresy, albo rozsyłając się z wykorzystaniem książki adresowej użytkownika (która zawiera nazwiska osób najpewniej w ogóle nie znanych twórcy wirusa, a więc doskonale przypadkowych). Z tego względu można uznać, że na

atak wirusa w jednakowym stopniu narażona jest praktycznie każda maszyna – i każda sieć.

W tym przypadku dodatkowymi czynnikami zwiększającymi zagrożenie są: szybkość, z jaką wirusy mogą rozpowszechniać się po sieci oraz dominacja firmy Microsoft. W sprzyjających warunkach wirus stworzony w Europie może znaleźć się na komputerach w Azji w ciągu kilku minut. Homogeniczność systemów stosowanych na większości komputerów PC, tj. powszechne stosowanie systemu Windows, którego nowo odkrywane słabości stają się natychmiast doskonale znane hakerom i twórcom wirusów na całym świecie powoduje, że przenoszenie się wirusa z maszyny na maszynę staje się trywialnie proste – ponieważ często jeden zastosowany mechanizm infekcji sprawdza się wobec wszystkich egzemplarzy systemu spod znaku Microsoft.

Rozdział 2. Ochrona systemu

2.1 Polityka bezpieczeństwa

Każda firma może posiadać dane dotyczące własnej działalności, których dostanie się w niepowołane ręce mogłoby być uznane za niedopuszczalne z punktu widzenia jej własnego interesu. Informacje takie zasługują na szczególne traktowanie i powinny być chronione przy pomocy niezbędnych sił i środków. Konieczność ochrony pewnych danych w firmie nie jest jednak dzisiaj już tylko kwestią wyboru czy wolnej woli kierownictwa firmy albo administratora sieci. Od 13 listopada 1997 obowiązuje bowiem w Polsce ustawa „Ochrona danych osobowych[14]", która określa „zasady postępowania przy przetwarzaniu danych osobowych oraz prawa osób fizycznych, których dane osobowe są lub mogą być przetwarzane w zbiorach danych(...)[15]". W ujęciu tej ustawy, dane osobowe mogą być przetwarzane wyłącznie ze względu na dobro osoby, której one dotyczą. Oznacza to, że firma, w której posiadaniu się one znajdują, nie może pozwolić na ich nieautoryzowany przepływ do firm lub osób trzecich.

Z tego względu pojawia się potrzeba przyjęcia pewnej strategii postępowania względem tych danych. Strategia ta przyjmuje materialną postać dokumentu zwanego polityką bezpieczeństwa. Polityka bezpieczeństwa jest ogółem przyjętych w danej sieci ustaleń, których celem jest uzyskanie i utrzymanie pożądanego poziomu bezpieczeństwa. Opisuje ona sposób, w jaki wykorzystywane będą konta użytkowników dostępne w systemie oraz kto, i na jakich warunkach, będzie miał dostęp do danych, które system przechowuje lub przetwarza.

Istnieje kilka generalnych zasad dotyczących polityki bezpieczeństwa. Pierwsza z nich głosi, że nie istnieje gotowy wzorzec dokumentu, który firma mogłaby wykorzystać. Plan ochrony danych jest w swej naturze bardzo złożony i bardzo ściśle uzależniony od charakteru działalności danej firmy, jej struktury

14 Z późniejszymi zmianami, datowanymi na 1 maja 2004.
15 Dziennik Ustaw Nr 99.11.95.

organizacyjnej, obiektów, które mają być chronione oraz tzw. mapy zagrożeń[16]. Wszystkie te czynniki mają charakter swoisty dla danej organizacji, dlatego też politykę bezpieczeństwa każda firma powinna opracowywać indywidualnie – zgodnie z odczuciami osób za nią odpowiedzialnych.

Przyjęte jest również, że polityka bezpieczeństwa powinna być przedstawiona w postaci pisemnej i stanowić dokument, który jasno definiuje cel swego istnienia. Wszyscy pracownicy (jak również osoby nowo przyjmowane do pracy) firmy powinni go poznać i zrozumieć. Z tego powodu dokument ten powinien posługiwać się prostym, zwięzłym i zrozumiałym dla użytkowników językiem, ponieważ uzyskanie zrozumienia w oczach pracowników jest jednym z bardzo istotnych elementów praktycznej realizacji ochrony danych.

Definiowanie zabezpieczeń organizacyjnych sprowadza się do precyzyjnego udzielenia odpowiedzi na takie pytania jak np:

- kto i kiedy może mieć dostęp do systemu komputerowego, komu i na jakich zasadach administrator zakłada konto,
- w jaki sposób przebiega identyfikacja,
- czy możliwy jest dostęp do systemu dla osób nie zalogowanych w systemie,
- czy z jednego konta może korzystać kilka osób, dotyczy to zarówno innych pracowników instytucji ale również np. członków ich rodzin czy znajomych,
- na jakich zasadach konta są likwidowane, np. w przypadku osób, które odeszły z pracy,
- jakie zasady określają system haseł, tj. jakie są wymagania co do ich długości, stopnia skomplikowania oraz okresu ich ważności,
- jakie oprogramowanie może być użytkowane w firmie, np. czy użytkownikom wolno samodzielnie instalować programy,

16 Mapa zagrożeń powstaje w wyniku uświadomienia sobie, jakie są możliwe zagrożenia i gdzie znajdują się najsłabsze punkty systemu. W stworzeniu mapy może pomóc lektura literatury fachowej, testy penetracyjne lub skorzystanie z usług firm trzecich.

- czy użytkownicy mają prawo do korzystania z peryferyjnych urządzeń wejścia (np. CD ROM), które często do normalnej pracy biurowej nie są potrzebne, a stanowią potencjalne wrota infekcji (lub instalowania nieautoryzowanego oprogramowania).

Po przyjęciu przez firmę polityki bezpieczeństwa, istotne jest wymuszenie jej przestrzegania, które powinno przebiegać dwutorowo: poprzez środki informatyczne (np. wymuszanie zmiany hasła co zadany okres czasu) oraz poprzez środki administracyjne (np. wpisywanie pewnych obowiązków u osób odpowiedzialnych do profilu ich stanowiska pracy). Skuteczność wprowadzenia w życie projektowanych działań polityki bezpieczeństwa powinna być oceniana periodycznie, w ramach tzw. audytów bezpieczeństwa, przeprowadzanych wewnętrznie w firmie przez wyznaczone do tego osoby lub za pośrednictwem wyspecjalizowanych firm zewnętrznych, które posiadają duże doświadczenie w wyszukiwaniu słabych punktów systemu i sprawdzania, w jaki sposób administracja sieci zamierza sobie z nimi radzić. W przypadku stwierdzenia niezgodności stanu faktycznego z założeniami systemu, powinny być przeprowadzane działania korekcyjne.

2.2 Model zabezpieczeń

Po przyjęciu polityki bezpieczeństwa, administrator powinien stworzyć model zabezpieczeń, który pozwoli na rozbicie złożonego problemu, jakim jest całościowa ochrona danych w sieci komputerowej na szereg prostszych elementów. Takie podejście ułatwi wdrożenie adekwatnych rozwiązań w praktyce.

Naczelną zasadą, jaką należy się kierować przy tworzeniu modelu zabezpieczeń jest *zasada* minimum koniecznego[17]. Głosi ona, że w całym systemie należy nadawać tylko te uprawnienia, które są niezbędne do wykonywanej pracy. W przeważającej mierze dotyczy to użytkowników. Nie powinni mieć oni dostępu do urządzeń, które służą do przetwarzania danych, a jednocześnie nie są ich narzędziami pracy, np. do serwerowni, elementów architektury sieci (takich jak szafki z przełącznikami sieciowymi, przewodami itp.) czy nawet do wnętrza własnego komputera PC. Przede wszystkim jednak nie powinni mieć zbędnych uprawnień do systemu plików. Oznacza to, że administrator powinien nadać im prawa tylko do tych programów i plików z dokumentami, które są rzeczywiście potrzebne w kontekście wykonywanych obowiązków. Takie podejście powinno również mieć swój skutek w zablokowaniu możliwości instalowania przez użytkowników oprogramowania.

System zarządzania uprawnieniami użytkowników jest łatwiejszy do ogarnięcia, jeśli ma strukturę hierarchiczną. W uproszczeniu można to wytłumaczyć tak, że przełożony powinien mieć możliwość wglądu do danych, które przetwarzają jego podwładni, natomiast sytuacja odwrotna powinna być niemożliwa. Kwestia, na ile dozwolona powinna być wymiana danych w strukturach poziomych, tj. pomiędzy pracownikami, jest typowym zagadnieniem zależnym od specyfiki danej firmy i przyjętej w niej polityki bezpieczeństwa. Przeważnie zachowuje się układ zbieżny z organizacją firmy, np. dostęp do dokumentów księgowych mają

17 J. Stokłosa T. Bilski, T. Pankowski: *op. cit.* s. 21.

pracownicy księgowości, do dokumentów związanych z promocją firmy – pracownicy działu marketingu itd.

Zasada minimum koniecznego nie dotyczy jednak tylko użytkowników. Rozciąga się ona bowiem również na sam system. Do minimum bowiem powinna być ograniczona np. liczba punktów, przez które system kontaktuje się ze światem. Takie wydzielone łącze powinno mieć jasno określone zasady wymiany informacji z zewnętrznymi sieciami i zezwalać jedynie na transmisje, które są z nimi zgodne oraz kontrolować ich przebieg. Obejmuje to np. charakter inicjowania połączeń, monitorowanie przesyłanych danych oraz – jeśli to konieczne – pozbawianie ich zawartości, uznanych za niebezpieczne lub niepożądane w świetle polityki bezpieczeństwa (przykładem mogą tu być wirusy lub aktywne załączniki stron internetowych). Urządzenie lub program, który pełni taką rolę nazywa się firewallem. Bardzo często pełni ono jeszcze jedną istotną rolę, tj. ukrywa wewnętrzną strukturę systemu poprzez maskowanie jej w taki sposób, aby z punktu widzenia świata zewnętrznego całość połączeń wyglądała jak, jakby była inicjowana z jednego komputera (właśnie firewalla), a nie z komputerów ukrytych za nim. Taka technologia, zwana NAT (Network Address Translation) jest szeroko stosowana, bowiem atakowanie ukrytych komputerów o nieznanym charakterze jest znacznie trudniejsze, niż komputerów bezpośrednio widocznych w Internecie. Jednocześnie, ponieważ firewall jest jedynym widocznym z zewnątrz komputerem, skupia on na sobie większość ataków, przez co administrator może skoncentrować się na ochronie i monitoringu właśnie tej jednej maszyny. Ponieważ firewall z założenia przeznaczony jest do odpierania ataków, zabezpieczanie go jest dużo łatwiejsze, niż całej grupy komputerów obecnych w firmie, wliczając w to również efekt skali.

Do niezbędnego minimum powinna być również ograniczona liczba uruchomionych w systemie programów czy usług, zwłaszcza dostępnych z zewnątrz, tak jak np. możliwość zdalnej pracy na serwerze czy sprawdzania poczty.

Każde bowiem odstępstwo od tej reguły może spowodować zwiększenie ilości "dziur", czyli słabych ogniw, mogących doprowadzić do zainfekowania sieci wirusami czy przejęcia nad nią kontroli przez osoby trzecie. Dzieje się tak dlatego, ponieważ każdy działający program może posiadać potencjalne błędy, które mogą być wykorzystywane przez włamywacza. Ilość tych błędów w systemie jest oczywiście wprost proporcjonalna do ilości uruchomionych czy zainstalowanych programów. W związku z tym obowiązkiem administratora jest powyłączanie usług, które są zbędne w danym systemie, czyli takich, z których się w danej firmie nie korzysta.

Zabezpieczany system powinien być monitorowany w sposób ciągły. Obejmuje to zarówno sprawdzanie zapisanych w systemie plików programem antywirusowym, jak i regularne przeglądanie dziennika systemowego, aby odpowiednio szybko reagować na wszelkie objawy nienormalnego zachowywania się systemu czy też na zapisy, świadczące o zaistnieniu czynników zagrożenia.

Innym, ważnym aspektem bezpieczeństwa, zapewniającym dostęp do danych jest tworzenie kopii zapasowych. Częstotliwość tworzenia kopii zapasowych, ich zawartość oraz długość czasu ich przechowywania są kolejnymi czynnikami zależnymi od przyjętej polityki bezpieczeństwa i charakteru działalności firmy. Przykładowo, kierownictwo firmy może uznać, że warto przechowywać wyłącznie dane robocze, ponieważ w przypadku awarii firmę stać jest na kilkudniowy przestój związany z ponowną instalacją oprogramowania. Skrajnie odmiennym podejściem jest tworzenie dokładnych obrazów skonfigurowanych w pełni systemów (czy nawet budowanie alternatywnej, zapasowej sieci) aby okres przestoju w przypadku awarii nie trwał dłużej niż kilka minut. Uniwersalną zasadą pozostaje jednak to, że kopie powinny być tworzone regularnie, oraz, że powinno być przeprowadzane okresowe sprawdzanie ich faktycznej przydatności – np. pełne odtwarzanie systemu z wykorzystaniem utworzonej wcześniej kopii awaryjnej[18].

18 Tak naprawdę kopia jest dobra dopiero, gdy jej zawartość dała się odczytać w warunkach awaryjnych i zawierała to,

Wprowadzanie zasady minimum koniecznego najłatwiej jest realizować w praktyce rozpoczynając od zera usług lub uprawnień, a następnie kolejno dodając te, które są potrzebne. Jest to podejście zwane zamkniętym, w odróżnieniu od podejścia otwartego, w którym punktem wyjścia jest nadanie użytkownikom wszystkich możliwych praw, które następnie po kolei się odcina, przykrawając je do założeń polityki bezpieczeństwa. Podejście zamknięte stwarza mniejsze możliwości przeoczenia jakiegoś elementu[19] i uznawane jest za najbardziej bezpieczne[20].

Zagadnienia zawarte w kolejnym rozdziale niniejszej pracy przedstawiają praktyczne wdrożenie opisanych tu elementów zabezpieczania sieci komputerowych.

czego oczekiwano.

19 O braku jakiegoś uprawnienia użytkownik z pewnością szybko administratora poinformuje. Natomiast informacja o istnieniu pewnego zapomnianego "wejścia" do systemu, może do administratora nie trafić przez długi okres bądź wcale.

20 A. Podstawczyński: Linux. Praktyczne rozwiązania, Wyd. Helion, Gliwice 2000, s. 178.

Rozdział 3. Realizacja praktyczna

W ujęciu praktycznym należy przede wszystkim zauważyć, że poszczególne elementy chroniące sieć bardzo ściśle się zazębiają. Przykładowo, antywirusowa ochrona poczty przychodzącej oraz plików na dysku może być realizowana przy użyciu tych samych modułów. Podobnie jest w przypadku firewalla, który może chronić sieć zarówno przed atakami z zewnątrz, jak i przed niektórymi przejawami aktywności koni trojańskich. Z tego względu nakreślona teoretycznie bariera podziału zagrożeń na zewnętrzne i wewnętrzne może mieć nieostre granice.

W ujęciu niniejszej pracy, zagrożenia zewnętrzne oznaczają wszystkie nieautoryzowane próby nawiązania połączenia z siecią, pochodzące z sieci zewnętrznych (Internetu). W celu ochrony przed nimi zdecydowano się na nadanie wszystkim stacjom roboczym tak zwanych prywatnych adresów IP z przedziału 10.0.0.1 - 10.0.0.254, z maską 255.255.255.0, czyli skonfigurowano sieć tak, aby poszczególne komputery wchodzące w jej skład były niedostępne bezpośrednio z Internetu. Jednocześnie umożliwiono połączenia z Internetem (jedynie te inicjowane od wewnątrz, czyli potrzebne użytkownikom do korzystania z zasobów Internetu) za pośrednictwem mechanizmu NAT (Network Address Translation, czyli maskowanie adresów). Wprowadzono też mechanizm filtracji pakietów IP na podstawie ich typu, adresu docelowego i źródłowego oraz wykorzystywanych numerów portów. Zadaniem maskowania adresów oraz filtrowaniem pakietów obarczony został przeznaczony specjalnie do tego celu komputer PC, pracujący jako firewall pod kontrolą systemu Linux Freesco.

W celu dodatkowego zwiększenia bezpieczeństwa, dodatkowy filtr pakietów uruchomiono na serwerze plików i drukarek (Linux Mandrake) konfigurując go tak, aby odrzucane były pakiety nie pochodzące z sieci lokalnej oraz takie, które mają swoje źródło na firewallu. Ma to na celu utrudnienie przejęcia

kontroli nad serwerem plików w przypadku, gdyby napastnik zdołał złamać zabezpieczania firewalla.

Problem zabezpieczenia sieci przed zagrożeniami wewnętrznymi potraktowano wielopłaszczyznowo. Jako sposób eliminacji zagrożeń obrano przede wszystkim uniemożliwienie użytkownikom sprowadzania, instalowania i uruchamiania wszelkich nieautoryzowanych i nie sprawdzonych programów. Zdecydowano się też na skanowanie antywirusowe poczty elektronicznej oraz zablokowanie możliwości wprowadzenia plików do sieci od wewnątrz za pośrednictwem urządzeń wejścia[21] na stacjach roboczych. W polityce bezpieczeństwa firmy ustalono też, że dane podlegające ochronie składowane będą wyłącznie na serwerze plików, gdzie będzie przeprowadzana ochrona antywirusowa oraz kontrola dostępu.

Istnieją dwie drogi, którymi plik z programem może dotrzeć na komputer PC: przez lokalnie włożony nośnik (dyskietka lub płyta CD / DVD) oraz przez sieć. Wykluczenie pierwszej możliwości jest dość proste. Ponieważ użytkownicy nie muszą korzystać z wyżej wspomnianych nośników danych w toku normalnej pracy, w środowisku stacji roboczych (NT 4.0) wyłączone zostały urządzenia *floppy* oraz *CDROM*. Do ich ponownej aktywacji potrzebne byłyby prawa administratora, których użytkownicy nie posiadają. Dokładny opis tego zabezpieczenia znajduje się w rozdziale “Stacje robocze” na stronie 68.

Drugą drogą wejścia w posiadanie plików jest transfer elektroniczny. Aby go uniemożliwić, należy uszczelnić sieć, co można osiągnąć poprzez następujące kroki:

- uniemożliwienie użytkownikom korzystania z usług sieciowych nie związanych bezpośrednio z wykonywaną pracą (np. IRC, FTP, różnego rodzaju „czaty” itd.) poprzez wykorzystanie filtra pakietów oraz brak możliwości instalacji programów klienckich,

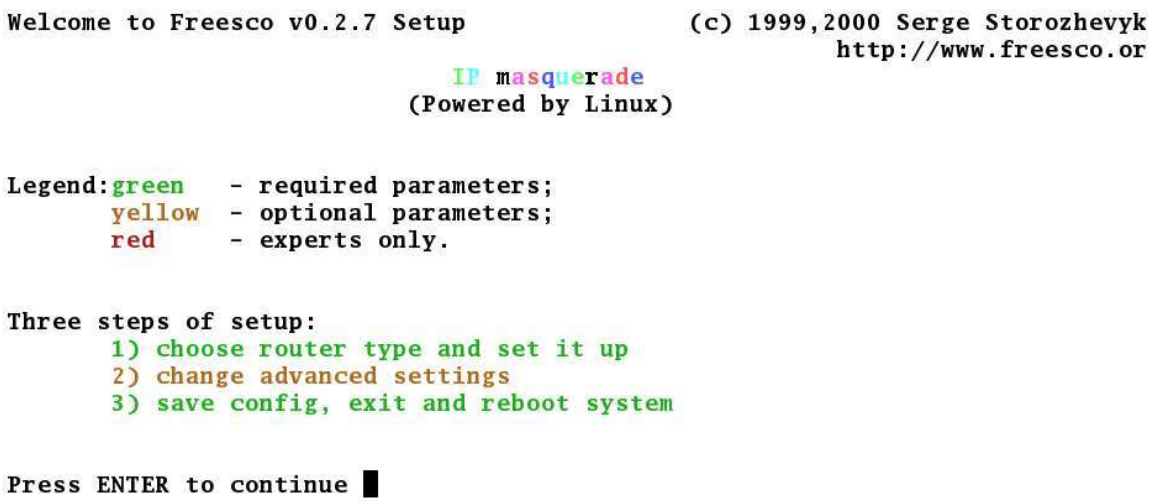
21 Najczęściej stosowanymi urządzeniami wejścia są stacja dyskietek oraz CDROM (oraz klawiatura i mysz).

- dopuszczenie usługi Gadu-Gadu, ale z wyłączeniem możliwości przesyłania plików – z wykorzystaniem filtra pakietów,
- likwidację możliwości pobierania z Internetu plików binarnych typu *.exe, *.rar, *.zip, *.avi, *.mp* i tym podobnych – z wykorzystaniem systemu do filtracji stron internetowych typu JunkBuster, nazywającego się Privoxy,
- zapewnienie skanowania załączników poczty elektronicznej - poprzez wykorzystanie programu MKS_vir w wersji pod Linuksa, którego elementy są tworzone w systemie Open Source, i którego wykorzystywanie na określonych zasadach jest możliwe nieodpłatnie.

3.1 Firewall

Jako firewall, czyli zaporę chroniącą sieć, najlepiej jest wykorzystać osobny komputer PC, skonfigurowany wyłącznie w tym celu. Taki sposób instalacji tego zabezpieczenia jest dość popularny, zwłaszcza w małych sieciach, ponieważ umożliwia łatwe zarządzanie zaporą oraz nie powoduje destabilizacji pracy sieci wewnętrznej gdy niezbędne są jakieś modyfikacje. Firewall zajmuje się tylko obsługą ruchu zewnętrznego oraz filtrowaniem pakietów – i nie spełnia oprócz tego żadnej dodatkowej roli. Na firewallu najlepiej jest nie umieszczać żadnych publicznych usług sieciowych, a tym bardziej żadnych danych, które mogłyby ułatwić włamywaczowi uzyskanie dostępu do głębszych warstw sieci biurowej (np. plików z hasłami dostępu do zasobów na serwerze plików). Nazwa konta administratora firewalla powinna być różna od nazwy konta na serwerze plików oraz posiadać inne hasło.

Jako system operacyjny dla firewalla została wybrana mini dystrybucja Linuksa o nazwie Freesco.



Rysunek 2 Okno powitalne systemu Freesco.

Nazwa Freesco pochodzi od złożenia słów 'free' oraz 'Cisco', ponieważ system ten został pomyślany jako darmowy substytut dla drogich produktów firmy Cisco, produkującej sprzęt sieciowy (w tym routery, mogące pełnić funkcje zapory

sieciowej). Freesco jest rozwijane na zasadzie Open Source[22] i, jak piszą autorzy w dokumentacji, "może sprawić, że tańsze modele routerów Cisco okażą się zbędne". Licencja Open Source sprawiła, że oprogramowanie Freesco jest intensywnie rozwijane. Powstało bardzo wiele programów, współpracujących z tą dystrybucją: począwszy od oprogramowania zliczającego i przedstawiającego w trybie graficznym ruch w sieci, poprzez serwery usług sieciowych (DNS, FTP, HTTP, DHCP, SAMBA, fax), do usług klienckich włącznie (w wersji konsolowej pod Freesco istnieje np. przeglądarka HTML oraz klient komunikatora Gadu-Gadu).

Jednakże ważną zaletą Freesco jest to, że wszystkie te wymienione wyżej pakiety są w wersji podstawowej niedostępne, przez co instalacja bazowa jest bardzo prosta. Macierzysta, w pełni sprawna dystrybucja Freesco mieści się na jednej dyskietce, z której uruchamia się komputer, i która obsługuje przekazywanie pakietów pomiędzy maksymalnie trzema kartami sieciowymi i dwoma modemami (dla połączeń przychodzących lub wychodzących) oraz umożliwia filtrację pakietów. Pozostałe paczki użytkowe można instalować na żądanie, jeśli jest taka potrzeba, po wcześniejszym przeniesieniu już skonfigurowanego i działającego systemu na dysk twardy (jego obecność w wersji podstawowej nie jest wymagana). Należy jednak zaznaczyć, że to właśnie brak we Freesco zbędnych na typowym firewallu usług stanowi o jego dużym marginesie bezpieczeństwa. Nie będąc serwerem, Freesco rzadko może stać się obiektem skutecznego ataku. Wszystko to oznacza, że do instalowania dodatkowych pakietów należy podchodzić bardzo ostrożnie – o czym zresztą autorzy dystrybucji piszą w swojej dokumentacji.

W odróżnieniu od wielu produktów komercyjnych instalacja Freesco jest bardzo prosta. Freesco posiada narzędzie konfiguracyjne, które sprawia, że cała administracja odbywa się z jednego miejsca i jest nieskomplikowana. Przy założeniu, że administrator posiada wymaganą wiedzę na temat konfiguracji swojej

22 "Open Source" oznacza oprogramowanie o otwartym kodzie źródłowym. W praktyce oznacza to, że każda osoba może zmodyfikować ten kod pod kątem własnych celów i wykorzystać go nieodpłatnie.

sieci (adresację IP, adres bramy, parametry własnej sieci dostępowej itd.), całość podstawowej instalacji można ukończyć w przeciągu 20 minut. Potem wystarczy uruchomić ponownie komputer i firewall jest gotowy do pracy. Bardzo łatwo jest też zrobić kopię zapasową systemu – wystarczy wykonać kopię skonfigurowanej już dyskietki.

3.1.1 Instalacja Freesco

Aby otrzymać wersję dyskietkową systemu Freesco potrzebny jest plik z obrazem dyskietki. Jeśli nagrywanie dyskietki będzie miało miejsce w systemie Windows/DOS potrzebny będzie jeszcze program *rawrite.exe*. W systemie Linux jest on zbędny, ponieważ można skorzystać z wbudowanego polecenia *dd*. Zarówno plik obrazu jak i program *rawrite.exe* można pobrać np. z sekcji Download na stronie Polskiej Grupy Freesco pod adresem http://www.freesco.pl.

Po umieszczeniu w napędzie dyskietki, w konsoli linuksowej wydajemy następnie polecenie:

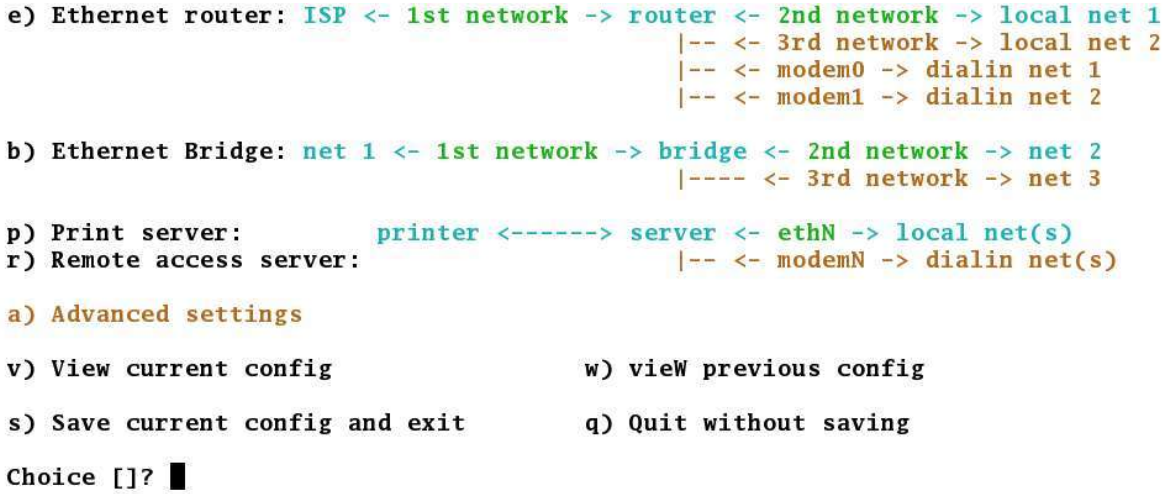
```
dd if=freesco.img of=/dev/fd0
```

które powoduje przepisanie obrazu instalacji z pliku na dyskietkę w napędzie. W systemie Windows/DOS należy włożyć do napędu czystą dyskietkę, uruchomić program rawrite, wskazać mu plik z obrazem i nacisnąć klawisz Enter w celu rozpoczęcia zapisu. Utworzona w ten sposób dyskietka jest dyskietką startującą, czyli, że przy ponownym uruchomieniu komputera (i przy włączonej w BIOSie opcji startowania maszyny z dyskietki) ładuje się z niej jądro systemu Linux, a następnie uruchamiają się procedury routera, umożliwiając zalogowanie się użytkownikowi *root*. Przy pierwszym uruchomieniu, zaraz po zalogowaniu się administratora, uruchomi się automatycznie skrypt setup.

3.1.2 Konfiguracja

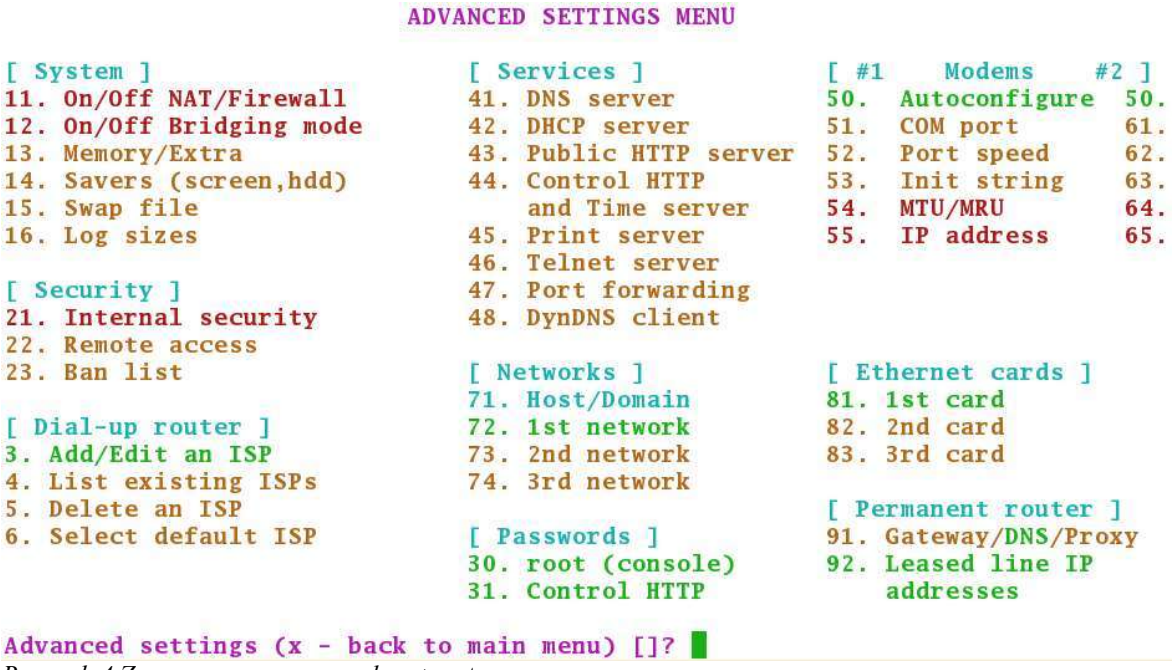
Skrypt setup, pojawiający się jako ekran powitalny systemu Freesco ma postać pełnoekranową, Rys. 2.

Administrator ma możliwość wybrania automatycznego kreatora, prezentującego do wyboru sześć trybów pracy systemu: router z dostępem modemowym wdzwanianym, router na linii modemowej dzierżawionej, router pomiędzy kartami sieciowymi, most, modemowe urządzenie dostępowe oraz zwykły serwer drukarki. Wszystkie te opcje są pokazane na Rys. 33.



Rysunek 3 Kreator skryptu setup.

Istnieje też możliwość przejścia do menu zaawansowanego, gdzie odpowiednie parametry podaje się w dowolnej kolejności – w odpowiednich opcjach. Na uwagę zasługuje bardzo przemyślana numeracja opcji: ich numery są przydzielane według grup tematycznych. Ułatwia to w znacznym zakresie pisanie (i czytanie) pomocy i dokumentacji, ponieważ nie ma możliwości błędnej interpretacji opisu mimo, że autor może posługiwać się numerycznymi skrótami. Oprócz tego opcje wyróżnione są kolorami w zależności od tego, czy należą do grupy opcji obowiązkowych (tj. takich, których skonfigurowanie jest niezbędne dla prawidłowej pracy systemu), warunkowych (jak np. konfiguracja trzeciej karty sieciowej) czy zaawansowanych (których ustawienia mogą mieć wpływ np. na wydajność lub bezpieczeństwo systemu). Okno zaawansowane widoczne jest na Rys. 4



Rysunek 4 Zaawansowane menu skryptu setup.

Podstawowa, czyli niezbędna do pracy konfiguracja routera obejmuje zdefiniowanie parametrów pracy kart sieciowych (opcje 81 i 82) lub karty sieciowej i modemu, jeśli dysponujemy wdzwanianym dostępem do dostawcy Internetu (opcje 50 – 52). Sieciowe karty oparte na magistrali PCI rozpoznawane są automatycznie; Freesco rozpoznaje i obsługuje kilkanaście popularnych modeli kart sieciowych, m. in. opartych na chipsecie 3Com, Intel, Realtek oraz zgodnych z NE2000. Karty oparte na szynie ISA wymagają ręcznego skonfigurowania numeru przerwania oraz adresów wejścia/wyjścia, ale również są obsługiwane. Modem może zostać wykryty automatycznie. Jeśli z jakiegoś powodu się to nie uda, administrator musi ręcznie podać jego parametry pracy (przerwanie, numer portu szeregowego, komendy inicjacyjne itd.).

Po zainstalowaniu sprzętu nadchodzi kolej na zdefiniowanie sieci, w której system ma pracować. Podstawową czynnością konfiguracyjną jest określenie adresów sieciowych IP wraz z odpowiadającą im maską. W sieci komputerowej omawianej w niniejszej pracy, komputery posiadają prywatne adresy sieciowe z przedziału 10.0.0.1 – 10.0.0.10 oraz maskę 255.255.255.0, przy czym

komputerowi Freesco przypisano pierwszy adresów z tej klasy (10.0.0.1). Konsekwentnie, komputery-stacje robocze, w swojej konfiguracji mają adres 10.0.0.1 zdefiniowany jako swoją domyślną bramę.

W skrypcie *setup*, opcje konfiguracji sieci umieszczone są w sekcji [Networks], pod numerami od 72 do 74. Użytkownik podaje kolejno nazwę interfejsu, zdefiniowanego wcześniej przy sprzętowym konfigurowaniu karty sieciowej (np. eth0), a następnie adres IP i maskę. Trzecim, nieobowiązkowym elementem jest uruchomienie na danym interfejsie serwisu DHCP. Jeśli się go uaktywni, Freesco będzie w stanie podawać informację o dynamicznej konfiguracji kart sieciowych stacjom roboczym, uruchamianym w tym segmencie sieci, czyli przydzielać im pierwszy wolny adres IP z określonej puli. Używanie serwera DHCP jest bardzo praktyczne, ponieważ upraszcza administrację pozostałymi komputerami – gdyby pojawiły się jakiekolwiek zmiany konfiguracyjne dotyczące sieci (np. zmiana adresu bramy), wystarczy uruchomić ponownie stacje robocze lub reinicjować ich karty sieciowe, a zmiany zostaną uwzględnione automatycznie. W przypadku konfigurowania DHCP należy po prostu podać, jaki zakres adresów ma być przydzielany stacjom roboczym. W modelowej sieci jest to zakres od 10.0.0.2 – 10.0.0.20. (Serwer DHCP posiada też możliwość dynamicznego przydzielania stacjom zawsze tych samych adresów IP, sprzęgając je z adresem sprzętowym MAC).

Po zakończeniu konfigurowania interfejsów sieciowych, zapisaniu konfiguracji i ponownym uruchomieniu system, router jest gotowy do pracy.

3.1.3 Założenia domyślne filtracji pakietów

Domyślne parametry filtracji pakietów systemu Freesco opierają się na następujących założeniach:

- odrzucane są wszelkie połączenia przychodzące z sieci zewnętrznej,

- przyjmowane są wszystkie połączenia przychodzące z sieci wewnętrznej, które mieszczą się w zakresie dozwolonych usług sieciowych. Jednocześnie na pakietach, wchodzących w skład tych usług wykonywana jest funkcja translacji adresów (NAT) w celu ukrycia rzeczywistej budowy sieci wewnętrznej.

Pierwsze założenie stanowi o tym, że system Freesco jest niedostępny z zewnątrz. Jednocześnie działanie translacji adresów (NAT) powoduje, że każdy pakiet jako adres źródłowy otrzymuje adres interfejsu zewnętrznego routera Freesco. W ten sposób komputery wewnętrzne w ogóle nie są widziane w sieci zewnętrznej, a ewentualny odbiorca pakietów wysyłanych do innych komputerów (mogą nimi być np. zapytania o strony internetowe kierowane do serwerów www) będzie miał wrażenie, że wszystkie one mają swoje źródło na routerze Freesco.

Opisane powyżej działania już same w sobie w znacznym stopniu zwiększają bezpieczeństwo sieci znajdującej się za firewallem. Przede wszystkim, na bezpośrednie działania z Internetu (np. próby włamania) narażony jest tylko jeden komputer, co upraszcza zarządzanie bezpieczeństwem. Natomiast dzięki translacji adresów NAT, sprawa obecności i ewentualnej liczby komputerów położonych głębiej, pozostaje jedynie w sferze domniemań.

3.1.4 Uzupełnianie konfiguracji filtra pakietów

Domyślna konfiguracja firewalla Freesco, chociaż zaprojektowana idealnie pod kątem zagrożeń zewnętrznych, daleka jest jednak od ideału w kontekście zagrożeń pochodzących z wnętrza sieci. Należy zwrócić uwagę, że każde połączenie z systemem zewnętrznym, zainicjowane od wewnątrz, będzie przez router przyjmowane. Takie podejście ma zasadniczą wadę. Najwyraźniej autorzy konfiguracji założyli, że użytkownicy sieci wewnętrznej zawsze wiedzą, co robią, w związku z tym należy im pozwolić na praktycznie dowolne operacje. Istnieją jednak programy szkodliwe, które, umieszczone wewnątrz sieci, mogą starać się nawiązać połączenie na zewnątrz, aby zameldować o sobie swemu twórcy i oddać

mu we władanie komputer, na którym się znajdują. Takie programy noszą nazwę koni trojańskich. Nietrudno zauważyć, że wobec takich zachowań, domyślna konfiguracja firewalla będzie nieskuteczna. Działanie konia trojańskiego, jako znajdującego się wewnątrz, zostanie potraktowane jak każda inna usługa, której zażąda użytkownik i połączenie zostanie przyjęte.

Dlatego też należy na początek zablokować wszystkie połączenia wychodzące, a następnie odblokować tylko te, które są niezbędnie potrzebne użytkownikom sieci do pracy. Takie podejście bardzo trafnie oddaje następująca metafora: zapora sieciowa jest murem, odgradzającym od Internetu, w którym administrator nawierca małe dziurki, pozwalające użytkownikom na korzystanie jedynie z uważnie wybranych usług[23].

Mechanizmem filtracji pakietów, który zastosowany jest w systemie Freesco, zbudowanym na jądrze 2.0, obsługuje się poleceniem ipfwadm. Jest to polecenie, które umożliwia tworzenie i usuwanie reguł filtracji. Poszczególne reguły łańcucha filtracyjnego tworzą tabelę. Nadchodzący pakiet jest porównywany z kolejnymi wierszami tabeli i jeśli któryś z nich pasuje, bieżąca reguła reguła zostaje zastosowana. Zgodnie z jej brzmieniem, pasujący pakiet może być przyjęty (przepuszczony) przez filtr (accepted) lub zablokowany i usunięty (rejected albo dropped). Jeśli pakiet został dopasowany do którejś reguły, kolejne reguły w szeregu nie będą już sprawdzane względem danego pakietu. Oprócz zdefiniowanych przez administratora reguł istnieje reguła domyślna zwana polityką (policy), stosowana dla wszystkich pakietów, wobec których reguły zdefiniowane przez administratora nie pasują. W systemie Freesco, domyślną regułą dla łańcucha filtracji pakietów przychodzących (Input) jest odrzucanie (reject). W ten sposób odrzucone będą wszelkie pakiety, z wyjątkiem tych, którym administrator zezwoli na przepływ. Oprócz domyślnej polityki, ustawionej na odrzucanie pakietów, skrypt setup wstawia do łańcucha Input regułę, dopuszczającą wszystkie pakiety, mające

23 A. Podstawczyński: Linux w sieci, Wyd. Helion, Gliwice 2002 s. 154.

swoje źródło w sieci wewnętrznej. Jak zostało wspomniane wyżej, należy zastąpić ją regułą blokującą pakiety, a następnie przepuścić przez filtr tylko wybrane usługi.

Składnia polecenia ipfwadm składa się zasadniczo z następujących elementów: nazwa łańcucha, komenda, akcja, protokół, adres źródłowy, port, adres docelowy, port. Przykładowo, polecenie

```
ipfwadm -I -i accept -P tcp -S 0.0.0.0/0  -D 212.244.75.161 80
```

oznacza wstawienie (-i) reguły do łańcucha Input (I), nakazującej przyjmować (accept) pakiety protokołu (P) TCP, pochodzące z dowolnego adresu źródłowego (S) i skierowane (D) do hosta 212.244.75.161 na port 80.

Skryptem, w którym administrator wpisuje swoje własne reguły jest jeden ze skryptów startowych, o nazwie rc_user. Miejsce przeznaczone w tym skrypcie na dokonywanie wpisów jest oznaczone następująco:

```
# Add your custom firewall rules here. Warning, incorrect rules could
# leave your system insecure[24].
```

Zgodnie z założeniem polityki bezpieczeństwa opisanej w niniejszej pracy, należy najpierw zablokować wszystkie pakiety:

```
ipfwadm -I -i reject -P tcp -S 0.0.0.0/0  -D 0.0.0.0/0
```

Powyższe polecenie blokuje absolutnie wszelkie pakiety (adres 0.0.0.0/0 oznacza "każdy adres"), a więc m. in. uniemożliwia użytkownikom korzystanie z usług, które nie są niezbędne w wykonywanej pracy tj. FTP, IRC czy Kazaa. Uniemożliwia też przesyłanie plików protokołem Gadu-Gadu, ponieważ ewentualny nadawca nie będzie mógł nawiązać bezpośredniego połączenia z odbiorcą, które jest do tego wymagane.

Następnym krokiem będzie umożliwienie korzystania z trzech usług: WWW, działającej na porcie 80, poczty elektronicznej umieszczonej na serwerach zewnętrznych, czyli protokołu POP3, działającego na porcie 110 oraz z usługi Gadu-Gadu, która operuje na portach z przedziału 82-89. Pierwsze dwie z podanych tu

24 "Dodaj tutaj swoje własne reguły. Uwaga, nieprawidłowe reguły mogą narażać bezpieczeństwo systemu"

usług nie będą jednak dostępne w stanie surowym – zostaną bowiem poddane dodatkowemu filtrowaniu, o czym mowa będzie w rozdziałach *3.2* i *3.3*. Z kolei usługa Gadu-Gadu, w której użytkownicy mogą łączyć się jedynie z serwerem, a nie bezpośrednio ze sobą, będzie umożliwiała jedynie wymianę komunikatów pisemnych – bez możliwości przesyłania plików oraz prowadzenia rozmów głosowych. Ponieważ filtracja poczty i stron www będzie odbywać się na serwerze plików i drukarek Linux Mandrake, noszącym adres sieciowy 10.0.0.2, reguły dopuszczające ruch na portach 80 i 110 nie będą umożliwiały kontaktu ze światem wszystkim stacjom roboczym, a tylko serwerowi, i będą miały postać:

```
ipfwadm -I -i accept -P tcp -S 10.0.0.2/24  -D 0.0.0.0/0 80
ipfwadm -I -i accept -P tcp -S 10.0.0.2/24  -D 0.0.0.0/0 80
```

podczas gdy reguła dopuszczająca ograniczone użytkowanie Gadu-Gadu będzie wyglądać następująco:

```
ipfwadm -I -i accept -P tcp -S 10.0.0.0/24  -D 0.0.0.0/0 82:86
```

W wyniku działania skryptu setup oraz komend wydanych przez administratora, tabela reguł dla filtra Input przedstawiać się będzie następująco:

```
IP firewall input rules, default policy: reject      n/a
type  prot    source          destination     ports
acc   tcp     10.0.0.2/24     0.0.0.0/0       * -> 80
acc   tcp     10.0.0.2/24     0.0.0.0/0       * -> 110
acc   tcp     10.0.0.0/24     0.0.0.0/0       *-> 82:86
rej   tcp     0.0.0.0/0       0.0.0.0/0       * -> *
```

Powyższa tabela oznacza następujące działanie systemu. Filtr sprawdzi, czy pakiet pochodzi z komputera 10.0.0.2 i czy portem docelowym jest 80 lub (w kolejnej linii) port 110. Jeśli nie, sprawdzi (w kolejnej linii) czy docelowym portem jest port z zakresu 82-86. Jeśli tak będzie – pakiet zostanie przyjęty. Jeśli jego adresem źródłowym, lub portem docelowym będzie jakikolwiek inny adres – pakiet zostanie odrzucony.

Pełna lista reguł filtracyjnych, zawierająca reguły opracowane przez autorów dystrybucji jest dużo obszerniejsza i bardzo starannie przemyślana. Gwarantuje na przykład odrzucanie pakietów, które przychodzą od strony Internetu, mając jednocześnie fałszywy adres źródłowy – tj. ustawiony tak, aby udawać pakiet pochodzący z wnętrza sieci[25]. Zapewnia też zapisywanie do dziennika systemowego wystąpień wszystkich pakietów, które zostały zablokowane, co umożliwia administratorowi nadzór nad kondycją zapory sieciowej.

Opisane w niniejszym rozdziale filtrowanie pakietów radykalnie ogranicza ruch sieciowy, zgodnie z zasadami przyjętymi w polityce bezpieczeństwa. Dalszym krokiem będzie filtrowanie dozwolonych usług przy pomocy bardziej wyspecjalizowanych narzędzi.

25 Jest to technika często stosowana przez włamywaczy. Jej angielska nazwa brzmi *spoofing*.

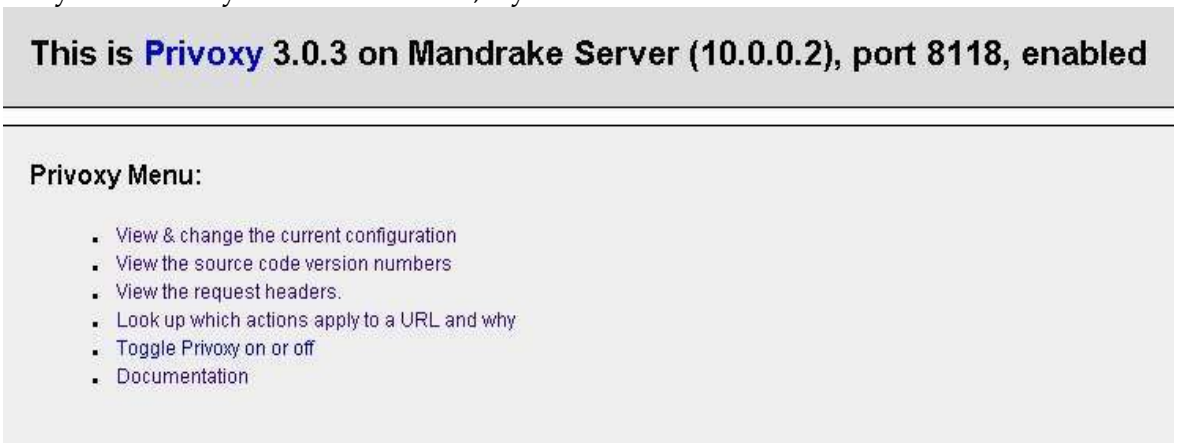
3.2 Privoxy

Jako filtr dla usługi www można zastosować program Privoxy. We wcześniejszych wersjach program ten nosił nazwę JunkBuster – co znakomicie określało jego rolę, tj. Wycinanie śmieci ze stron internetowych. Privoxy działa w warstwie aplikacji stosu protokołów TCP/IP. W istocie jest to program typu proxy server, który posiada bardzo szerokie możliwości filtracyjne. Pozwala na ochronę prywatności, odsiewanie zawartości stron www, na zarządzanie tzw. ciasteczkami, czyli plikami zapisywanymi przez serwery www na komputerach użytkowników, na kontrolę dostępu do zasobów http oraz na usuwanie ze stron reklam, banerów reklamowych, wyskakujących okien i innych niebezpiecznych czy denerwujących czynników, które można nieświadomie pobrać z witryn www. Privoxy nadaje się do użytku zarówno przez jednego użytkownika na lokalnym komputerze, jak i przez wielu, łączących się przez sieć. W modelowej sieci zdecydowano się na tę drugą możliwość.

Ideowy schemat działania programu jest prosty. Privoxy lokuje się na wybranym komputerze jako usługę, nasłuchującą na wybranym porcie (domyślnie jest to 8118). Następnie przeglądarki stron internetowych (głównie Internet Explorer) na poszczególnych stacjach roboczych konfiguruje się tak, aby do nawiązywania połączeń w celu przeglądania stron posługiwały się maszyną z zainstalowanym Privoxy jako serwerem pośredniczącym (proxy). Gdy użytkownik wpisze adres strony internetowej, tak skonfigurowany filtr odbiera od przeglądarki zapytanie http. Jeśli adres docelowy http zawarty w zapytaniu (albo zawarty w nim ciąg znaków) jest obecny na liście adresów zakazanych, Privoxy nie przekazuje połączenia dalej. Zamiast tego, odsyła do przeglądarki komunikat o błędzie, którego brzmienie administrator może dowolnie konfigurować.

Jeśli natomiast adres docelowy nie jest przez administratora zablokowany, Privoxy dokonuje "zwrotu o 180 stopni" i w imieniu przeglądarki

wysyła identyczne zapytanie do serwera witryny www w Internecie. Po pobraniu żądanej strony rozkłada ją na czynniki pierwsze, tzn. interpretuje jej strukturę (podobnie, jak czyni to przeglądarka) i wykonuje na niej różnorakie akcje, zdefiniowane przez administratora w plikach konfiguracyjnych. Obejmuje to np. usuwanie ze strony skryptów, aktywnych kontrolek, banerów reklamowych itd. Następnie tak spreparowaną stronę odsyła do przeglądarki, z której na początku przyszło żądanie http. Cała operacja odbywa się w czasie rzeczywistym i jest przezroczysta dla użytkownika końcowego. Oczywiście obróbka strony musi zajmować pewien czas, ale z reguły jest on nadrabiany przy wyświetlaniu strony, ponieważ kod html pozbawiony banerów reklamowych wyświetla się w przeglądarce szybciej. W ten sposób zawartość stron www zostaje pozbawiona zawartości, które administrator uznaje za niebezpieczne. Fakt, że filtr tkwi pomiędzy przeglądarką a światem zewnętrznym, pozwala mu na wyświetlanie własnej wizytówki i innych komunikatów, Rys. 5.



Rysunek 5 Ekran zgłoszeniowy programu Privoxy.

Program Privoxy może być stosowany jako zabezpieczenie z sieci z kilku powodów. Część z nich wymieniona jest powyżej; są to mechanizmy usuwania aktywnej zawartości stron (m. in. skryptów), które mogą np. uruchamiać automatyczne pobieranie lub uruchamianie pliku binarnego ze specjalnie spreparowanych witryn. Dodatkowym elementem ochrony jest jednak możliwość definiowania filtrów przez administratora. Może on np. zabronić użytkownikom

pobierania plików binarnych innych niż .jpg, .gif i .bmp (czyli obrazków). W ten sposób użytkownicy nie będą mogli wprowadzać do wnętrza sieci programów z rozszerzeniami .com, .exe, .scr lub .pif, będących potencjalnymi nosicielami wirusów oraz archiwów .zip, .rar lub .arj, których zawartość może być problematyczna, czy wreszcie plików z muzyką lub filmami, np. .mp3, .mpg albo . avi, nierzadko będących nosicielami treści pirackich i zajmujących niepotrzebnie przestrzeń dyskową. Dodatkowo można zakazać otwierania stron interaktywnych: np. bardzo dobrze działa filtrowanie w Privoxy stron z usługą typu "czat", ponieważ znakomita większość witryn, które ją zawierają ma w adresie URL ciąg znaków "czat" lub "chat", a ponadto wymaga języka Java, który również można blokować. Warto zauważyć, że taki sam ciąg znaków pojawi się w adresie URL wysyłanym z przeglądarki w momencie naciskania przycisku "Szukaj" dowolnej wyszukiwarki. Strona z wynikiem takiego wyszukiwania będzie więc również podlegała odfiltrowaniu, przez co użytkownicy nie będą tracić czasu na *poszukiwanie* zakazanych w firmie usług sieciowych. Podobnie można postąpić z programami do wymiany plików typu Kazaa czy Morpheus. Programy te używają różnych portów do połączeń, co ciężko jest przewidzieć i zablokować. Jak dotąd jednak posiadają jedną wspólną cechę: mechanizm wyszukiwania plików oparty jest na interfejsie www, ponieważ z niego żyją reklamodawcy. Zablokowanie głównej strony serwisu uniemożliwia więc wyszukiwanie plików, czyli pośrednio – pobieranie ich. W ten sposób korzystanie z tych usług staje się niemożliwe lub wysoce utrudnione.

Teoretycznie, użytkownik mógłby tak ustawić swój system, aby nie korzystał z komputera–proxy do przeglądania stron, próbując się łączyć bezpośrednio. Tu jednak do akcji wkracza filtr pakietów, czyli zapora sieciowa Freesco, omówiona w poprzednim rozdziale. Odrzuca ona pakiety przychodzące z innych komputerów, a zezwala jedynie na łączenie się z hostami internetowymi na port 80

wyłącznie za pośrednictwem maszyny z Privoxy. Widać tu wyraźnie, że oba mechanizmy filtrowania, prowadzone na dwóch oddzielnych warstwach sieciowych, uzupełniają się wzajemnie.

3.2.1 Instalacja Privoxy

Dla systemu Linux Mandrake, Privoxy jest dostępny w postaci pakietu RPM, czyli skompilowanych plików binarnych, które pobiera się z Internetu (z serwerów oprogramowania) i instaluje przy pomocy Instalatora oprogramowania (RPM Drake) dostępnego z menu powłoki graficznej lub z linii poleceń jako rpm. Pakiet zainstalowany w modelowej sieci to wersja RPM 3.0.0-1 mdk.

Skrypt instalacyjny kopiuje niezbędne pliki z pakietu RPM do odpowiednich lokalizacji na dysku twardym: wykonywalny plik demona Privoxy do katalogu /usr/sbin/privoxy, skrypt startowy do etc/rc.d/init.d/privoxy a pliki konfiguracyjne do /etc/privoxy. Po zainstalowaniu, demon uruchamia się przy każdym starcie systemu i czeka na połączenia na domyślnym porcie.

3.2.2 Konfiguracja Privoxy

Program można konfigurować przez edycję plików config, default.action, default.filter, standard.action oraz user.action. Pliki konfiguracyjne mają postać tekstową i posiadają wewnątrz wyczerpujące komentarze. Taka edycja nie jest jednak potrzebna, ponieważ Privoxy oferuje bardzo dogodne narzędzie diagnostyczno-konfiguracyjne oparte o protokół http. Po ustawieniu dowolnej przeglądarki do korzystania z Privoxy, wystarczy wpisać adres http://config.privoxy.org. URL o takiej postaci zostanie przechwycony przez program i zamiast z Internetem, przeglądarka połączy się z generowaną lokalnie stroną, będącą centrum sterowania programem. Można stąd obserwować stan programu, włączać i wyłączać program bez potrzeby rekonfigurowania stacji klienckich oraz zarządzać jego konfiguracją przy pomocy stron http, będących w istocie formularzami.

Język, którym tworzy się reguły filtracyjne w programie Privoxy jest zgodny z wyrażeniami regularnymi Perla. Struktura wzorów, do których dopasowywane są adresy URL składa się z dwóch elementów, o postaci <domena>/<ścieżka>. Obie te części są opcjonalne; innymi słowy można ustawić wyzwalacz filtra albo na domenę, albo na ścieżkę, albo na oba człony adresu jednocześnie. Konsekwentnie, symbol / pasuje do każdej domeny i dowolnej ścieżki. Poniżej garść przykładów:

www.przykład.com/

Taki ciąg znaków oznacza dopasowanie do domeny www.przykład.com, niezależnie od tego, jaki dokument jest z niej pobierany,

www.przykład.com/index.html

Taki ciąg znaków oznacza dopasowanie do domeny www.przykład.com ale wyłącznie, jeśli jest z niej pobierana strona index.html. Inne strony (np. dalsze odnośniki zawarte w stronie głównej) nie będą już pasować do filtra.

/index.html

Taki ciąg oznacza dopasowanie do dokumentu o nazwie index.html, niezależnie od tego, skąd jest on pobierany.

Dodatkowe opcje dopasowania filtrów dostępne są przy wykorzystaniu znaku kropki:

.przykład.com – będzie pasował do dowolnego adresu www, *kończącego się* na "przykład.com".

www. - będzie pasowało do dowolnego adresu, rozpoczynającego się od "www" .

.przykład. - będzie pasowało do dowolnej domeny, która *zawiera* w nazwie ciąg "przykład".

Dodatkowo, można stosować typowe znaki globalne, np.

reklam*.przykład.com – będzie pasować do domen reklama.przykład.com, reklamy.przykład.com, reklamowanie.przykład.com i tak dalej. Z kolei ciąg:

reklama[0-9].przykład.com – będzie pasować do domen reklama1.przykład.com, reklama2.przykład.com, reklama3.przykład.com i tak dalej.

Typowa składnia z Perla, jest dostępna dla filtracji dotyczącej dokumentów, a nie domen. W ujęciu niniejszej pracy jest ona najważniejsza, ponieważ celem zastosowania programu Privoxy nie jest ograniczenie użytkownikom dostępu do wybranych domen, a raczej uniemożliwienie pobierania plików i dokumentów, których nazwa, a więc i domniemana zawartość, spełnia określone warunki. W języku Perl:

. - pasuje do dowolnego, pojedynczego znaku, np. “a”, “A”, “;”, “8” itp.

? - oznacza, że znak (lub wyrażenie) poprzedzające dopasowany będzie zero razy, albo jeden raz (albo – albo).

+ - oznacza, że znak (lub wyrażenie) poprzedzające dopasowany będzie jeden lub więcej razy

* - oznacza, że znak (lub wyrażenie) poprzedzające dopasowany będzie zero lub więcej razy.

\ - znak specjalny, określający, że znak postawiony zaraz za nim ma być traktowany dosłownie, a nie jako znak specjalny, czyli w ciągu przykład\.com kropka będzie oznaczać wyłącznie kropkę, a nie “dowolny, pojedynczy znak”, jak w definicji powyżej.

[ ] - nawiasy kwadratowe ograniczają znaki, z których jeden może być pasującym. Np. [0-9] oznacza dopasowanie dowolnej cyfry, a [a-z] – dowolnej litery. Zakres [0-Z] oznacza dowolną cyfrę lub literę. Oczywiście symbole można łączyć,

dlatego [0-9]+ będzie pasować do dowolnej cyfry, występującej jeden lub więcej razy.

() - nawiasy okrągłe grupują wyrażenia; również wielokrotne.

| - pionowa kreska oznacza alternatywę. Np. ciąg "(dobry | zły) przykład" będzie pasować do wyrażenia "dobry przykład" ale i do wyrażenia "zły przykład".

Składnia wyrażenia, która będzie blokować pobieranie plików o dowolnej nazwie, i kończących się na .exe będzie wyglądać następująco:

/.*[0-z]*\.exe

Interpretacja jest następująca:

Ciąg zaczyna się od znaku / a więc dotyczy nie domeny, tylko dokumentu. Zaraz za nim stoi wyrażenie regularne ".*", oznaczające jakikolwiek znak (.) występujący zero lub więcej razy. Celem jego umieszczenia jest uniezależnienie dalszej części wyrażenia od postaci i głębokości ścieżki, z której pobierany mógłby być plik. Kolejnym elementem ciągu jest "[0-z]*", które należy interpretować jako dowolną liczbę znaków, będących cyframi od 0 do 9 lub literami od a do z. Ta część wyrażenia ma odpowiadać dowolnej nazwie pliku. Ostatnim segmentem ciągu jest "\.exe" - czyli dosłownie potraktowana kropka, po której następuje ciąg exe, oznaczający zwykle program wykonywalny.

Po jednym ciągu filtracyjnym jak wyżej, należy przygotować dla każdego rozszerzenia plików, których pobieranie miałoby być zablokowane, np.

/.*[0-z]*.exe

/.*[0-z]*.com

/.*[0-z]*.zip

/.*[0-z]*.rar

/.*[0-z]*.arj

/.*[0-z]*.swf

i tak dalej. Warto też zauważyć, że wpisanie rozszerzenia .jpg umożliwiłoby użytkownikom korzystanie ze stron internetowych, ale bez grafiki w plikach jpg. Zablokowanie ciągu .exe jest też znakomitą ochroną przed zainstalowaniem sobie, przez nieuświadomionego użytkownika, programu zmieniającego numer dostępowy do Internetu na kosztowny 0-700, czyli tak zwanego dialera. Ekran ustawiania filtrów widoczny jest na Rys. 6.

Rysunek 6 Ekran definiowania filtrów Privoxy.

Standardowo, program posiada bardzo wiele filtrów, będących odpowiedzią na coraz bardziej agresywną reklamę, obecną w kodzie stron internetowych. Filtry te (jak również i odpowiadające im ciągi – wyzwalacze) zawierają m. in. znane adresy internetowe serwerów, z których pochodzą reklamy. Filtracja ta jest domyślnie uaktywniona i jako taka nie wymaga edycji, a jedynie okresowego uaktualniania "czarnej listy" adresów internetowych firm wyświetlających reklamy. Użytkownik programu może użyć trzech poziomów agresywności filtrowania oraz np. usunąć któryś z predefiniowanych serwerów reklamowych, jeśli jego filtrowanie

powoduje błędy na jakiejś odwiedzanej stronie. Zalecanym rozwiązaniem jednak jest postąpienie w odwrotny sposób, tj. dodanie problematycznej domeny do filtra, który de facto *wyłącza* filtrowanie (takie filtry istnieją zdefiniowane w systemie).

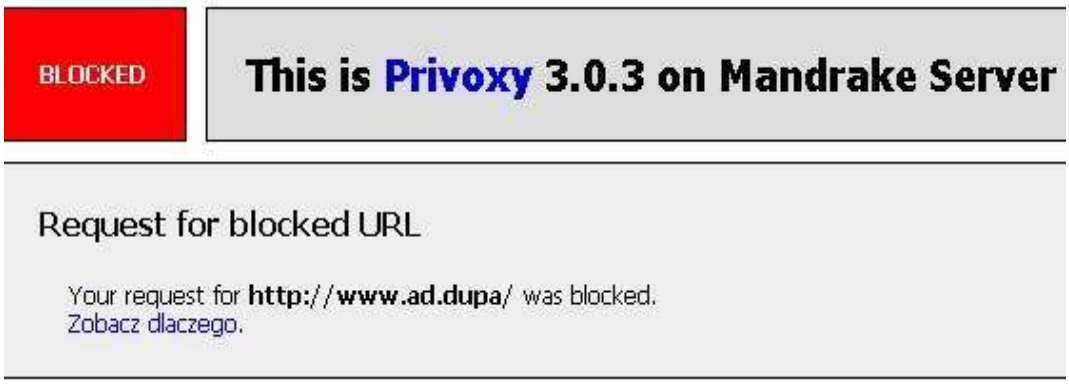
Ciąg wyzwalający działanie filtra jest jednak tylko jednym z dwóch elementów pełnej filtracji. Jej drugą częścią jest zadecydowanie, jaka akcja ma być podjęta przez Privoxy, gdy wyzwalacz zadziała. Możliwych do podjęcia akcji jest kilkadziesiąt. Oto niektóre z nich:

js-annoyances	- usuwa denerwujące skrypty java
refresh-tags	- zapobiega samoprzeładowywaniu się strony
all-popups	- wyłącza wszelkie wyskakujące okna
jumping-windows	- wyłącza przemieszczanie się okien po ekranie
frameset-borders	- włącza ramki umożliwiające skalowanie okien
Deanimate-gifs	- likwiduje animację plików graficznych typu gif.
quicktime-kioskmode	- pozwala na zapisanie filmów quicktime
banners-by-size	- wycina banery reklamowe po rozmiarze
ie-exploits	- nie dopuszcza kodu, wykorzystującego luki bezpieczeństwa w Internet Explorerze (ten filtr wymaga uaktualniania na bieżąco, w miarę jak pojawiają się nowe luki w tym oprogramowaniu)

Pod względem opisywanego wyżej blokowania pobierania plików binarnych, liczyć się jednak będą tylko dwie akcje: "block" oraz "handle as image".

Celem obu akcji jest niedopuszczenie do pobrania przez użytkownika pliku, który pasowałby do wzorca. Różnią się jednak efektem końcowym. Akcja "block" powoduje wyświetlenie na ekranie przeglądarki strony internetowej generowanej przez Privoxy, która informuje o odbytej filtracji i umożliwia, mimo wszystko, pobranie filtrowanego materiału. Jest to rozwiązanie adresowane dla świadomych użytkowników, i dlatego nie nadaje się do zabezpieczania zbiorowego sieci. Zamiast tej akcji, należy użyć więc opcji "handle as image". Różni się ona tym, że zamiast strony wyjaśniającej, Privoxy podrzuca przeglądarce plik graficzny. Domyślnie jest to pusty (przeźroczysty) plik gif, przez co strona wydaje się pusta. Użytkownik może się z niej wycofać naciskając wsteczny klawisz przeglądarki.

Administrator, definiując akcję “set image blocker” może jednak przekierować wywołanie na dowolny URL – decydując tym samym, jaki komunikat zobaczy użytkownik. Standardowy, nie zmodyfikowany ekran blokujący widoczny jest na Rys. 7.



Rysunek 7 Ekran blokujący Privoxy.

Należy też wspomnieć, że chociaż Privoxy jest programem, który de facto ogranicza swobodę surfowania po Internecie, to jednak użytkownicy go lubią. Dzieje się tak dlatego, że strony internetowe pozbawione mrugających, nachalnych reklam przegląda się dużo łatwiej.

3.3 Ochrona poczty elektronicznej

Program MKS_vir jest obecnie produktem firmy MKS, spółki z o. o., która powstała w roku 1996. Korzenie programu sięgają jednak roku 1987, kiedy to był tworzony przez jednego autora, Marka Sella, i już wtedy bardzo popularny w Polsce.

Przez cały czas program jest sukcesywnie rozwijany. Obecnie główną wersją jest wersja zaprojektowana dla systemów Windows, w liniach rozwojowych 95/98/Millenium oraz NT/2000/XP. W celu umożliwienia skanowania dysków twardych bez startowania systemu operacyjnego (który, po infekcji, może być trwale uszkodzony przez wirusa) nadal utrzymywana jest wersja programu MKS_vir przeznaczona dla systemu operacyjnego DOS oraz namiastka tego systemu, czyli mikrosystem operacyjny MKS_DOS, umożliwiający uruchomienie komputera z płyty CD, uzyskanie dostępu do dysków twardych z partycjami typu FAT i przeprowadzenie skanowania DOSową wersją MKS_vir prosto z płyty.

Ostatnim produktem, nad którym pracuje obecnie firma MKS jest jednak program MKS_vir w wersji dla systemów Linux, FreeBSD, NetBSD, OpenBSD oraz Solaris. Systemy te są mało podatne na infekcje typowymi wirusami dostępnymi w sieci. Istnieje ku temu kilka powodów. Po pierwsze, autorzy wirusów starają się tworzyć produkty jak najbardziej skuteczne, rozprzestrzeniające się bez przeszkód, a więc "kompatybilne" z jak największą liczbą komputerów PC, a to oznacza w praktyce infekowanie systemu Windows, który jest najpopularniejszy. Systemy typu Unix są względnie mało popularne, zwłaszcza w zastosowaniach "na biurku", a więc nie są wymarzonym celem ataku. Dodatkowo, instalacja i utrzymywanie takich systemów jest względnie trudna, a więc ich administratorzy należą niewątpliwie do grupy bardziej zaawansowanych użytkowników, którzy zawsze są bardziej świadomi tego, co robią. Kolejnym powodem odporności antywirusowej systemów zbliżonych do Uniksa jest ich architektura. Konta

użytkowników są bardzo ściśle rozdzielone i mają rygorystycznie nadane uprawnienia. Działające procesy, jeśli tylko nie muszą, nie są uruchamiane z prawami administratora (roota) lub, jeśli taka konieczność istnieje (np. w przypadku uzyskania dostępu do określonego portu), natychmiast rezygnują z tych uprawnień, gdy tylko nie są one już potrzebne. Dodatkowo, niektóre programy są napisane tak, że natychmiast rezygnują z pracy, jeśli uruchomiono je omyłkowo z uprawnieniami administracyjnymi (gdy nie jest to konieczne do ich pracy; tak np. zachowuje się wygaszacz ekranu xscreensaver). Inną cechą odpornościową jest fakt, że systemy typu Linux czy BSD posiadają wbudowaną możliwość zaawansowanej filtracji pakietów IP, jaką wykorzystuje wyżej opisany system Linux Freesco, podczas gdy w systemach Windows odpowiednie programy trzeba nabywać i instalować osobno – czego większość użytkowników nie jest nawet świadoma.

Wszystkie te cechy sprawiają, że wobec dziesiątek tysięcy wirusów i ich odmian na systemy Windows, na systemy Uniksowe wirusów istnieje tylko kilka. Można by więc postawić pytanie, po co instalować skaner antywirusowy na komputerze wyposażonym w system, któremu nie grozi infekcja. Otóż system Linux może pracować jako serwer poczty oraz plików. W takim przypadku, przez jego dyski przewijają się dane wysyłane przez użytkowników Windows i do nich adresowane. W ten sposób, system Linux, sam niewrażliwy na ataki wirusów, w sposób bierny przyczyna się do ich rozpowszechniania. W tym celu coraz większe zainteresowanie wzbudza możliwość prowadzenia skanowania antywirusowego na serwerze, niejako w imieniu nieświadomych niczego użytkowników Windows.

W chwili, gdy powstaje niniejsza praca (maj-lipiec 2004), wersja programu MKS_vir dla systemów Uniksowych jest darmowa. Ma ona status wersji rozwojowo-testowej i jest tworzona na wpół oficjalnie przez pracowników firmy MKS. Co więcej: demon zajmujący się skanowaniem plików został napisany

niezależnie od firmy MKS i jest programem o otwartym kodzie źródłowym. MKS_vir w wersji uniksowej jest w pełni sprawny i spełnia swoje zadanie. Biblioteki z sygnaturami wirusów są przez firmę MKS publikowane w postaci paczek tgz na serwerze FTP równolegle z bibliotekami dla wersji windowsowej. Program może być użytkowany zarówno przez osoby prywatne jak i przez firmy, ale z jednym zastrzeżeniem: spod Linuksa nie wolno skanować zasobów sieciowych fizycznie zlokalizowanych na dyskach twardych komputerów pracujących pod kontrolą systemu Windows. Takie podejście firmy wydaje się całkowicie uzasadnione. Lokalne dyski na komputerze z systemami zgodnymi z Uniksem / Linuksem / BSD można jednak kontrolować bez ograniczeń.

3.3.1 Zasada działania

Aby wykorzystać program MKS_vir do skanowania poczty należy:

- posiadać działający serwer pocztowy, np. postfix,
- zainstalować pakiet Amavis,
- zainstalować pakiet MKS_vir ze skanerem pracującym jako demon,
- zintegrować powyższe programy aby ze sobą współpracowały.

Instalacja MKS_vira jako demona jest niezbędna w celu zaoszczędzenia zasobów komputera. Przy każdym starcie programu potrzebny jest czas na załadowanie odpowiednich bibliotek i bazy wirusów – co wymaga zużycia czasu procesora. Powtarzanie takiego procesu dla każdego skanowanego pliku byłoby nieekonomiczne. o wiele prościej jest uruchomić więc skaner antywirusowy tylko raz – w tle – i przesyłać mu pliki do skanowania w momencie, gdy te się pojawią w poczcie.

Pakiet Amavis-new jest przeznaczony do wypakowywania zawartości przesyłki e-mailowej tzn. treści i załączników (które mogą być dodatkowo skompresowane programami typu zip czy rar) i przekazywania ich do przeskanowania antywirusowego. W przypadku, gdy MKS_vir stwierdzi obecność

wirusa w pliku, zwraca kod błędu do programu Amavis, a ten generuje listy elektroniczne z informacją o tym fakcie i przesyła je do nadawcy, odbiorcy lub administratora. W celu sprawdzenia obecności wirusów, przesyłki e-mail mogłyby być w zasadzie wysyłane przez serwer pocztowy SMTP prosto do programu MKS_vir, ale nie potrafi on jeszcze obsługiwać wielu formatów kompresowalnych archiwów i nie generuje ładnych, czytelnych komunikatów, dlatego musi korzystać z programu Amavis. Nie jest to wielkim utrudnieniem, ponieważ Amavis nie zużywa wiele dodatkowych zasobów, gdyż drzemie w systemie jako demon, i aktywuje się wyłącznie w miarę potrzeb.

3.3.2 Instalacja i konfiguracja Postfiksa

Dla systemu Linux Mandrake, Postfix jest dostępny w postaci pakietu RPM, czyli skompilowanych plików binarnych, które pobiera się z Internetu (z serwerów oprogramowania) i instaluje przy pomocy Instalatora oprogramowania (RPM Drake) dostępnego z menu powłoki graficznej lub z linii poleceń jako rpm. Pakiet zainstalowany w modelowej sieci to wersja RPM 2.0.6-1mdk.

Skrypt instalacyjny kopiuje niezbędne pliki z pakietu RPM do odpowiednich lokalizacji na dysku twardym: konfiguracja trafia do katalogu /etc/postfix/, biblioteki do /usr/lib/postfix/, a pliki wykonywalne do /usr/sbin i /usr/bin/. Po instalacji, dokumentację można znaleźć w /usr/share/doc/postfix/. Po zainstalowaniu, demon uruchamia się przy każdym starcie systemu i czeka na połączenia na domyślnym porcie (25).

Postfix, którego autorem jest Wietse Venema, jest bardzo dużym i złożonym serwerem pocztowym. Zaprojektowany został około roku 1998 jako alternatywa dla Sendmaila (w którym odkryto na przestrzeni lat wiele słabości i dziur), a głównymi założeniami projektu były skalowalność, niezawodność i bezpieczeństwo[26]. Cele te osiągnięto rozbijając program na moduły, które ładowane są do pamięci w miarę potrzeb. Zdobycie przez włamywacza jednego modułu nie

26 R. J. Hontanon: *op. cit*. s. 178.

umożliwia przejęcia wszystkich praw, które posiada pracujący Postfix – właśnie przez ową modularność. Większość procesów Postfiksa jest uruchamiana w systemie z niskimi uprawnieniami, mając dostęp jedynie do ograniczonego drzewa plików. Procesy są odizolowane od siebie, więc problemy z jednym z nich nie przenoszą się na inne, a ich właścicielem nie jest użytkownik, który je uruchamia, a specjalny demon, również działający z niskimi uprawnieniami. Jeśli dodać do tego, że Postfix jest odporny na typowy błąd przepełnienia bufora na wejściu – można uznać, że mieści się w czołówce programów pocztowych, uznawanych za bezpieczne[27].

Podstawową konfigurację przeprowadza się następująco. Najpierw należy wyedytować plik /etc/postfix/main.cf i parametrowi myhostname nadać pełną nazwę domenową komputera, np.

myhostname = smtp.biuro.pl oraz taki sam adres ustawić jako źródło wysyłania wiadomości w sieć (posługując się zmienną): myorigin = $mydomain oraz jako adres docelowy dla danej domeny: mydestination = $myshosname, $mydomain.

Ze względów bezpieczeństwa należy też przekierować wiadomości z konta użytkownika root na któreś konto uprzywilejowane. Jest to niezbędne, ponieważ Postfix pracuje jako program nieuprzywilejowany i, jako taki, nie byłby w stanie uzyskać dostępu do katalogów z pocztą należących do użytkownika root[28].

Jeśli serwer pracuje prawidłowo, można przystąpić do przekierowania poczty przychodzącej najpierw do programu Amavis, który będzie wywoływał skanowanie antywirusowe. Przedtem jednak trzeba zainstalować i uruchomić i przetestować oba programy.

27 Ibidem.
28 M. D. Bauer: *op. cit.* s. 261 – 262.

3.3.3 Instalacja i konfiguracja MKS_vir

Pakiety programu MKS oraz opis jego instalacji są dostępne na stronie http://linux.mks.com.pl. Z tej samej lokalizacji można również pobierać aktualne bazy z sygnaturami wirusów. Programowi nie towarzyszy żaden skrypt instalacyjny: dostępne są jedynie archiwa .tgz, (mks32-Linux-i386-1-9-6.tgz, mksdLinux-1.15.2.tgz) zawierające binarne pliki wykonywalne oraz dokumentację. Instalację należy prowadzić zgodnie z zawartym w niej opisem.

W pierwszym rzędzie, należy wydobyć z archiwum plik mks32 i umieścić go w katalogu programów dostępnych dla użytkowników, np. /usr/bin i nadać mu odpowiednie prawa. Następnie należy utworzyć katalog, w którym przechowywane będą bazy antywirusowe, np. /usr/mks/bazy i zapisać w nim pobrane ze strony firmy MKS pliki z sygnaturami wirusa (mksbase*.dat). Kolejnym krokiem jest utworzenie pliku konfiguracyjnego dla programu, tj. /etc/mks_vir.cfg i wstawienie w nim wpisu, który będzie informował program, gdzie bazy sygnatur się znajdują, tj. linię

--mks-vir-dat-path=/usr/mks/bazy/ .

W tym momencie można już przeprowadzić pierwszą próbę działania programu antywirusowego. Należy wywołać program mks32, jako argument podając nazwę podejrzanego pliku do przeskanowania. Dobrym pomysłem jest przeskanowanie specjalnego pliku testowego, wywołującego fałszywy alarm. Plik taki można pobrać ze strony www.eicar.com – jest to zwykły plik tekstowy, zawierający krótki ciąg drukowalnych znaków, konkretnie jest to

```
X.5.O.!.P.%.@.A.P.[.4.\.P.Z.X.54(P^)7CC)7}$EICAR-
        STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

na który oprogramowanie antywirusowe, na skutek umowy między producentami, reaguje komunikatem o wykryciu wirusa. Plik ten jest stosowany diagnostyce. Skanowanie pliku testowego powinno zakończyć się następującym komunikatem:

```
mks_vir: init...  1.9.6 for Linux i386, 2004.03.09
mks_vir: database version 2004 5 14  15 35
```

```
mks_vir: init OK, scan mode
mks_vir: check file(s)
mks_vir: file: /home/andrzej/eicar
mks_vir:        --heuristic for virus Eicar.Test
mks_vir: status: virus found: /home/andrzej/eicar
mks_vir: exit code: 0x01
```

Jak widać, program na tym etapie działa i wykrywa wirusy ("virus found").

Kolejny etap to instalacja demona mksd. W jego skład wchodzą dwa pliki wykonywalne, mksd oraz mksscan, które należy umieścić w katalogu /user/sbin i nadać im odpowiednie prawa. Ponieważ demonom prawa roota nie są niezbędne do pracy, najlepiej jest utworzyć specjalnego użytkownika (np. antywir) i uruchamiać programy z jego konta. Należy też stworzyć katalog /var/run/mksd, jego właścicielem uczynić użytkownika antywir i upewnić się, że ma on do tego katalogu pełne prawa. Pozostali użytkownicy powinni mieć prawa do wejścia do tego katalogu oraz do czytania zawartych w nim plików.

Teraz można przetestować skanowanie plików w trybie demona. W tym celu, jako użytkownik antywir należy uruchomić demona mksd. Następnie, jako dowolny użytkownik należy ponownie wykonać test, tym razem jednak wywołując program mksscan. Komunikat o infekcji powinien wyglądać tak:

```
VIR Eicar.Test /home/andrzej/eicar
```

natomiast komunikat o zdrowym pliku:

```
OK /home/andrzej/praca_magisterska.doc
```

3.3.4 Instalacja i konfiguracja pakietu Amavis

Dla systemu Linux Mandrake, Amavis jest dostępny w postaci pakietu RPM, czyli skompilowanych plików binarnych, które pobiera się z Internetu (z serwerów oprogramowania) i instaluje przy pomocy Instalatora oprogramowania (RPM Drake) dostępnego z menu powłoki graficznej lub z linii poleceń jako rpm. Pakiet zainstalowany w modelowej sieci to wersja RPM amavis-ng-0.1.3-2mdk.

Skrypt instalacyjny kopiuje niezbędne pliki z pakietu RPM do odpowiednich lokalizacji na dysku twardym: plik wykonywalny do /usr/bin/amavis a plik konfiguracyjny do /etc/amavis/amavis.conf. Bardzo wyczerpująca dokumentacja do programu jest instalowana do /usr/share/doc/amavis-ng-0.1.3/. Plik README zawiera opis programu i zasadę jego działania, a oprócz niego istnieją pliki opisujące współpracę z każdym rodzajem serwera poczty (w tym przypadku jest do plik README.postfix).

Dodatkowo, do katalogu /usr/lib/perl5/vendor_perl/5.8.0/AMAVIS/AV kopiowane są pliki z definicjami poleceń współpracy programu Amavis z rozmaitymi skanerami antywirusowymi. Skrypt obsługujący współpracę z programem MKS_vir nazywa się MKS.pm.

Po zainstalowaniu, program musi być skonfigurowany do współpracy zarówno z MKS_virem, jak i z serwerem poczty. Należy wyedytować plik /etc/amavis/amavis.conf, odkomentowując linie odpowiedzialne za wybór MTA i skanera: Amavis posiada bowiem predefiniowane konfiguracje współpracy z wieloma serwerami SMTP (m. in. Sendmail, Postfix, Exim) oraz skanerami antywirusowymi (m. in. MKS, Sophos, Panda, FPROT, Trend czy ClamAV, program antywirusowy OpenSource, również dostępny za darmo). Odkomentowanie, czyli aktywacja wybranej linii, odbywa się przez usunięcie znaku " ; " z jej początku:

```
; Which MTA to use. Specify one.
; mail-transfer-agent = DebugMTA
; mail-transfer-agent = Exim
; mail-transfer-agent = EximPerl
mail-transfer-agent = Postfix
; mail-transfer-agent = Sendmail
; mail-transfer-agent = SMTP
;; Which virus scanner to use. Use more than one if you desire
; virus-scanner = FSAV
; virus-scanner = AVP
; virus-scanner = FSP
; virus-scanner = hbEDV
```

```
; virus-scanner = Sophos
virus-scanner = MKS
; virus-scanner = Sophie
; virus-scanner = Bitdefender
; virus-scanner = FPROT
```

Dodatkowo trzeba skonfigurować obsługę popularnych formatów kompresji i pakowania plików – główny powód zastosowania Amavisa. System Linux przeważnie obsługuje ZIP, GZIP, TAR czy BZIP2. Z Internetu można pobrać darmowe dekompresory, np. unRAR, który jest dość popularny. Formaty mniej znane, np. ZOO, ARC albo LZH można albo doinstalować, albo wyłączyć w konfiguracji. Poczta, zawierająca załączniki w nie obsługiwanych formatach będzie wtedy blokowana. Podczas ładowania, program Amavis sprawdza, czy odpowiednie dekompresory są dostępne.

W dalszej części pliku konfiguracyjnego amavis.conf należy podać ścieżkę dostępu do MKS_vira.

```
[MKS]
mks = /usr/bin/mks32
```

Kolejnym krokiem jest włączenie przekierowania poczty do Amavisa z programu Postfix, w celu spowodowania jej przeskanowania programem antywirusowym. Dokonuje się tego edytując plik /etc/postfix/master.cf, który steruje głównym procesem programu pocztowego. Włączenie przekierowania składa się z dwóch kroków: zdefiniowania filtra oraz zadeklarowania, że usługa SMTP ma z niego korzystać.

Filtr definiuje się, wpisując na końcu pliku master.cf dwie linie, przytoczone w przykładowych plikach konfiguracyjnych Amavisa:

```
filter unix - n n - - pipe flags=Rq user=mail
argv=/usr/bin/amavis ${sender} -- ${recipient}
```

Aby wymusić stosowanie filtra, należy nakazać usłudze SMTP korzystnie z niego. W tym celu należy wykomentować linię:

```
smtp inet      n – y - - smtp
```

i zastąpić ją linią:

```
smtp inet      n - - - - smtpd -o content_filter=filter:
```

która powoduje przekierowanie ruchu do filtra zdefiniowanego wyżej, czyli przesyłanie e-maili do skanera antywirusowego i odbieranie przez niego komunikatów. Schematycznie wygląda to następująco:

Tak zestawiony układ należy teraz przetestować. W tym celu należy wysłać e-mail do domeny, obsługiwanej przez własny serwer Postfix i obejrzeć log Amavisa:

```
May 15 21:56:08 localhost amavis[5457]: Unpacking message in /
tmp/amavis-unpack-40a675d8-1551
May 15 21:56:08 localhost amavis[5457]: File 00000000 is type
message/rfc822
May 15 21:56:08 localhost amavis[5457]: File 00000001 is type text/plain
May 15 21:56:08 localhost amavis[5457]: Scanning for viruses with
AMAVIS::AV::MKS
May 15 21:56:08 localhost amavis[5457]: AMAVIS::MTA::Postfix: Message
clean – forwarding
```

Komunikat "Message clean - forwarding" oznacza, że przesyłka jest czysta i że będzie przekazana dalej. a oto, jak zachowa się Amavis w przypadku otrzymania od Postfiksa przesyłki z wirusem:

```
May 16 15:50:23 localhost amavis[26780]: File 00000002 is type
text/plain
May 16 15:50:23 localhost amavis[26780]: File 00000001 is type
text/plain
May 16 15:50:23 localhost amavis[26780]: Scanning for viruses with
AMAVIS::AV::MKS
May 16 15:50:23 localhost amavis[26780]: AMAVIS::MTA::Postfix: Discarded
- VIRUS: Eicar.Test
```

```
May 16 15:50:23 localhost amavis[26780]: AMAVIS: Cleaning up.
May 16 15:50:23 localhost amavis[26780]: AMAVIS: Removing /tmp/amavis-
unpack-40a7719e-689c
```

Wiadomość została zlikwidowana (Discarded). Odbiorca, nadawca oraz administrator systemu otrzymają powiadomienie e-mailem, przekazane z Amavisa za pośrednictwem serwera pocztowego Postfix:

```
This is the Postfix program at host andrzej.gryfitki.wmc.com.pl.
I'm sorry to have to inform you that the message returned
below could not be delivered to one or more destinations.
host localhost[10.0.12.20] said: 550
VIRUS Eicar-Test-Signature FOUND
```

(“Z przykrością informuję, że przytoczona poniżej wiadomość nie mogła być dostarczona do miejsca przeznaczenia. Lokalny serwer mówi: znaleziono wirusa Eicar-Test.”)

Od tej pory ochrona antywirusowa działa. Administrator musi jednak regularnie uaktualniać bazy wirusów. Firma MKS publikuje nowe bazy mniej więcej raz na dobę; można je pobierać z serwera ftp.mks.com.pl. Do tego celu służy skrypt, który można znaleźć w pliku instalacyjnym pakietu MKS.

3.4 Zabezpieczenia serwera plików

System Linux na obecnym etapie rozwoju nie sprawdza się zbyt dobrze jako system na biurko, którego można by używać w pracy typowo administracyjnej. Dla każdej firmy istnieje cała gama programów, które pracują wyłącznie pod kontrolą systemu Windows, a użytkownik prędzej czy później będzie zmuszony z nich skorzystać, a więc – nie będzie mógł zrobić tego wyłącznie w systemie Linux. Do takich programów należą np. ZUSowski program "Płatnik", program "PIT-y 2003", umożliwiający rozliczanie się z Urzędem Skarbowym, czy wreszcie typowe programy do prowadzenia działalności gospodarczej jak "Subiekt", "Fakturowanie" itp. W praktyce oznacza to, że użytkownik na stacjach roboczych będzie miał zainstalowany system MS Windows.

System Windows uznawany jest powszechnie za system podatny na infekcje wirusowe i inne naruszenia bezpieczeństwa[29]. Jest po temu wiele przyczyn. Jedną z nich niewątpliwie jest podejście firmy Microsoft, która w latach dziewięćdziesiątych nie doceniała wagi problemu i stawiała na łatwość obsługi systemu oraz intensywną reklamę, zamiast na bezpieczeństwo. Bardzo trafnie ujął to Tomasz Polus: "Microsoft poprzez swój marketing wyrobił u użytkowników, a także (niestety) administratorów błędne przekonanie, że system jest tak przyjazny dla użytkownika, że wszystko robi się samo, a interwencja administratora potrzebna jest tylko w przypadku awarii (...)"[30]. Komercyjne podejście do swojej działalności nakazuje firmie Microsoft wypuszczać nową wersję systemu co 2 – 3 lata. Efektem takiego postępowania było dopuszczanie do obrotu wersji systemu, które w sumie nie były dopracowane z powodu wyścigu z czasem, który starali się wygrać programiści. Naprawianie błędów odraczało się dopiero do wydania kolejnych

29 Przykładowo, w raporcie CERT-Polska (tj. *Zespołu d/s reagowania na przypadki naruszenia bezpieczeństwa teleinformatycznego* przy NASK) za IV kwartał roku 2003 opisano aż 15 luk w bezpieczeństwie, wykrytych w tym okresie. Dotyczyły one bezpośrednio systemu Windows albo oprogramowania stanowiącego jego integralną część: Internet Explorera, Outlook Expressa, Exchange (program Exchange jest serwerem pocztowym zintegrowanym np. z MS Windows 2000 SBS Server). Kilka spośród tych luk umożliwiło właśnie tak intensywne rozprzestrzenianie się robaków *Swen, Sasser czy Blaster* na początku roku 2004.

30 T. Polus: Łamanie haseł w NT, IT-FAQ, http://www.it-faq.pl, 2001.

ServicePacków, czyli zbiorczych pakietów z "łatami", instalujących zabezpieczenia wykrytych błędów. Np. dla systemu Windows NT ukazało się ich aż sześć. Karmiąc użytkownika intensywną reklamą i wizjami bezpiecznego i stabilnego systemu, firma Microsoft w istocie jednak niechętnie przyznawała się do istnienia w systemie dziur wykrytych przez osoby trzecie, często pozostawiając takie sygnały bez komentarza i nie stosując natychmiast jakichkolwiek kroków zaradczych. Wszelkie błędy "jedynie słusznego" systemu nie były po prostu zgodne z linią marketingową produktu, jaki stanowił system Windows.

Innym czynnikiem, który na pewno jest nie bez znaczenia przy rozpatrywaniu bezpieczeństwa Windows jest po prostu popularność tego systemu. Obecność okienek na praktycznie każdym PC na świecie sprawia, że jest to system, który świetnie znają zarówno potencjalni włamywacze, jak i autorzy wirusów. Oznacza to, że mogą być oni niezmiernie skuteczni w swojej przestępczej działalności.

Wszystkie te przyczyny mają jeden skutek: obecnie na system Windows istnieją dziesiątki tysięcy wirusów, które mogą mu szkodzić, np. uszkadzając pliki systemowe. Zadaniem administratora jest przeciwdziałanie takim sytuacjom.

Jednym z założeń ochrony powinna być centralizacja składowania plików. Ochrona każdej z osobna stacji roboczej wymagałaby zwiększonych sił i środków oraz byłaby powodem wzrostu kosztów operacyjnych.

Dlatego też najłatwiej jest składować wszystkie istotne dla działalności firmy pliki i programy na jednej maszynie, i na niej stosować ochronę antywirusową. Takie podejście stwarza dwupłaszczyznowy problem: system Linux musi być w stanie udostępniać pliki stacjom roboczym z systemem

Windows, posługując się zrozumiałym dla nich protokołem a do tego powinien mieć wdrożoną ochronę antywirusową.

Rozwiązanie obu problemów jest możliwe przy użyciu serwera Samba oraz omówionego wyżej skanera antywirusowego MKS_vir. Ideą systemu jest skanowanie antywirusowe plików w czasie ich zapisu na serwer oraz skanowanie plików już złożonych na serwerze w regularnych odstępach czasu, gdy serwer jest najmniej zajęty – np. w nocy. Skanowanie periodyczne plików znajdujących się na serwerze jest niezbędne, bowiem może się zdarzyć, że w momencie skanowania przy zapisie pliku, wirus może pozostać nie wykryty. Sytuacja taka jest możliwa, ponieważ systemy antywirusowe opierają swoje działanie na tzw. sygnaturach wirusów. Oznacza to, że firma antywirusowa bada dostarczone lub przechwycone w sieci okazy podejrzanych plików i, w przypadku stwierdzenia ich szkodliwego charakteru, stara się wybrać z nich mały odcinek, który jest dla nich na tyle charakterystyczny, że może posłużyć do ich identyfikacji. Odcinek ten, będący swego rodzaju odciskiem papilarnym wirusa, nazywany jest sygnaturą. Zestaw aktualnych sygnatur jest przygotowywany przez firmy antywirusowe np. raz na dobę i publikowany na serwerach. Oprogramowanie antywirusowe pobiera te sygnatury, zwane też "bazami wirusów", i od tej pory jest w stanie rozpoznać zdefiniowane w nich wirusy. Zalecane jest uaktualnianie baz wirusów przynajmniej raz w tygodniu, lub częściej, np. raz na dobę, jeśli jest to możliwe. Niestety – nadal jednak prawdopodobna jest sytuacja, w której użytkownik pobierze wirusa z Internetu, lub otrzyma go pocztą w chwili, gdy jest on jeszcze nierozpoznawalny przez bieżące bazy. W ten sposób wirus ma możliwość przeniknięcia na serwer plików. W takim przypadku dopiero skanowanie antywirusowe następnej nocy, tj. po pobraniu nowych sygnatur, ma szansę wirusa ujawnić. Widać więc, że regularne sprawdzanie plików na serwerze jest niezbędne.

3.4.1 Zasada działania

Aby wykorzystać MKS_vir do skanowania plików na serwerze należy:

- posiadać działający serwer plików Samba,
- zainstalować pakiet MKS_vir ze skanerem pracującym jako demon,
- zainstalować pakiet samba-vscan-mks,
- zintegrować powyższe pakiety do wspólnej pracy,
- zainstalować demona systemowego Cron (większość systemów linuksowych instaluje go w standardzie) w celu skanowania plików w określonych odstępach czasowych.

Instalacja MKS_vir została opisana w poprzednim rozdziale. Demon skanujący pliki jest tym samym demonem, który skanuje pocztę.

Pakiet samba-vscan-mks jest systemem, który spowoduje, że plik, który użytkownik sieci stara się zapisać na serwerze zostanie wysłany najpierw do programu antywirusowego. Po przeskanowaniu z wynikiem negatywnym plik zostanie zapisany. Jeśli jednak w pliku zostanie wykryty wirus, plik będzie zamiast tego przeniesiony do specjalnego katalogu kwarantanny, gdzie, niedostępny dla użytkownika, pozostanie w dyspozycji administratora systemu. Użytkownik natomiast dostanie ostrzeżenie z informacją o dokonanej operacji i jej przyczynie.

3.4.2 Instalacja i konfiguracja serwera Samba

Dla systemu Linux Mandrake, Samba jest dostępna w postaci pakietu RPM, zawierającego już skompilowane pliki binarne. Pakiet można pobrać z instalacyjnej płyty CD systemu, albo z jednego z wielu serwerów oprogramowania (np. ftp.samba.org). Instalacji dokonuje się przy pomocy `Instalatora oprogramowania (RPM Drake)`. Skrypt instalacyjny umieszcza główne pliki Samby (demony smbd i nmbd) w katalogu sbin oraz dodatkowe pliki niezbędne do pracy serwera w katalogach /usr/bin, /usr/lib, usr/share/samba i /var/lib. Głównym plikiem konfiguracyjnym jest /etc/samba/smb.conf. Konfigurację przeprowadza się

edytując jego zawartość, a następnie wydając polecenie `samba restart`, powodując odczytanie zawartości pliku konfiguracyjnego i uruchomienie się usługi z nowymi parametrami pracy.

Plik smb.conf podzielony jest na sekcje. Każda sekcja składa się z parametrów, z których każdy ma wpływ na sposób pracy serwera. Trzy sekcje mają charakter specjalny. Są to sekcje *global, homes* oraz *printers*. W strukturze pliku, sekcje są oznaczane nagłówkami w nawiasach kwadratowych, np. [global]. Pozostale wiersze mają postać *nazwa_parametru=wartość*. Wartości stanowią ustawienia pracy, przypisane do parametru. Mogą nimi być np. *yes, no, true, false,* ale również ścieżki itp. Dodatkowo, plik smb.conf może zawierać wiesze, będące komentarzem. W takim przypadku pierwszym znakiem wiersza jest #.

Sekcja [global] zawiera parametry sterujące ogólnym zachowaniem się Samby. Obejmuje to m. in. ustawienia bezpieczeństwa oraz nazw sieciowych NetBIOS, stosowanych przez system Windows w "Otoczeniu sieciowym".

Sekcja [homes] określa, że każdemu użytkownikowi serwera Samba ma być udostępniony jego katalog domowy.

Sekcja [printers] umożliwia udostępnianie użytkownikom drukarek, podłączonych do serwera Samby.

Pozostałe sekcje są tworzone przez administratora i określają katalogi, które dostępne będą dla użytkowników w sieci. Nazwa sekcji odpowiada nazwie zasobu, jaki użytkownik Windowsa zobaczy w "Otoczeniu sieciowym". Jako przykład można podać [Dokumenty_wspolne], [Muzyka] czy [MS_Office].

Prosty (a zarazem działający) plik smb.conf może wyglądać np. tak[31]:

```
[global]
workgroup = Biuro
guest account = pcguest
```

31 Za: C. Hunt: Serwery sieciowe Linuksa, Wyd. Mikom, Warszawa 2000, s. 405.

```
log file = /var/log/samba/log.%m
Server String = Serwer plików
[homes]
comment = Katalogi macierzyste
browseable = no
writable = yes
[Dokumenty_wspolne]
comment = Wszystko razem
path = /home/samba/dokumenty_wspolne
browseable = yes
writable = no
```

Dokładne znaczenie parametrów jest następujące:

`Workgroup` oznacza tzw. grupę roboczą, stosowaną przez komputery systemu Windows. Komputer X widzi w "Otoczeniu sieciowym" inne komputery, których grupa robocza jest taka sama jak komputera X. W tym przypadku, wszystkie komputery powinny być skonfigurowane do pracy w grupie roboczej o nazwie "Biuro".

`Server string` stanowi opis, który pojawi się w opisie widocznego w sieci serwera.

`Guest account` określa nazwę konta gościa, jeśli jest ono potrzebne w systemie.

`Log file` określa położenie i nazwę pliku dziennika, w którym zapisywane będą błędy, umożliwiając ich analizę w miarę potrzeb. Użycie parametru %m umożliwia skonstruowanie nazwy dziennika ze słowa log, kropki, oraz nazwy komputera, który podłączał się do serwera (wartość %m). Oznacza to, że w praktyce każdy podłączający się do serwera komputer będzie miał własny plik z dziennikiem błędów, np. "log.maciek".

`Comment` zawiera komentarz do udostępnionego zasobu.

`Browseable` określa, czy udział będzie widoczny w momencie przeglądania sieci. Jeśli wartość jest ustalona na `no`, zasób będzie niewidoczny

w momencie przeglądania i użytkownik będzie mógł do niego wejść jedynie wtedy, gdy będzie znał ścieżkę dostępu do niego (i dysponował np. skrótem umieszczonym na pulpicie).

`Writable` oznacza, czy do udostępnionego zasobu można zapisywać pliki. Domyślnie, wszystkie zasoby są skonfigurowane jedynie do odczytu. Aby umożliwić zapis do nich, administrator musi wartość tego parametru na `yes`.

Serwer samba skonfigurowany w powyższy sposób udostępnia użytkownikom ich katalogi domowe oraz zasób wspólny o nazwie Dokumenty_wspólne, ale nie zapewnia skanowania antywirusowego. Aby się ono odbywało, konieczne jest dodanie kolejnych wierszy konfiguracyjnych. Na początek w sekcji [global] należy umieścić:

```
vfs object = vscan-mksd
vscan-mksd: config-file = /etc/samba/vscan-mks32.conf
```

Dodatkowo, każdy chroniony zasób należy zaopatrzyć w wiersz, mówiący serwerowi Samba, że przy dostępie do udostępnianego pliku, powinien skorzystać z modułu, współpracującego ze skanerem antywirusowym:

```
[Dokumenty_wspolne]
vfs object = /usr/lib/samba/vfs/vscan-mks.so
comment = Wszystko razem
path = /home/samba/dokumenty_wspolne
browseable = yes
writable = no
```

Dopisane linie stanowią o wyposażeniu serwera Samba we współpracę z modułem samba-vscan-mks, który z kolei integruje ją ze skanerem MKS_vir. W sekcji [global] zdefiniowane są: typ modułu antywirusowego (vscan-mksd) i ścieżka do pliku konfiguracyjnego (/etc/samba/vscan-mks32.conf). W sekcji odpowiedzialnej za zasób [Dokumenty_wspolne] podana jest z kolei ścieżka do modułu integracji ze skanerem antywirusowym (vscan-mks.so). Za przesłanie pliku do skanera antywirusowego oraz decyzję, co zrobić w przypadku wykrycia wirusa, odpowiedzialny jest pakiet samba-vscan-mks.

3.4.3 Instalacja i konfiguracja pakietu samba-vscan-mks.

Dla systemu Linux Mandrake, pakiet samba-vscan-mks jest dostępny w postaci pakietu RPM, zawierającego już skompilowane pliki binarne. Pakiet można pobrać z z jednego z wielu serwerów oprogramowania (np. ftp.mandrake.org). Wersja wykorzystana na potrzeby niniejszej pracy nosi numer 2.2.7a-3 mdk. Instalacji dokonuje się przy pomocy `Instalatora oprogramowania (RPM Drake)`. Skrypt instalacyjny umieszcza pliki w katalogu /usr/lib. Miejsce usytuowania pliku konfiguracyjnego trzeba wskazać w pliku konfiguracyjnym Samby, w linii `vscan-mksd: config-file`. Domyślną lokalizacją jest /etc/samba/vscan-mks32.conf.

Zawartość tego pliku decyduje o parametrach pracy modułu. Poprzez jego edycję, administrator ma możliwość określenia sposobu, w jaki skanowane będą pliki. Przykładowo, można zadecydować, jaki będzie maksymalny rozmiar skanowanego pliku, czy moduł powinien blokować dostęp, gdy skanowanie nie będzie możliwe (np. gdy demon mksd nie jest aktywny), czy użytkownik powinien dostać na ekran komunikat z ostrzeżeniem oraz co moduł ma zrobić z zainfekowanym plikiem (przenieść do kwarantanny, skasować czy nic nie robić). Typowa zawartość pliku konfiguracyjnego przedstawia się następująco:

```
[samba-vscan]
max file size = 0
```

```
scan on open = yes
scan on close = yes
deny access on error = yes
deny access on minor error = yes
send warning message = yes
infected file action = quarantine
quarantine directory = /tmp/kwarantanna
quarantine prefix = vir-
```

W podanym powyżej przykładzie pliki będą skanowane niezależnie od ich rozmiaru (wartość zero oznacza brak limitu), a samo skanowanie będzie się odbywało przy otwieraniu oraz przy zamykaniu pliku. Gdy wystąpią błędy skanowania dostęp do plików będzie zablokowany, a w przypadku wykrycia wirusa użytkownik zostanie poinformowany komunikatem poprzez usługę *Windows Messanger*. Zainfekowane pliki będą składowane w katalogu kwarantanny, umieszczonym w /tmp. Jednocześnie ich nazwa będzie poprzedzana przedrostkiem vir.

3.4.4 Skanowanie periodyczne systemu plików.

Oprócz skanowania plików przy otwieraniu ich przez użytkowników z sieci, niezbędne jest skanowanie co jakiś czas, aby umożliwić wykrywanie wirusów, które mogły być "przegapione". Proces skanowania najlepiej jest prowadzić w chwilach, kiedy serwer jest najmniej obciążony: np. W nocy albo w weekendy. Do tego celu najlepiej nadaje się program Cron.

Cron jest demonem (czyli programem pracującym w tle), który jest obecny chyba w każdej dystrybucji systemu Linux. Służy on do automatycznego uruchamiania zadań o określonych porach. W jego skład wchodzi program crond, pracujący w tle przez cały czas, plik crontab, zawierający harmonogram wykonywania zadań oraz polecenie crontab, które służy do edycji tego pliku.

Składnia pliku crontab może wydawać się w pierwszej chwili dość skomplikowana, ale pakietowi towarzyszy dokumentacja z dokładnym opisem oraz przykładami.

Oprócz tego, istnieje wiele nakładek graficznych, które umożliwiają wygenerowanie pliku crontab poprzez klikanie myszą.

Ogólny wzór wiersza odpowiedzialnego za wykonanie polecenia o określonym czasie wygląda następująco:

```
min godz data miesi?c dzie? u?ytkownik /katalog/polecenie
```

Obecność znaku gwiazdki w dowolnym poleceniu oznacza "każdy". Aby uzyskać skanowanie co noc, np. o godzinie 2:05, w imieniu użytkownika "administrator", należy do crontaba wpisać wiersz:

```
05 2 * * * administrator /usr/antywirus/skanowanie
```

W tym przypadku, polecenie "skanowanie" jest prostym skryptem, który wywołuje program antywirusowy. MKS_vir w wersji dostępnej dla systemu Linux potrafi jedynie skanować pliki, których nazwy otrzymuje na standardowym wejściu w momencie wywołania programu. Dlatego też najprostszym sposobem jest uruchomić program `find`, który będzie szukał zadanych plików w określonej lokalizacji (np. w katalogu i jego podkatalogach) a następnie, poprzez potok, przekazywał ich nazwy na wejście programu mks32, powodując ich sekwencyjne skanowanie. Skrypt "skanowanie" może posiadać przykładową zawartość:

```
#!/bin/bash
find /home/rodzinka/Dokumenty -type f -print | mks32 -s -S
```

W tym przykładzie, program `find` będzie szukał obiektów o typie plik (-type f, od "file"), a wynik wyszukiwania pokazywał (-print) do potoku (`|`) na którego drugim końcu znajduje się program `mks32`, dokonujący właściwego skanowania. Program `mks32` jest wywoływany z dwoma parametrami. Jeden z nich (-S) nakazuje mu pobieranie nazw pliku ze standardowego wejścia (tu: z potoku), natomiast drugi decyduje, co powinno stać się z plikiem, w przypadku stwierdzenia infekcji. Dostępne opcje to:

-s (scan)	skanuj, co w praktyce oznacza "nic nie rób"; w tym przypadku użytkownik zostanie poinformowany e-mailem o wynikach skanowania,
-c (clean)	wyczyść, co powoduje usunięcie wirusa z pliku
-r (rename)	zmień nazwę / przenieś, co powoduje zmianę nazwy pliku, poprzez dodanie przyrostka .vir.

Kolejnym poleceniem po zakończeniu skanowania można przenosić pliki np. do katalogu kwarantanny, do dyspozycji administratora, który może je przeglądać, usuwać, lub starać się uratować z nich ocalałą treść.

3.5 Stacje robocze

W zabezpieczaniu sieci lokalnej bardzo istotne jest zapobieganie problemom, których źródłem mogą być użytkownicy i ich stacje robocze. W praktyce, im jest ich więcej, tym trudniej jest je kontrolować, a potencjalnie każda z nich może stać się "wrotami infekcji".

Użytkownik może wprowadzić do systemu program szkodliwy zasadniczo na dwa sposoby: albo za pośrednictwem poczty elektronicznej lub stron WWW – zapisując i uruchamiając otrzymany z zewnątrz załącznik, albo korzystając z urządzenia wejścia, np. CD-ROMu lub stacji dyskietek.

Pierwsza z tych możliwości powinna być wyeliminowana poprzez antywirusowe skanowanie poczty oraz filtrację stron internetowych. Oba te sposoby zostały omówione w rozdziałach 3.2 i 3.3. Niestety, skanowanie nie zawsze musi być skuteczne, a poza tym nie uwzględnia pewnego ważnego aspektu: są nim programy komputerowe, które nie stanowią zagrożenia sensu stricto, ale są np. nielegalne lub nie związane z wykonywaną przez użytkownika pracą. Instalacja takiego oprogramowania, oprócz sprawiania problemów prawnych, zajmuje niepotrzebnie przestrzeń na dyskach, obciąża system, a ponadto może narazić firmę na konsekwencje prawne.

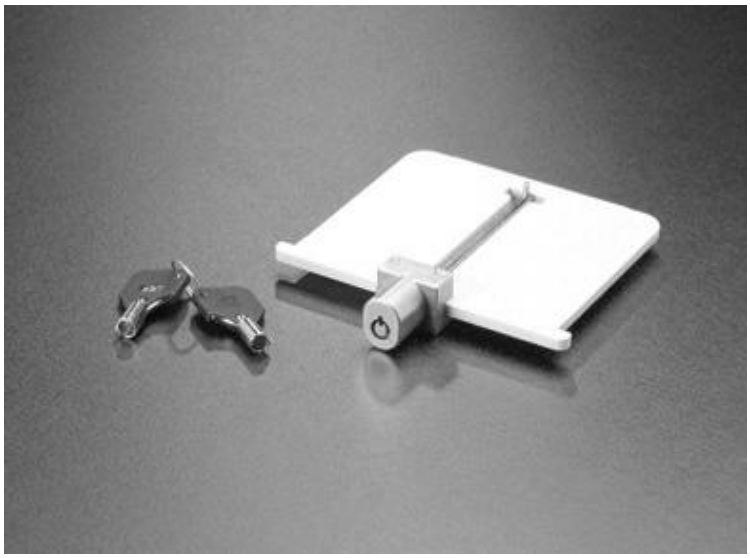
Zgodnie z wspomnianą wielokrotnie zasadą minimum koniecznego[32], należy więc nadawać użytkownikom jedynie te prawa, których potrzebują do wykonywania swoich obowiązków. W praktyce oznacza to, że należy zablokować również drugi sposób wprowadzenia programów do systemu, czyli możliwość korzystania z nośników wymiennych (o ile nie są potrzebne) oraz możliwość instalacji oprogramowania.

3.5.1 Blokady sprzętowe

Istnieją zasadniczo trzy sposoby na odłączenie wybranych urządzeń wejścia. Najbardziej radykalnym z nich, ale też i prymitywnym, jest niewątpliwie

32 J. Stokłosa T. Bilski, T. Pankowski: *op. cit.* s. 21.

odłączenie przewodów wewnątrz obudowy komputera. Jeśli do tego obudowa jest zabezpieczona fizycznie przed otwarciem (np. przy pomocy zamka lub miejsca na kłódkę – jak to można zauważyć na wielu firmowych obudowach, np. IBM), zabezpieczenie takie na pewno jest skuteczne. Niestety, posiada ono zasadniczą wadę: w toku administrowania stacjami roboczymi prędzej czy później pojawi się potrzeba podłączenia stacji dyskietek czy CD-ROMu. Niezbędne jest wtedy wyłączanie i otwarcie komputera oraz fizyczna manipulacja w obudowie. Dużo wygodniejsze są więc dostępne w handlu wkładki z kluczykiem, którymi blokuje się albo całe obudowy, nierzadko przy okazji chroniąc je przed kradzieżą, albo tylko napęd dyskietek, Rys. 8.

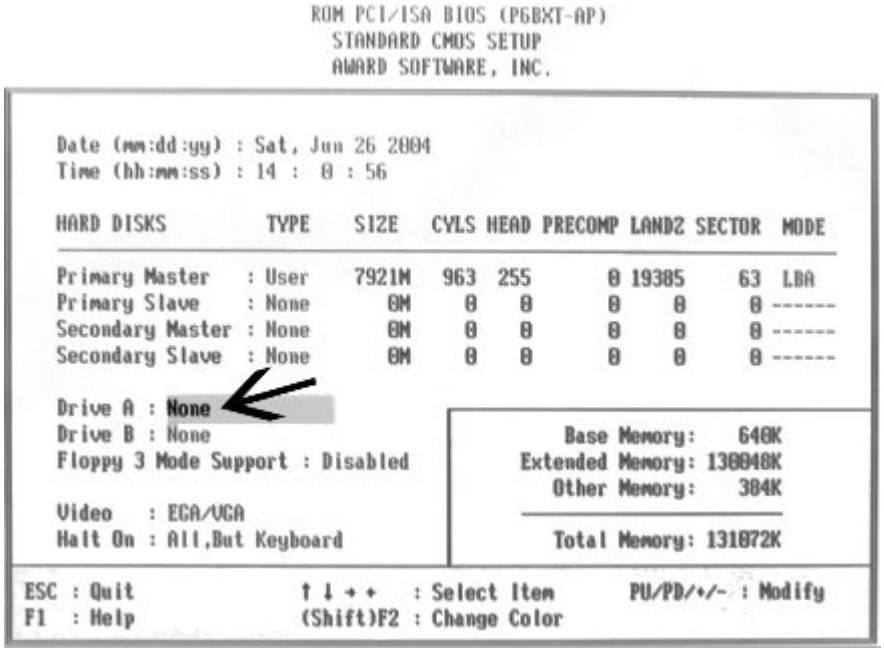


Rysunek 8 Blokada stacji dyskietek.

Innym wyjściem, które wydaje się bardziej elastyczne, jest rozwiązanie programowe. Są to albo restrykcje nałożone w BIOSie komputera, albo ograniczenia wykorzystujące możliwości systemu operacyjnego.

Blokada urządzeń wejścia w BIOSie jest bardzo prosta do przeprowadzenia. Sposób jej zakładania może różnić się w zależności od typu płyty głównej komputera. Zasadniczo jednak, wystarczy wejść w menu konfiguracji (przeważnie dokonuje się tego wciskając klawisz Del w momencie, gdy komputer

uruchamia się) i ustawić pole wyboru odpowiedzialne za rodzaj urządzenia (floppy, CDROM) na wartość NONE, Rys. 9.



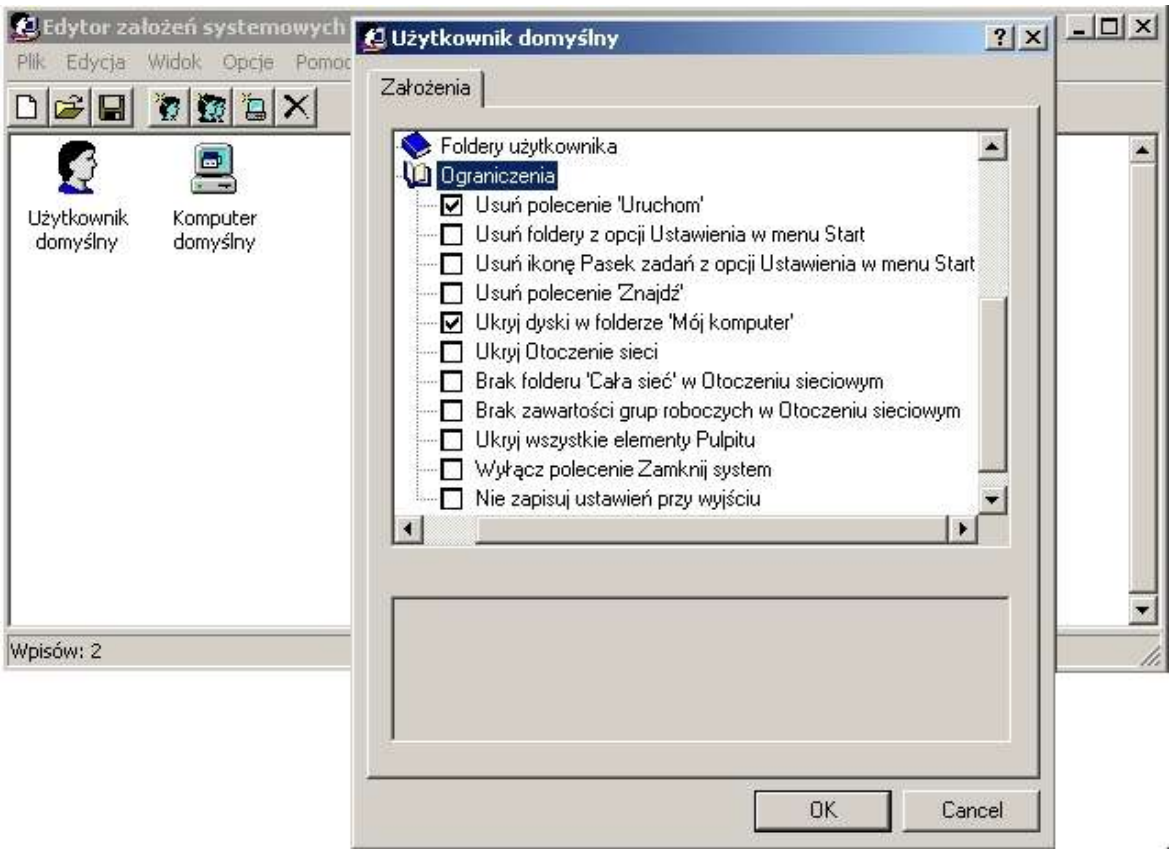
Rysunek 9 Wyłączanie napędu dyskietek w BIOSie.

Następnie na BIOS należy nałożyć hasło administracyjne. Zamknięcie fizyczne obudowy jest niezbędne, ponieważ bardziej zaawansowany użytkownik mógłby skasować zawartość BIOSu, np. poprzez wyjęcie na jakiś czas baterii zasilającej. Jednak pomimo tego ograniczenia, wyłączanie zbędnych urządzeń wejścia przy pomocy BIOSu wydaje się dość skuteczne.

3.5.2 Blokady programowe

Pozostałe rozwiązania programowe blokady urządzeń wejściowych zależą od użytego systemu operacyjnego. Najmniejsze możliwości mają systemy Windows 95/98 i Millennium. Jako nakładki na system DOS, nie posiadają zaawansowanego systemu uprawnień do obiektów plikowych i urządzeń. Microsoft zaproponował administratorom narzędzie o nazwie "Edytor założeń systemowych" (*Policy editor,* czyli de facto: edytor "polityki bezpieczeństwa"), ale jest to narzędzie o ograniczonych możliwościach. Jego wygląd ilustruje Rys. 10. Poledit umożliwia między innymi usuwanie wybranych funkcji systemu Windows, np. likwidację

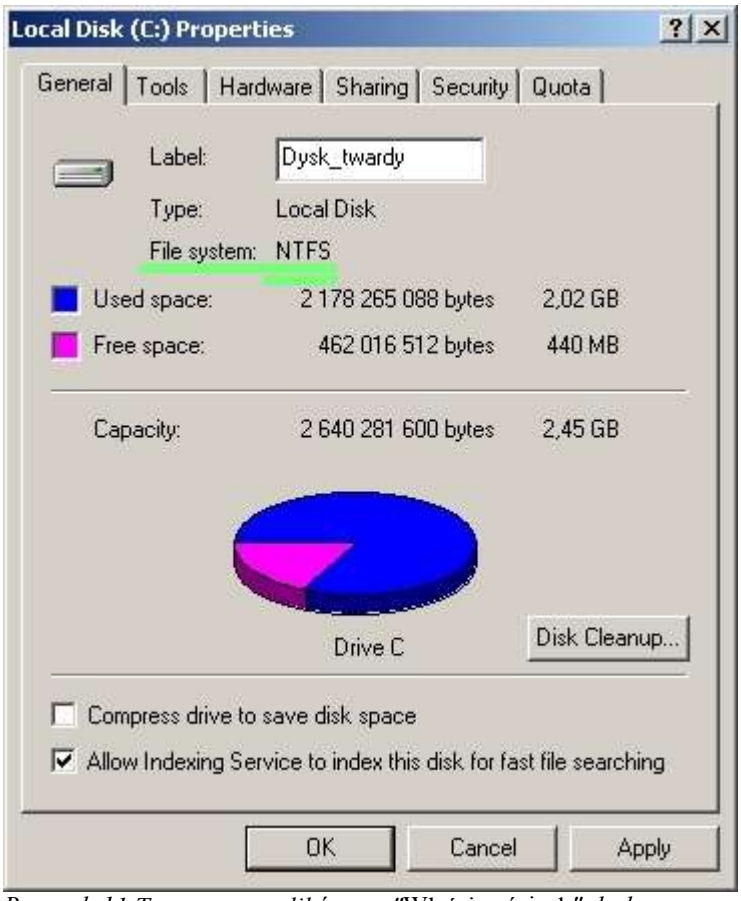
polecenia „Uruchom" z menu „Start" czy schowanie dysków w oknie *Mój komputer* – ale, niestety, nie są to zabezpieczenia, które można uznać za szczelne. Przykładowo, schowane dyski będą nadal widoczne w oknach „Otwórz plik"



Rysunek 10 Okno programu Poledit.

rozmaitych programów, np. w oknie MS Worda *Otwórz dokument*, pozwalając użytkownikowi na manipulację nimi z menu kontekstowego. Inną możliwością, którą daje ten program jest wskazanie wybranych plików wykonywalnych, jako jedynych, które można uruchamiać w systemie. Blokada ta blokada działa wyłącznie po nazwie, więc nazwanie dowolnego pliku tak jak plik dozwolony, umożliwia jego uruchomienie. Nie byłoby tak, gdyby ograniczenia te ustawiane były nie tylko według nazwy, ale również np. według wyliczonej z niego wartości funkcji skrótu (np. MD5), czyli tak, jak robi to wiele aplikacji typu firewall, wykrywając w ten sposób wszelkie podmiany plików, posiadających określone uprawnienia. Niestety, takiego mechanizmu w Windows 95 i 98 zabrakło.

System Windows NT i jego późniejsze wcielenia (2000, XP i 2003 Serwer) dysponują o wiele bardziej zaawansowanymi narzędziami służącymi do kontroli bezpieczeństwa i nakładania restrykcji. Przy założeniu, że system zainstalowany jest na partycji z systemem plików NTFS (która umożliwia przypisanie uprawnień do plików i folderów) można wyłączać z użytku dla konkretnych użytkowników wybrane pliki wykonywalne oraz gałęzie rejestru.



*Rysunek 11 Typ systemu plików we "*Właściwościach*" dysku.*

Ponadto, w celu zwiększenia bezpieczeństwa systemu, użytkownika można przypisać do jednej z wielu grup, np. *Gość (Guest)* które nie posiadają praw do instalowania nowego oprogramowania czy wprowadzania zmian do rejestru systemowego. Niestety – rozwiązania te nadal nie posiadają opcji blokowania dostępu do urządzeń, np. do wchodzących w życie dysków USB (w Windows 2000 i nowszych), przez co nadal trudno jest nazwać je doskonałymi. W tej sytuacji

blokada urządzeń w BIOSie albo zastosowanie zamknięć fizycznych wydają się jednak bezkonkurencyjne[33].

Aby móc w pełni wykorzystywać ograniczenia, nałożone na użytkowników dzięki systemowi uprawnień, Windows powinien być zainstalowany na partycji NTFS, która umożliwia ich użycie. Jeśli nie dokonano takiego wyboru już na etapie instalacji, można dokonać konwersji później, w dogodnym momencie. Dokonuje się tego wydając polecenie `convert c: /fs:ntfs`. Po dokonaniu ponownego uruchomienia, w systemie będzie można ustawić odpowiednie ograniczenia. Rodzaj systemu plików na dysku można ustalić, oglądając jego właściwości, np. w oknie *Mój komputer*, Rys. 11, podczas gdy ekran polecenia `convert` widoczny jest na Rys. 12.

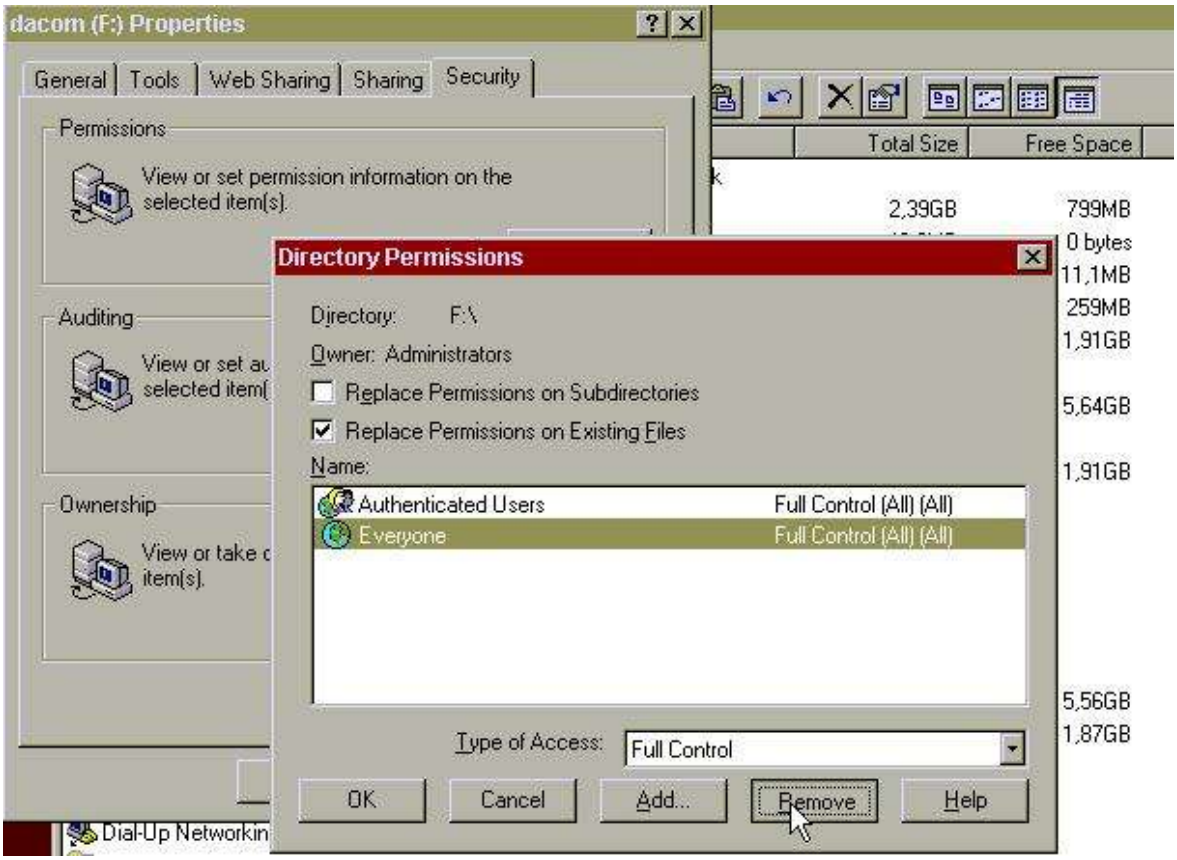
```
C:\>convert /?
Converts FAT volumes to NTFS.

CONVERT drive: /FS:NTFS [/V]

  drive       Specifies the drive to convert to NTFS.  Note that
              you cannot convert the current drive.
  /FS:NTFS    Specifies to convert the volume to NTFS.
  /V          Specifies that Convert should be run in verbose mode.

C:\>_
```

Rysunek 12 Polecenie konwersji plików do systemu NTFS.

Mając już system plików NTFS należy zmienić domyślne uprawnienia do systemu plików, ponieważ domyślnie jest on nadawany wszystkim (tj. grupie *Everyone*). Jest to dość znaczne naruszenie bezpieczeństwa, ponieważ grupa ta obejmuje również użytkowników anonimowych, a więc nie zalogowanych w systemie. Grupę *Everyone* należy usunąć, na jej miejsce podstawiając grupę *Authenticated users*, czyli użytkowników, którzy przeszli przez weryfikację tożsamości i uzyskali autoryzację w systemie. Sposób prawidłowego nadawania praw do dysku widoczny jest na Rys. 13.

33 Warto jednak wspomnieć, że problem blokady urządzeń występuje w Windows. Systemy Linux, Unix czy BSD są pod tym względem o wiele bardziej zaawansowane.

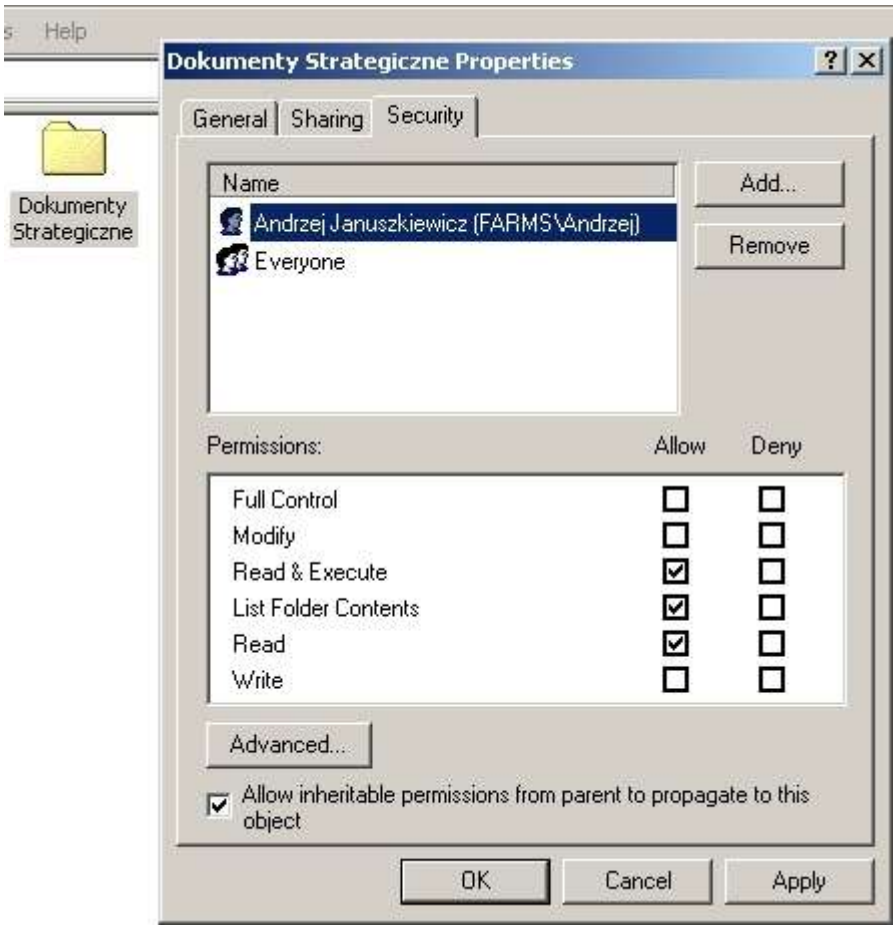
Stosując edytor praw użytkowników należy też zadecydować o możliwości wykonywania przez nich dodatkowych czynności, np. zmiany parametrów pracy systemu operacyjnego. W bezpiecznym systemie lepiej jest nie zezwalać użytkownikom na edycję rejestru systemowego czy dostęp do plików w katalogach systemowych Windows. Zadanie tych ograniczeń uniemożliwi też instalację oprogramowania. Uprawnienia użytkowników ustawia się w panelu zarządzania komputerem, dostępnym w Windows NT w menu Start/Programy/Narzędzia Administracyjne" a w XP i 2000 pod prawym klawiszem myszki na ikonie "Mój komputer", w menu "Zarządzaj/Użytkownicy i Grupy".



Rysunek 13 Nadawanie praw do przykładowego dysku F.

Warto też wspomnieć, że począwszy od Windows XP, pełne ustawianie uprawnień użytkowników możliwe jest jedynie w wersjach XP Professional. Wersje Home Edition mają jedynie możliwość uczynienia użytkownika administratorem (wszystkie możliwe prawa) lub zwykłym użytkownikiem (brak

możliwości edycji rejestru i instalowania oprogramowania). Zalecane jest pracowanie na co dzień na koncie nieuprzywilejowanym[34]. Istotne jest też nadanie folderom z ważnymi dokumentami jedynie uprawnień, wynikającym z zasady minimum koniecznego, jak to pokazano na Rys. 14.

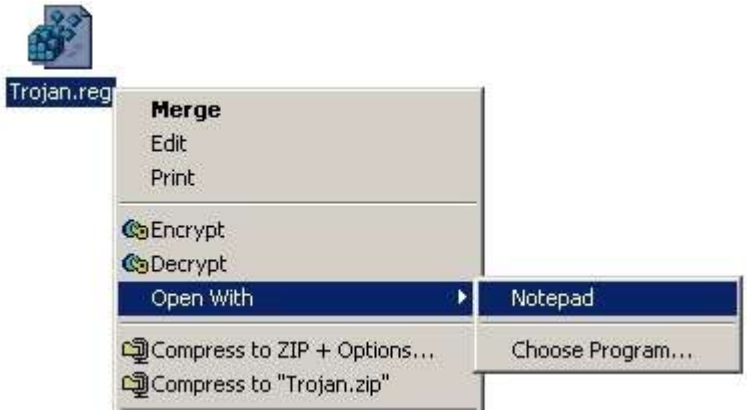


Rysunek 14 Usuwanie zbędnych praw do foldera z danymi.

Ostatnim elementem uszczelniania stacji roboczej będzie zmiana skojarzenia typu plików *.reg z rejestrem. W ten sposób zminimalizuje się możliwość przypadkowego wstawienia zawartości takiego pliku do rejestru. Oczywiście najlepiej jest, jeśli użytkownik w ogóle nie będzie miał praw do zmiany wpisów w rejestrze, co uzyskuje się korzystając ze wspomnianych wyżej uprawnień w systemie, ale zmiana skojarzenia pliku może mimo wszystko pomóc uniknąć

34 Niestety, na skutek niedociągnięć Microsoftu, praca na koncie nieuprzywilejowanym, chociaż zalecana ze względów bezpieczeństwa, bardzo często sprawia problemy z uruchamianiem programów do obsługi niektórych urządzeń – np. skanerów lub nagrywarek płyt CD. Chcąc-niechcąc użytkownicy korzystają więc w codziennej pracy z kont administracyjnych.

pomyłki, a przy tym łatwo ją przeprowadzić. Wystarczy stworzyć dowolny plik z rozszerzeniem .reg, kliknąć na nim prawym klawiszem myszy trzymając wciśnięty klawisz Shift, wybrać z menu *Otwórz z i* wskazać jakikolwiek neutralny program, np. Notatnik, zaznaczając jednocześnie opcję "*Zawsze używaj tego programu do otwierania plików takiego typu*". Po wykonaniu tej operacji, kliknięcie na pliku .reg otworzy jego zawartość w notatniku, co jest całkowicie nieszkodliwe, zamiast automatycznie wczytać go do rejestru i spowodować być może, w zależności od tego, co plik zawiera, zmianę parametrów pracy systemu. Operacja pokazana jest na Rys. 15.

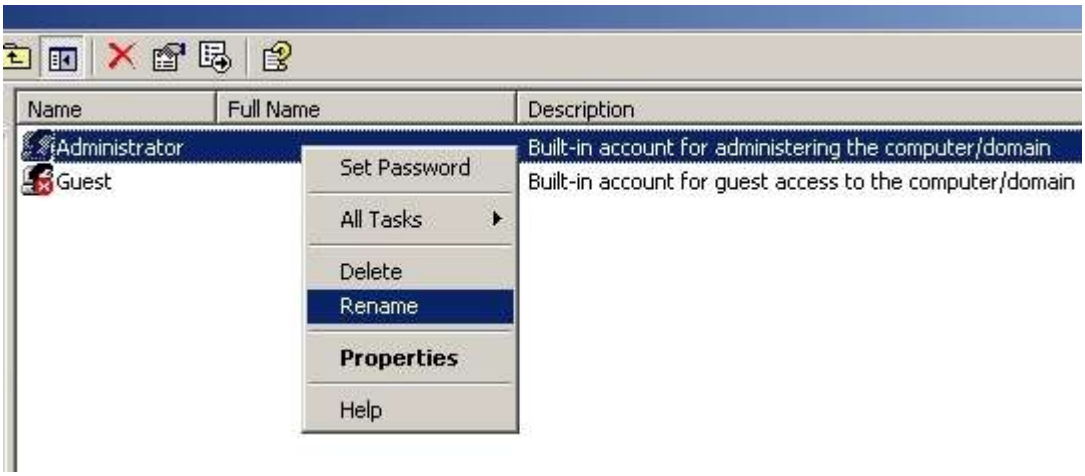


Rysunek 15 Zmiana skojarzenia plików .reg.

3.5.3 Blokady od strony sieci

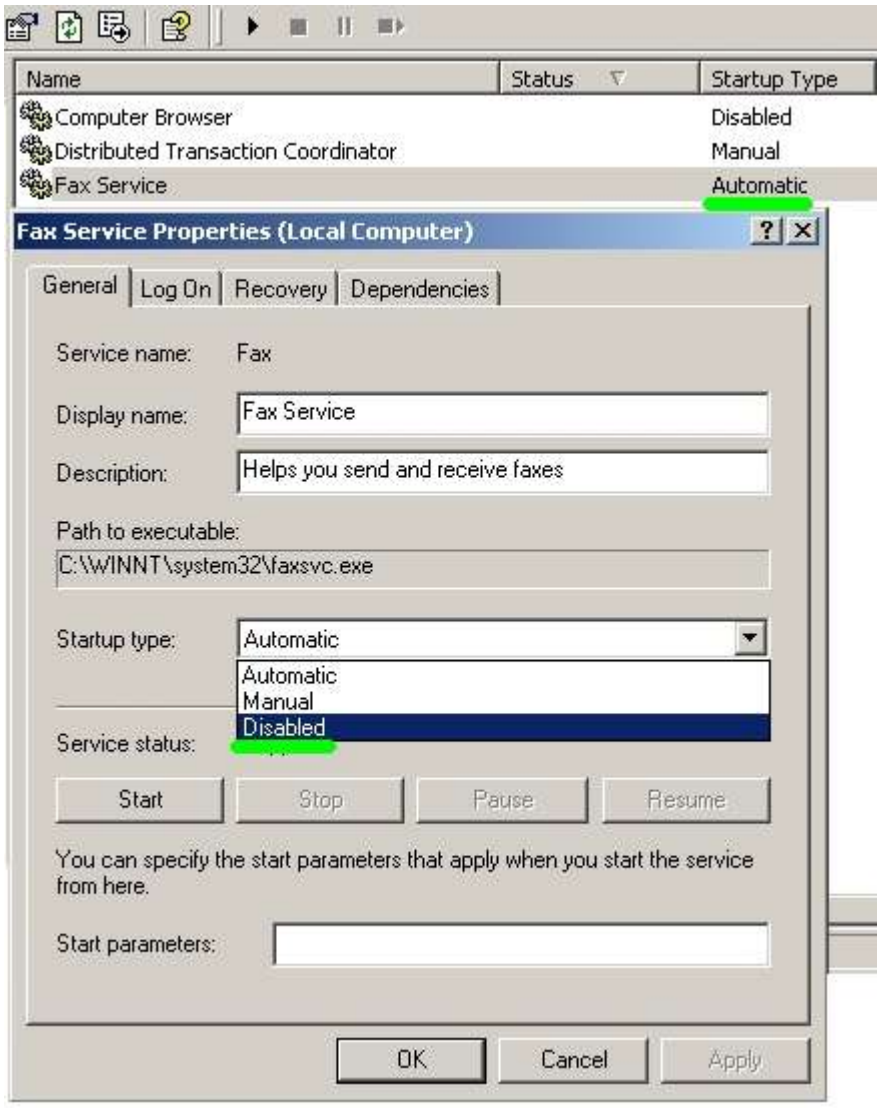
Kolejnymi etapem zabezpieczania stacji roboczej jest ochrona jej przed dostępem od strony sieci komputerowej. Składa się na nią wiele czynników. Literatura zaleca przede wszystkim zmianę nazwy konta *Administrator*[35] na taką, która z administracją się zupełnie nie kojarzy. Krok ten przysporzy potencjalnemu włamywaczowi dodatkowych trudności: chcąc włamać się na konto administratora będzie musiał zgadnąć nie tylko pasujące do tego konta hasło, ale także jego nową, niestandardową nazwę. Oprócz tego zaleca się nawet dodatkowe utrudnienie: nadanie fałszywej nazwy "Administrator" specjalnie do tego celu założonemu kontu, które nie posiada żadnych uprawnień w systemie, i zaopatrzenie go w bardzo długie i trudne do odgadnięcia hasło. Konto takie ma na celu przyciągnięcie uwagi potencjalnego włamywacza i zmarnowanie jego czasu, poświęconego na odgadnięcie hasła, które na pewno mu się na nic potem nie przyda. Nie wszyscy autorzy uznają jednak celowość stosowania takiego wybiegu i traktują go wręcz jak "dodatkowe zabezpieczanie zamkniętej kasy pancernej poprzez oklejanie drzwi taśmą klejącą"[36]. Zmiana nazwy konta „Administrator" widoczna jest na Rys. 16.

35 Artykuł "Hardening Win2000", grupa dyskusyjna comp.os.ms-windows.nt.admin.security, 2001
36 Ibidem.



Rysunek 16 Zmiana nazwy konta "Administrator".

Po zabezpieczeniu urządzeń wejścia i systemu plików, należy powyłączać w systemie wszystkie usługi, ponieważ mogą być one wykorzystane do przejęcia kontroli nad komputerem (takie zagrożenie stwarzają np. usługi telnet, albo zdalne zarządzanie rejestrem systemowym).



Rysunek 17 Wyłączanie zbędnych usług w systemie.

Z tego powodu, obowiązująca tu reguła jest identyczna, jak wielokrotnie tu już wspomniana: niezbędne minimum. Oznacza to, że aktywne powinny zostać wyłącznie te usługi, z których się korzysta. Pozostałe należy deaktywować, przy okazji zwalniając pamięć RAM, co pozwoli szybciej pracować programom użytkowym. Ekran wyłączania usług widoczny jest na Rys. 17. Przykładowe usługi, które na ogół nie są w systemie potrzebne, a które Windows 2000 po instalacji zawsze uruchamia to:

Computer browser	*Przeglądarka komputerów w sieci; zbędna, jeśli używa się systemu Samba, posiadającego opcję serwera WINS*
Clipbook	*Narzędzie umożliwiające przesyłanie zawartości schowka między dwiema maszynami*
Fax Service	*Serwis do wysyłania faksów*
Indexing Service	*System ułatwiający poszukiwanie dokumentów; działa lokalnie, więc jest bezużyteczny, jeśli swoje dokumenty użytkownicy przechowują na serwerze plików*
Internet Connection sharing	*Usługa umożliwiająca udostępnianie połączenia internetowego*
Remote Registry Service	*Zdalne zarządzanie rejestrem danego komputera*
Task Scheduler	*Menedżer zadań; o ile użytkownicy z niego nie korzystają*
Telnet	*Usługa umożliwiająca zdalne logowanie się do maszyny; powszechnie uznawana za niebezpieczną, ponieważ posługuje się jawnie przesyłanymi hasłami.*
SNMP	*Usługa do automatycznego zarządzania urządzeniami (potrzebna w dużych sieciach).*

Warto wspomnieć też, że do Windows XP Microsoft załączył oprogramowanie nazwane "zaporą połączenia internetowego". Jest to prosty firewall, który, po włączeniu, uniemożliwia dostęp do komputera z zewnątrz.

Na pewno warto jest go uaktywniać w systemie, aczkolwiek nie należy liczyć na zbyt wiele. Badania tego firewalla, przeprowadzone niezależnie przez magazyn IT-FAQ wykazały bowiem[37], że w pewnych warunkach może on być całkowicie nieskuteczny. Odpowiedzialna za to jest pewna specyficzna, "innowacyjna" cecha użytkowa, w który firma Microsoft wyposażyła ten produkt w celu ułatwienia sobie pracy. Firewall ów może bowiem być wybiórczo wyłączany, na żądanie specyficznych programów. Ma to pomagać w komunikatywności systemu

37 T. Polus: Dobry firewall czy bubel wszechczasów?, IT-FAQ, http://www.it-faq.pl, 2003.

i umożliwiać pracę komunikatorom, np. głosowym, które nie radzą sobie w połączeniach zza firewalla. Chwilowo korzystają z tego mechanizmu programy Microsoftu, np. program Windows Messanger, który otwiera sobie potrzebne porty TCP/UDP. Niestety, praktyka wykazuje, że kwestia pojawienia się wirusów, które będą umiały wykorzystać ten sam mechanizm wyłączania firewalla jest wyłącznie kwestią czasu[38].

Mówiąc o ochronie sieci od strony stacji roboczych należy też wspomnieć o odpowiednim zabezpieczeniu pozostałego sprzętu sieciowego. Zbędne gniazda sieciowe powinny być odłączone od koncentratorów czy przełączników sieciowych. Dostęp fizyczny do urządzeń sieciowych powinien być ograniczony przez stosowanie zamknięć, Rys. 18. Jest to niezbędne, jeśli chce się uniemożliwić podłączanie obcego sprzętu. Zrozumiałe jest również, że serwery sieciowe powinny znajdować się w pomieszczeniach, do których dostęp mogą mieć wyłącznie osoby upoważnione.

Rysunek 18 Zamknięta szafka z przełącznikiem i koncentratorem.

38 Pod koniec lat osiemdziesiątych, firma Microsoft wprowadziła do systemu DOS 6.0/Windows 3.1 wbudowaną ochronę antywirusową swojego autorstwa (MSAV). Kilka miesięcy później praktycznie każdy nowy wirus potrafił ją unieszkodliwiać.

Rozdział 4. Analiza SWOT

Większość oprogramowania użytkowanego na komputerach PC pod kontrolą systemu Windows ma postać zamkniętą. Oznacza to, że firmy komercyjne rozpowszechniają wyłącznie finalną, binarną postać swoich produktów. Posiadając tylko ją, nie sposób jest zrozumieć, na jakiej zasadzie program działa i co dokładnie robi. Nie można też wprowadzić zmian, jeśli jest taka potrzeba. Ponadto, producent nie musi w zasadzie zapewniać efektywności działania, ponieważ nikt nie widzi jego pracy. Na szczęście nie jest to jedyny wybór.

W roku 1984 programista Richard Stallman napotkał na trudności z naprawą urządzenia komputerowego w swoim biurze: nie mógł tego zrobić sam, ponieważ nie miał dostępu do kodu źródłowego programu, a jednocześnie nie mógł wyegzekwować naprawy od producenta, ponieważ ten nie był tym zainteresowany. Frustracja Stallmana okazała się w rezultacie bardzo pożyteczna: w roku 1985 założył on Fundację Otwartego Oprogramowania, która działa do dzisiaj i bardzo dynamicznie się rozwija. W ramach tej fundacji oraz związanego z nią projektu Open Source powstaje dzisiaj wiele programów, tworzonych i publikowanych następnie wraz z kodem źródłowym przez programistów – hobbystów rozsianych po całym świecie. Programy te są szeroko dostępne w Internecie, bezpłatne, a przy tym mają opinię bardzo dobrych. Wynika to z wielu przyczyn: dostępność kodu źródłowego pozwala młodym programistom uczyć się przez studiowanie gotowych wzorców, a prawo do swobodnej manipulacji kodem umożliwia wykorzystywanie rozwiązań opracowanych już w przeszłości, zamiast bezproduktywnego powielania pracy, która już została przez kogoś wykonana. Szeroko dostępny kod umożliwia przy tym bardzo łatwe wykrycie ewentualnych niedociągnięć oraz szybkie ich naprawienie.

Całość oprogramowania służącego do zabezpieczenia sieci komputerowej, omówionego w niniejszej pracy, jest dostępna nieodpłatnie i powstaje

w większości[39] w ramach projektu Open Source. Pojawia się pytanie, czy rzeczywiście taki model rozwiązania jest lepszy od zastosowania produktów komercyjnych. Problem ten jest dość złożony z uwagi na to, że w jego skład wchodzi wiele czynników. Aby je uporządkować, można skorzystać z analizy SWOT, która pozwala skonkretyzować atuty, słabości, szanse oraz zagrożenia związane z korzystaniem z takiego oprogramowania. Analiza ma postać tabel z punktacją, opatrzonych komentarzem.

39 Jedynym wyjątkiem, chociaż nie do końca, jest program MKS_vir. Kod źródłowy programu nie jest dostępny (stanowi własność firmy MKS) ale inicjatywa przeniesienia na platformę linuksową oprogramowania dostępnego pierwotnie jedynie pod Windows oraz udostępniania go i wspierania *bez opłat licencyjnych* jest niewątpliwie godna pochwały.

4.1 Czynniki SWOT

Czynniki wewnętrzne

Atuty	*Punkty*	*Waga*	*Wynik*
Zerowe koszty nabycia	*5*	*0,20*	*1,00*
Szerokie wsparcie w Internecie	*5*	*0,15*	*0,75*
Szybka reakcja na wykryte błędy	*5*	*0,20*	*1,00*
Możliwość załatania błędu samodzielnie	*4*	*0,15*	*0,60*
Znajomość kodu	*3*	*0,15*	*0,45*
Uwolnienie od monopolu	*3*	*0,15*	*0,45*
Suma		*1,00*	*4,25*

Słabości	*Punkty*	*Waga*	*Wynik*
Brak reklamy	*-4*	*0,30*	*-1,20*
Wymagania wobec administratorów	*-3*	*0,40*	*-1,20*
Brak jednego właściciela	*-1*	*0,30*	*-0,30*
			0,00
			0,00
			0,00
Suma		*1,00*	*-2,70*
Atuty i Słabości razem			***1,55***

Tabela 1 Analiza SWOT dla oprogramowania Open Source. Czynniki wewnętrzne. Źródło: opracowanie własne na podstawie: P. Suwiński: Otwarte Oprogramowanie w biznesie, Politechnika Gdańska, http://www.flug.org.pl/.

Punkty określają siłę oddziaływania danego czynnika. Zakres punktacji wynosi od -5 do -1 dla negatywnych i od 1 do 5 dla pozytywnych czynników. Waga określa stopień istotności czynnika w danym obszarze analizy. Sumaryczny wynik może zawierać się w przedziale od -5 do 5.

Czynniki zewnętrzne

Szanse	*Punkty*	*Waga*	*Wynik*
Otwarte standardy	*5*	*0,25*	*1,25*
Niezależność technologiczna	*5*	*0,25*	*1,25*
Wzrost świadomości społecznej	*5*	*0,15*	*0,75*
Nastroje antymonopolowe	*4*	*0,15*	*0,60*
Rozwój Internetu	*3*	*0,10*	*0,30*
Niskie bariery wejścia na rynek	*3*	*0,10*	*0,30*
		1,00	*4,45*

Zagrożenia	*Punkty*	*Waga*	*Wynik*
Patenty na oprogramowanie	*-5*	*0,20*	*-1,00*
Agresywna polityka Microsoftu	*-5*	*0,25*	*-1,25*
Szkodliwa polityka RP	*-4*	*0,15*	*-0,60*
Nieodpowiedni system edukacji	*-4*	*0,20*	*-0,80*
Zamykanie standardów przez monopolistów	*-3*	*0,20*	*-0,60*
		1,00	*-4,25*
Szanse i Zagrożenia razem			***0,20***

Tabela 2 Analiza SWOT dla oprogramowania Open Source. Czynniki zewnętrzne. Źródło: opracowanie własne na podstawie: P. Suwiński: Otwarte Oprogramowanie w biznesie, Politechnika Gdańska, http://www.flug.org.pl/.

Punkty określają siłę oddziaływania danego czynnika. Zakres punktacji wynosi od -5 do -1 dla negatywnych i od 1 do 5 dla pozytywnych czynników. Waga określa stopień istotności czynnika w danym obszarze analizy. Sumaryczny wynik może zawierać się w przedziale od -5 do 5.

4.1.1 Atuty

Zerowe koszty nabycia

Jest to niewątpliwie atut, zwłaszcza dla osób fizycznych i instytucji budżetowych. Oparty na Linuksie w pełni funkcjonalny serwer plików, drukarek oraz poczty elektronicznej (tudzież innych, nie omawianych w niniejszej pracy usług jak np. serwer www) nie kosztuje zupełnie nic, o ile jest pobierany z Internetu. Inną drogą nabycia pełnego pakietu Linux jest zakup gazety informatycznej (np. Linux+) z płytami CD, zawierającymi pełną instalację. Koszt gazety wynosi często mniej niż 30 zł, co w zestawieniu z kwotą ok. 450 zł za podstawową instalację Windows, nie zawierającą w zasadzie żadnych aplikacji użytkowych, wypada bardzo korzystnie. Dodatkowym atutem jest dodatkowe oprogramowanie, również dostępne za darmo, i to nie tylko serwerowe, np. darmowy pakiet OpenOffice, zastępujący MS Office, kosztujący około 1500 zł. Co jest istotne, licencja na oprogramowanie Open Source pozostanie zawsze darmowa, podczas gdy polityka licencyjna Microsoftu bezustannie się zmienia, a poza tym stwarza problemy w momencie wymiany sprzętu, wymagając np. ponownego przeprowadzania rejestracji oprogramowania, co łączy się z dodatkowymi kosztami, np. na rozmowy telefoniczne z siedzibą Microsoft Polska.

Szerokie wsparcie w Internecie

Oprogramowanie Open Source nie jest tworzone anonimowo. Bardzo łatwo jest wejść w kontakt z autorami programu, lub nawet poszczególnych jego funkcji. Autorzy standardowo dołączają swoje adresy elektroniczne do pakietów dystrybucyjnych i chętnie czekają na sugestie. Ale nie są oni jedynymi osobami, do których można zwracać się z prośbą o poradę: ponieważ kod programu jest otwarty, wielu innych programistów jest w stanie wytropić przyczyny ewentualnych problemów i pomóc w ich rozwiązaniu. Istnieje też grupa hobbystów, którzy, nie będąc programistami, a tylko zaawansowanymi użytkownikami, specjalizuje się

w pisaniu dokumentacji do programów. Dokumentacja taka ma dwojaką postać: albo systematyczny opis zadań, jakie wykonuje dany program i wszystkich jego funkcji, albo opis wdrażania pewnych usług systemu przy pomocy różnych programów. Ten drugi system, zorientowany bardzo praktycznie, nosi nazwę HowTo (w polskim tłumaczeniu: Jak To Zrobić) i nie ma właściwie swojego odpowiednika w świecie Windows. Całą tę dokumentację bardzo łatwo jest znaleźć w Internecie, bez polegania na jednym autorytatywnym źródle (producencie), jak byłoby w przypadku oprogramowania zamkniętego – np. opartego na Windows.

Szybka reakcja na wykryte błędy

Oprogramowanie Open Source tworzone jest w ramach hobby, a więc dla uzyskania moralnej satysfakcji autorów. Sprawia to, że programy są na znakomitym poziomie jakościowym a ich doskonalenie lub naprawianie odbywa się na bieżąco. Reakcja na wykryty błąd, zgłoszona autorowi lub publicznie, na forum internetowym, często pojawia się w czasie krótszym niż 24 godziny. Dzieje się tak dlatego, że po pierwsze, likwidacja błędu we własnym dziele jest dla autora jakby sprawą honoru, a po wtóre: stwierdzenie, czy zgłaszany błąd rzeczywiście występuje jest bardzo łatwe, ponieważ można sięgnąć do źródeł.

W przypadku systemów zamkniętych sytuacja jest dokładnie odwrotna. Kod jest niedostępny dla użytkowników, a więc uznanie błędu zależy jedynie od dobrej woli producenta. Ten zaś przeważnie nie chce uznać czy potwierdzić istnienia luki w zabezpieczeniach, dopóki nie sprawdzą tego wewnętrzni eksperci. Dzieje się tak, ponieważ producent często nie chce „stracić twarzy", albo usiłuje dać sobie dodatkowy czas na przygotowanie poprawek. Jednak ich wydanie jest jedynie dobrą wolą producenta. Jeśli stwierdzi on, że wykryty błąd dotyczy np. jedynie 5% komputerów na świecie, może uznać, że jest nieistotny. Może też zalecić poczekanie do momentu pojawienia się nowszej wersji systemu.

Możliwość załatania błędu samodzielnie

Jest to kolejna zaleta oprogramowania Open Source. Ponieważ użytkownik ma dostęp do kodu źródłowego, może zlikwidować występujące u siebie błędy według własnego życzenia. Nawet jeśli nie posiada po temu odpowiednich kwalifikacji, może po prostu wynająć programistę, który to dla niego zrobi. W przypadku oprogramowania zamkniętego jest to nie tylko niemożliwe, ale nawet często prawnie zabronione, ponieważ większość licencji otwarcie zabrania wszelkich modyfikacji produktu. Tym samym uzależnia to użytkownika całkowicie od woli i umiejętności producenta oprogramowania.

Znajomość kodu

Znajomość kodu programu umożliwia pełną kontrolę tego, co program robi. Jest to bardzo istotne z punktu widzenia bezpieczeństwa, ponieważ znane są przypadki, gdy programy nabyte w celu ochrony sieci okazały się sprzyjać bardziej ich producentowi, niż użytkownikowi. Najbardziej spektakularnym przykładem jest firewall firmy Symantec o nazwie „Norton Internet Security 2000": dociekliwi użytkownicy tego programu, którego zadaniem jest blokowanie nieautoryzowanej komunikacji z siecią Internet, wykryli, że program ten, niezależnie od ustawień, zawsze przepuszcza ruch do serwerów kilku firm, wytwarzających oprogramowanie typu spyware, zbierające informacje z komputera użytkownika i wysyłające je do reklamodawców. Dekompilacja tego firewalla, nielegalna z punktu widzenia producenta, ujawniła wewnątrz kodu całą listę firm, z którymi, jak się potem okazało, firma Symantec miała podpisaną umowę. Właśnie na jej mocy, niektóre firmy miały uzyskać przywilej niekontrolowanego pobierania danych z komputera użytkownika. Wynika z tego, że bezpieczeństwo klienta może czasem zajmować niższe miejsce w hierarchii celów niż zysk producenta.

W przypadku oprogramowania otwartego taka sytuacja nie może mieć miejsca, ponieważ kod źródłowy oglądają setki oczu.

Uwolnienie od monopolu

Korzystanie z oprogramowania otwartego daje większy wybór. Zawsze istnieje kilka wersji programów, które mogą służyć do wykonywania tego samego zadania. Użytkownik nie jest też skazany na jeden system operacyjny (głównie Windows), a może wybrać jakąś dystrybucję Linuksa. Istnieje ich obecnie kilkadziesiąt, stworzonych w różnych celach, często wyspecjalizowanych w wykonywaniu jednego zadania, więc bardzo łatwo można wybrać coś, co pasuje do aktualnych potrzeb. Linux nie jest jednak jedyną alternatywą: programy Open Source mogą być przenoszone na inne platformy, np. Unix, FreeBSD czy NetBSD. Kolejnym atutem jest brak wymogu używania najnowszej wersji oprogramowania. Bardzo często okazuje się, że warto korzystać z wersji starszej, ale już sprawdzonej, niż eksperymentować z nowościami. Oprócz stabilnej pracy zyskuje się też możliwość wykorzystywania sprzętu o gorszych parametrach technicznych (np. firewall Freesco może wydajnie pracować na komputerach 486, co unaocznia, jak wiele potencjału drzemiącego w sprzęcie potrafią marnować nowoczesne programy i ich rozbudowane funkcje, z których się często nie korzysta). Producenci oprogramowania licencyjnego zdają się postrzegać tę zaletę Open Source jako zagrożenie dla siebie, ponieważ starają się za wszelką cenę przywiązać do siebie użytkownika. Doskonałym przykładem jest inicjatywa ostatnich dwóch lat, która wprowadza zwyczaj dzierżawy oprogramowania, zamiast jego nabywania na własność. W świetle tej nowinki prawnej, użytkownik nie staje się nigdy właścicielem programu; nabywa jedynie prawo do jego użytkowania na dany okres, o którego długości jednostronnie decyduje producent. Jako kuriozalny przykład monopolistycznego podejścia dużych producentów do klienta można podać Microsoft. Umowa licencyjna na pewien produkt tego koncernu zawarta była w kopercie, zamkniętej nalepką o następującej treści: „Rozerwanie tej nalepki oznacza akceptację umowy zawartej wewnątrz koperty". Bardzo często umowy tego

monopolisty zawierają wiele klauzul gwarantujących brak odpowiedzialności za wszelkie straty, poniesione przez użytkownika na skutek użytkowania programu, a nawet brak gwarancji, że program jest w jakikolwiek sposób użyteczny[40].

4.1.2 Słabości

Brak reklamy

Systemy i programy powstające jako darmowe oprogramowanie nie są nastawione na zysk. Z tego powodu nie istnieje zapotrzebowanie na ich reklamę. Jest to niewątpliwą słabością takich rozwiązań, ponieważ często są nieznane i zepchnięte na margines przez systemy komercyjne, które usilnie starają się zaistnieć w świadomości użytkowników. Znalezienie informacji o darmowym odpowiedniku oprogramowania komercyjnego nierzadko wymaga pewnego wysiłku. Nie wszystkich użytkowników na niego stać, przez co często mogą oni nie wiedzieć, że w ogóle coś tracą.

Wymagania wobec administratorów

Systemy darmowego oprogramowania nie są nastawione na masowego odbiorcę i z tego powodu administracja nimi nie jest łatwa, a przynajmniej wymaga dogłębnej wiedzy na określony temat. Jest to czynnik, który może odstraszać potencjalnych użytkowników, zwłaszcza administrację lub zarząd firmy, który może obawiać się, że koszty obsługi darmowego systemu mogą przerosnąć oszczędności uzyskane poprzez niskie koszty zakupu. Część z tych obaw może okazać się prawdziwa; administrator powinien posiadać wiedzę fachową i wiedzieć, co robi. Z drugiej strony jednak, brak wiedzy czy kwalifikacji administratora Windows, ujawni się z czasem. Microsoft jedynie sprawia wrażenie, że programami

40 *In extenso* fragment umowy licencyjnej systemu Windows brzmi: „(...) w żadnym przypadku Microsoft lub jego dostawcy nie będą odpowiedzialni za jakiekolwiek celowe, przypadkowe, pośrednie lub wynikowe uszkodzenia (obejmujące bez ograniczeń straty w zyskach, przerwanie działalności firmy, utratę danych firmowych i inne skutki) powstałe na skutek użytkowania Produktu lub braku możliwości użytkowania Produktu (...) nawet jeśli Microsoft był uprzednio poinformowany o możliwości zaistnienia takich uszkodzeń (...). W zakresie dozwolonym przez prawo właściwe, wyłączona jest także ustawowa odpowiedzialność z tytułu rękojmi za to, że Oprogramowanie wraz z innymi materiałami (...) są odpowiedniej jakości i nadają się do użytku".

jego produkcji administruje się łatwo[41]. W rzeczywistości jednak brak solidnych podstaw praktycznie gwarantuje niepowodzenie, a inwazje wirusów *Beagle*, *Blaster* czy *Sasser*, które w błyskawicznym tempie opanowały miliony komputerów na świecie, w tym serwery *update.microsoft.com* są tego dowodem.

Brak jednego właściciela

Jest to główna słabość w oczach decydentów, który wybierają oprogramowanie nabywane przez firmy. Bardzo często są to osoby z kręgów finansowych, a nie informatycznych. Z reguły nie są w stanie ocenić programu Open Source pod względem merytorycznym i porównać go na tej samej płaszczyźnie z oprogramowaniem komercyjnym. Żywią natomiast obawy przed programem, za którym nie stoi firma-właściciel, do której można by skierować swoje roszczenia, gdyby jego działanie odbiegało od specyfikacji. Często wydaje im się również, że program dostępny za darmo nie może być lepszy od produktu, za którym stoi duży kapitał. Błędność takiego przekonania można łatwo wykazać, wskazując ponownie na fragmenty umowy licencyjnej Microsoftu, przytoczone w rozdziale 4.1.1 na stronie 90: nietrudno zauważyć, że wobec takiej jej wymowy, uzyskanie jakiejkolwiek rekompensaty ze strony producenta oprogramowania jest niezwykle mało prawdopodobne.

4.1.3 Szanse

Otwarte standardy

Systemy Open Source bazują na otwartych standardach wymiany danych. Takimi standardami są na przykład międzynarodowe normy. Oznacza to całkowitą swobodę współpracy pomiędzy programami – w odróżnieniu od sytuacji, jaka ma miejsce w systemach zamkniętych, preferujących własne, wewnętrzne rozwiązania. Typowym przykładem może być niemożność otwarcia w programie X dokumentu, utworzonego w programie Y, będącym produktem innej firmy. Jest

41 *Vide:* cytat z artykułu T. Polusa w rozdziale 3.4 na s. 57.

to szczególnie dotkliwe, jeśli wiąże się z tym konieczność nabycia przez użytkownika programu czy nawet całego systemu operacyjnego, którego używa partner biznesowy lub nawet instytucja publiczna.

Niezależność technologiczna

Niezależność technologiczna jest kolejną szansą oprogramowania Open Source. Program otwarty, nie posiadający właściciela i o otwartym, szeroko dostępnym kodzie źródłowym nie może zniknąć, o ile będzie istniało zainteresowanie tym programem. Natomiast jak najbardziej może zniknąć program, którego właściciel zbankrutuje, lub straci zainteresowanie jego dalszym rozwojem. Taka sytuacja była udziałem programu Proxomitron, który w systemie Windows pełnić może podobne cele, jak Privoxy, tj. filtrować strony internetowe pod kątem reklam i szkodliwych załączników. Program był dostępny za darmo, ale bez otwartego kodu. Jak wynika ze słów autora na witrynie internetowej programu, projekt został zamknięty, ponieważ autor będzie się teraz zajmował czymś innym. Mimo sporego zainteresowania użytkowników, w tym – programistów chcących rozwijać program dalej, jego autor, który ma do tego pełne prawo, nie zgodził się ujawnić kodu źródłowego. W związku z taką sytuacją, nowe wersje nie powstaną już nigdy. Jeśli wobec tego zaistnieją jakieś nowe sposoby załączania reklam do stron internetowych, Proxomitron nie będzie w stanie ich obsłużyć. Mimo, iż na razie jest bardzo dobry, stanie się nieskuteczny i zniknie z rynku[42].

Proxomitron może być znany jedynie informatykom, ale nie jest jedyny. Teoretycznie nic nie stoi na przeszkodzie, aby identyczny los spotkał całkiem dużą aplikację czy pakiet biurowy. Dwukrotnie bardzo blisko takiej sytuacji znalazł się szeroko używany kiedyś w Polsce i w Stanach Zjednoczonych edytor tekstu WordPerfect – po raz pierwszy po bankructwie WordPerfect Corporation, a po raz drugi gdy został przejęty, a następnie dość szybko porzucony przez firmę Novell. Te

42 Natomiast opisywany tu program Privoxy, posiadając otwarte źródła, został przeniesiony również i na platformę Windows.

zawirowania wystarczyły, aby zaniepokojeni użytkownicy przestawili się na oprogramowanie konkurencyjne.

Oprogramowanie Open Source wyklucza taki bieg wydarzeń, pod warunkiem, że istnieje przynajmniej jedna osoba zainteresowana rozwojem danego programu. W przypadku programów mniej popularnych, korzystanie z oprogramowania otwartego wyraźnie zmniejsza ryzyko uzależnienia użytkownika od woli producenta.

Wzrost świadomości społecznej

Wzrost świadomości społecznej wynika z rozwoju społeczeństwa doby informacyjnej. Coraz więcej osób ma dostęp do informacji pochodzących z niezależnych źródeł. W połączeniu z rosnącymi umiejętnościami odsiewania informacji istotnych od szumu (m. in. reklamowego), w społeczeństwie wykształca się krytycyzm i samodzielne myślenie. Prowadzi to do częściowego uniezależnienia się społeczeństwa od wizji marketingowych, rozsnuwanych przez producentów oprogramowania komercyjnego – w tym Microsoft, które nie zawsze odpowiadają sytuacjom realnym. Użytkownicy zauważają oprogramowanie alternatywne i potrafią dokonywać racjonalnego wyboru.

Nastroje antymonopolowe

Schemat postępowania firm (nie tylko informatycznych) wobec klienta zwykle jest bardzo podobny: jednym z celów zawsze jest przywiązanie klienta do siebie. W przypadku, jeśli pozycja firmy jest już ugruntowana, polityka ta nabiera cech jeszcze bardziej radykalnych. Przykładem może być wprowadzona przez Microsoft polityka obowiązkowej aktywacji produktu, nawet po pewnych zmianach dokonanych w sprzęcie, na którym pracuje system. Można przewidywać, że dalekosiężnym celem Microsoftu może być chęć wymuszania na kliencie zakupu nowych wersji programów, niezależnie od tego, że wersja stara może nadal dobrze spełniać swoją rolę, aby w ten sposób sztucznie utrzymać wysoki obrót. Ten sam

skutek Microsoft uzyskuje bezwzględnie usuwając z rynku starsze wersje swojego systemu, który mógłby być, na przykład, sprzedawany po niższej cenie, pozostawiając jedynie wersję najnowszą. Zwiększająca się świadomość społeczna powoduje powstanie nastrojów antymonopolowych, które skłaniają klientów do wybierania rozwiązań otwartych. Tego typu inicjatywy są wspierane w Polsce np. przez organizację Ruch Wolnego Oprogramowania (RWO)[43].

Rozwój Internetu

Internet jest uznany za szansę w analizie SWOT, ponieważ stał się tu narzędziem komunikacji, które umożliwiło wymianę informacji oraz koordynację działań podczas tworzenia oprogramowania przez rozproszone zespoły programistów. Dzięki Internetowi zniknęły bariery czasowe i geograficzne, a społeczność otwartego oprogramowania zyskała wielu członków.

Niskie bariery wejścia na rynek

Otwarty kod programu umożliwia programistom uczenie się między innymi przez studiowanie kodu napisanego przez bardziej zaawansowanych kolegów. Dzięki temu zjawisku, zunifikowany kod może być tworzony bez posiadania zaawansowanej i zorganizowanej hierarchicznie infrastruktury. Wykorzystywanie otwartych standardów nie wiąże się z żadnymi kosztami pośrednimi. Z tych powodów bariery wejścia na rynek są niskie.

4.1.4 Zagrożenia

Patenty na oprogramowanie

Patenty na oprogramowanie oznaczają w gruncie rzeczy opatentowanie pewnych "rozwiązań programistycznych". Ich wprowadzenie przez Parlament Europejski mogło by mieć taki skutek, że, w uproszczeniu, niemożliwe byłoby stworzenie od zera programu Open Source, który wykonywałby podobne działanie jak program zamknięty, na który uzyskano by wcześniej patent. Przy-

43 Ruch na rzecz Wolnego Oprogramowania: http://www.rwo.org.pl.

kładem może być podwójne kliknięcie myszą, na które, według prawa USA posiada patent firma AMAZON. Wprowadzenie podobnych przepisów przez Parlament Europejski oznaczałoby to spowolnienie rozwoju informatyzacji oraz umocnienie pozycji firm monopolistycznych. Problem, który pojawia się w Unii Europejskiej, opisany na stronach Ruchu na rzecz Wolnego Oprogramowania, jest pojmowany przez coraz większą część społeczeństwa, które zdaje sobie sprawę z jego szkodliwości i wpływu na umocnienie pozycji jedynie kilku koncernów.

Agresywna polityka Microsoftu

Dostrzegając w oprogramowaniu Open Source zagrożenie własnych interesów, duże koncerny, w tym Microsoft, starają się, oprócz legalnego konkurowania, wykorzystać swój potencjał w osłabianiu pozycji konkurentów. Jednym z przejawów takiego postępowania jest polityka określana jako FUD (Fear, Uncertainty, Doubt), czyli obawa, niepewność, wątpliwości. Polega ona na wykorzystywaniu prasy (w tym własnych wydawnictw, np. "Microsoft Today"), środków masowego przekazu i innych technik po to, aby u klienta zasiać niepewność co do produktów konkurencji. Informator twierdzi na przykład, że system konkurencyjny jest w zaniku, ponieważ firma go rozwijająca traci pozycję, albo nie jest nim zainteresowana.

Bardzo dobrym przykładem takiego podejścia Microsoftu do produktów konkurencji był popularny swego czasu w Polsce pakiet biurowy firmy Lotus, zawierający produkty Ami Pro oraz Lotus123, przeznaczony dla Windows 3.1. Po wymianie systemu na Windows 95 (którego jednym z celów była kompatybilność wstecz, czyli możliwość uruchamiania starszych programów) i próbie instalacji pakietu Lotus, użytkownik otrzymywał na ekranie komunikat mówiący, ze "ten program może być niekompatybilny z systemem Windows 95 i nie działać poprawnie". Pakiet po instalacji sprawował się dobrze, ale komunikat wywierał negatywne wrażenia u użytkownika – mimo, iż nie było ku temu żadnych

podstaw[44]. Warto zauważyć, że komunikat ten nie pojawiał się przy instalacji starszych wersji oprogramowania MS Office.

Innym czynnikiem zagrożenia, który obejmuje niniejszy punkt jest lobbowany przez duże koncerny w USA projekt Trusted Computing. Miałby on wymuszać na producentach sprzętu instalowanie w elementach komputera bliżej nie sprecyzowanych elementów "zabezpieczających", które miałyby umożliwiać wgląd w przechowywane i obrabiane na nich dane. Projekt ten, obecnie reklamowany jako zorientowany na "bezpieczeństwo w Internecie" mógłby posłużyć do inwigilacji oraz np. wymuszania stosowania programów "zatwierdzonych" przez autorów tej inicjatywy. W rzeczy samej, skutki przeforsowania Trusted Computing mogłyby być bardzo niebezpieczne dla Otwartego Oprogramowania.

Szkodliwa polityka RP

Rząd i Parlament Polski zdają się nie dostrzegać problemu w uzależnianiu instytucji publicznych od rozwiązań zamkniętych. Przykładem jest informatyzacja ZUSu i program Płatnik.

Stworzenie tego programu wyłącznie pod platformę Windows zmusiło wiele firm do zakupu osobnych komputerów z systemem Microsoftu wyłącznie po to, aby móc rozliczać się z ZUSem[45]. Ruch na rzecz Wolnego Oprogramowania ocenia, że w tym celu zakupiono w Polsce sto tysięcy kopii systemu Windows, którego wartość wynosi od 400 do 1000 zł za pojedyncze stanowisko[46].

Jako alternatywa programu Płatnik, powstał w Polsce program otwartym kodzie źródłowym, zatytułowany Janosik, który z założenia miał umożliwić rozliczanie się z ZUSem, a jednocześnie pracować na platformach Linux, Windows, FreeBSD i Mac OS, oraz innych, jeśliby zaszła taka potrzeba. Jeden ze

44 Gdyby kod Windows był otwarty, można by zobaczyć, dla jakich innych produktów konkurencji Microsoft przewidział wyświetlanie podobnego komunikatu.

45 Istnieje wiele firm, które zajmują się grafiką lub składem czasopism, które posługiwały się wyłącznie systemami firmy Apple – na który programu Płatnik nie przewidziano.

46 Ruch na rzecz Wolnego Oprogramowania, *op. cit.*

współautorów projektu, Sergiusz Pawłowicz, powołując się na "Ustawę o dostępie do informacji publicznej" zwrócił się z prośbą do ZUS o specyfikację techniczną formatu przesyłania danych w programie Płatnik w marcu 2002. Odpowiedzi na swoją petycję nie otrzymał do czerwca 2004 z powodów czysto proceduralnych[47]. ZUS argumentował, że nie jest w stanie podać tej specyfikacji, ponieważ jej nie posiada, gdyż autor, firma Prokom, dostarcza program w formacie zamkniętym.

Powyższy przykład stawia polskie prawo i instytucje publiczne RP w bardzo złym świetle. Wynika bowiem z niego, że administracja państwowa nie popiera wolnej konkurencji oraz jest skłonna wydawać publiczne pieniądze na projekty, nad którymi nie sprawuje potem żadnej kontroli.

Nieodpowiedni system edukacji

Jak podają źródła Ruchu na rzecz Wolnego Oprogramowania[48], nauczanie w polskich szkołach przedmiotów informatycznych sprowadza się przeważnie do poznawania produktów firmy Microsoft. Powoduje to wzrastanie nowych pokoleń, które są uzależnione od monopolisty i miały styczność wyłącznie z oprogramowaniem zamkniętym.

Zamykanie standardów przez monopolistów

W walce o utrzymanie przy sobie klienta, firmy informatyczne mogą wykorzystywać zamykanie powszechnie przyjętych standardów wymiany czy zapisu danych poprzez umieszczanie w nich sobie tylko właściwych dodatków. Działanie takie polega na przykład na tym, że rozpowszechniany jest darmowy program do tworzenia stron WWW. Jednak stwarzane przy jego pomocy strony funkcjonują prawidłowo jedynie na serwerach wyposażonych w oprogramowanie monopolisty, np. Microsoft IIS Server Extensions. W ten sposób wymusza się zakup oprogramowania odpowiedniego typu. Podobną taktyką jest zmiana typu formatu zapisu dokumentu, np. .doc, w nowej wersji edytora tekstu, wymuszająca

47 W momencie pisania niniejszej pracy (lipiec 2004) sprawa nadal była w toku, a NSA, do którego sprawa dotarła w międzyczasie orzekł, że nie jest organem właściwym do jej rozpatrywania i skierował ją do sądu powszechnego.

48 Ruch na rzecz Wolnego Oprogramowania: *op. cit.*

kupowanie nowych wersji produktu u użytkowników, otrzymujących od swoich korespondentów dokumenty, których nie są w stanie otworzyć.

4.2 Wnioski

Wyniki analizy SWOT charakteryzują potencjał Otwartego Oprogramowania. Tabela SWOT numer 1, na stronie 84, wykazuje przewagę Atutów nad Słabościami. Wynika ona przede wszystkim z głównej idei Open Source, która daje swobodę, samodzielność i wsparcie, a przy tym jest atrakcyjna ekonomicznie. Jednakże czynniki zewnętrzne, w Tabeli SWOT numer 2 na stronie 85 wykazują, że przewaga Szans nad Zagrożeniami jest bardzo nikła. Jest to skutkiem sytuacji prawnej w Polsce oraz ugruntowanej pozycji czynników monopolistycznych i ich umiejętności jej wykorzystania w celu ochrony swoich interesów.

Strategią Otwartego Oprogramowania powinna być więc działalność, która będzie osłabiała Zagrożenia i pozwalała Atutom wykazać ich możliwości. Czynnikiem, który powinien wpłynąć na ograniczenie Zagrożeń byłoby podwyższenie świadomości społecznej. Świadomość społeczna, stymulowana odpowiednio wcześnie przez dostęp do informacji oraz oświatę, powinna wpłynąć na zwiększenie się Szans. Jednocześnie, mając swój wyraz w wpływaniu na urzędy publiczne i ustawodawstwo, mogłaby pośrednio wpłynąć na częściową niwelację Zagrożeń, na przykład przez odpowiednią modyfikację prawa. Brak edukacji oraz wynikający z niego brak uświadomienia sobie sytuacji w społeczeństwie mają skutek dokładnie odwrotny.

Sytuację mogłaby wydatnie polepszyć kampania informacyjna oraz popularyzacja prowadzone przez czynniki niezależne od kapitału i ugrupowań lobbystycznych albo przez organizacje, mające na względzie szeroko pojęty interes państwa polskiego w ujęciu długofalowym. Kampania taka mogłaby doprowadzić do nacisków na przykład na ugrupowania parlamentarne, które poparłyby takie inicjatywy, uświadomiwszy sobie znaczenie rozwoju potencjału miejscowego

zamiast przysparzania zysku koncernom, mających swoje siedziby poza granicami kraju. Efektem takich działań byłyby zapisy prawne, zapewniające popyt na rozwiązania alternatywne, przyczyniające się następnie do lepszego zaspokojenia potrzeb rynkowych oraz spadku cen i poprawie jakości usług wykonywanych dla instytucji publicznych.

Rozdział 5. Zakończenie

Zamierzeniem pracy było wykazanie, że do zabezpieczenia sieci lokalnej o małej skali możliwe jest wykorzystanie oprogramowania dostępnego nieodpłatnie. Jest to możliwe głównie dzięki istnieniu Internetu. Ta światowa sieć komputerowa po pierwsze umożliwia tworzenie Otwartego Oprogramowania przez programistów na całym świecie. Po drugie, dostarcza możliwości jego nieograniczonego rozpowszechniania. Po trzecie, oferuje bardzo wyczerpującą i kompletną informację o tym, jak z tego oprogramowania skorzystać.

Ponad wszelką wątpliwość, przeciętny administrator sieci ma dużą możliwość wyboru. Jedną z alternatyw jest skorzystanie z pakietów komercyjnych, które można nabyć w pudełku w postaci gotowej do użycia. W niniejszej pracy starano się wykazać, że nie musi być to rozwiązanie najlepsze. Programy komercyjne bardzo często izolują ich użytkownika od prawdziwej płaszczyzny swojego działania. Takie podejście producentów jest jak najbardziej zrozumiałe. Można jednak stwierdzić, że czytanie dokumentacji programów Open Source daje administratorowi pełniejsze zrozumienie podejmowanych przez siebie działań oraz przyczynia się do zapewnienia lepszego bezpieczeństwa.

Siła zabezpieczenia sieci, podobnie jak i zagrożenia, są bowiem w głównej mierze zależne od czynnika ludzkiego: na szkodę sieci działają bowiem ludzie, w związku z czym do innych ludzi należy stworzenie ochrony. Jest to zadanie, którego nie można przenieść na najlepszy i najbardziej nawet dopracowany program komputerowy. Jak starano się wykazać, bezpieczeństwo sieci nie jest stanem trwałym. Należy je raczej postrzegać jako rodzaj dynamicznej równowagi sił. Dlatego też działania administratora nie kończą się na zainstalowaniu programu ochronnego czy na jednorazowym skonfigurowaniu systemu, ale na jego ciągłej konserwacji. Codziennie pojawiają się w sieci nowe rodzaje zagrożeń oraz wykrywane są błędy w oprogramowaniu. Stosowane metody ochrony starzeją się

i dezaktualizują. Zadaniem administratora sieci jest więc bycie w stałym kontakcie z nowymi technologiami oraz śledzenie wiadomości z zakresu bezpieczeństwa, aby jak najszybciej dowiadywać się o nowych problemach i o sposobach ich rozwiązywania. Oprogramowanie Otwarte stanowi wspaniały materiał do tego celu oraz oferuje potencjał, który często jest nieporównywalny z oprogramowaniem komercyjnym.

Dziedziną bezpieczeństwa, która wymaga w chwili obecnej najszybszego reagowania jest ochrona antywirusowa. Ilość wirusów pojawiających się w sieci w ciągu miesiąca wynosi obecnie dziesiątki, jeśli nie setki mutacji. Większość liczących się serwisów antywirusowych jest zmuszona udostępniać aktualizację swoich baz przynajmniej raz na dobę, a czasem krytyczne uaktualnienia pojawiają się nawet kilkakrotnie w ciągu dnia. Prowadzi to do stwierdzenia, że wkrótce może zostać przekroczona pewna granica, za którą programy antywirusowe działające na dzisiejszych zasadach przestaną być skuteczne, ponieważ tworzące je firmy nie będą w stanie zdążyć na czas z pisaniem szczepionek. W takiej sytuacji rdzenna, wewnętrzna odporność systemu i szybkość reakcji jego autorów na sygnalizowane zagrożenia staje się kwestią kluczową dla bezpieczeństwa, przy czym, jak wykazano, systemy o otwartym kodzie źródłowym potrafią wyprzedzać w tej dziedzinie oprogramowanie komercyjne w znacznym stopniu.

Spis ilustracji

Spis tabel

Bibliografia

Bauer D. M.	„Linux – Bezpieczeństwo serwerów", RM, Warszawa 2003
Fisher B.	„Przestępstwa komputerowe i ochrona informacji" Kantor Wydawniczy Zakamycze 2000
Hontanon R. J.	„Bezpieczeństwo systemu Linux", Mikom, Warszawa 2002
Hunt C.	„Serwery sieciowe Linuksa", Mikom, Warszawa 2000
Lipski P.	„Hardening Win2000", Internet, comp.os.ms-windows.nt.admin.security
Pełech T.	„Polityka bezpieczeństwa danych w firmie – aspekt organizacyjny i prawny" , Gazeta IT, Warszawa 2003
Podstawczyński A.	„Linux w sieci", Helion, Gliwice 2002
Podstawczyński A.	„Linux – rozwiązania praktyczne", Helion, Gliwice 2002
Polus T.	„Łamanie haseł w NT", IT-FAQ, www.it-faq.pl
Polus T.	„Dobry firewall czy bubel wszechczasów?", IT-FAQ, www.it-faq.pl
Serwis Cert-Polska,	Internet, www.cert.pl
Serwis F-secure,	Internet, www.f-secure.com
Serwis Freesco,	Internet, www.freesco.pl
Serwis MKS,	Internet, www.mks.com.pl
Serwis Privoxy,	Internet, www.privoxy.org
Serwis RWO,	Internet, www.rwo.org.pl
Sharpe J., Potter T.	„Samba dla każdego", Helion, Gliwice 2002
Suwiński P.	„Otwarte Oprogramowanie w biznesie", Politechnika Gdańska,Internet, www.flug.org.pl/