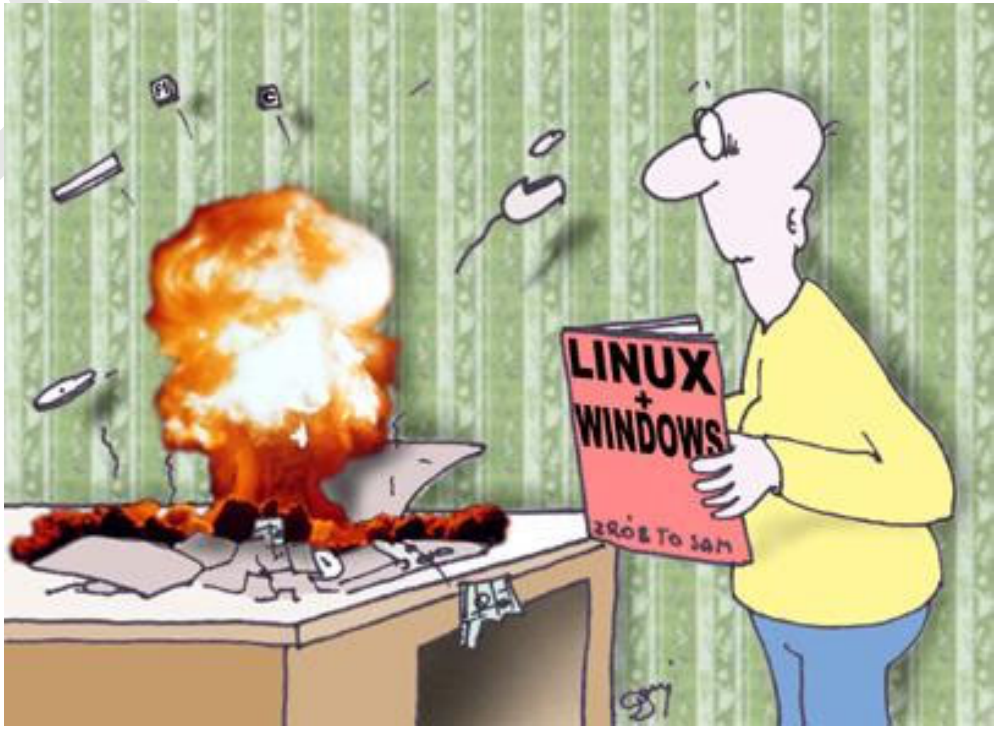
Podstawowe problemy bezpieczeństwa systemów operacyjnych



Zagadnienia

1. **Naruszenia bezpieczeństwa**
   - typowe formy ataku na system operacyjny
   - komponenty szczególnie podatne
2. **Problemy uwierzytelniania i kontroli dostępu**
3. **Wirusy i konie trojańskie**
4. **Zamaskowane kanały komunikacyjne**

Naruszenia bezpieczeństwa

Klasy naruszeń bezpieczeństwa systemu operacyjnego

1. **Włamania i kradzieże danych**
2. **Destrukcja systemu operacyjnego lub jego komponentów czy aplikacji**

Naruszenia bezpieczeństwa

Zagrożenia dla bezpieczeństwa

1. błędy i luki bezpieczeństwa („dziury”) w komponentach systemu lub w ich konfiguracji
2. furtki (backdoor)
3. konie trojańskie
4. wirusy oraz bomby logiczne i czasowe

Naruszenia bezpieczeństwa

Scenariusz typowego ataku na system operacyjny

1. zlokalizowanie systemu do zaatakowania
2. wtargnięcie na konto legalnego użytkownika
   (brak hasła, złamanie łatwego hasła, podsłuchanie hasła)
3. wykorzystanie błędów i luk w komponentach systemu lub w ich konfiguracji w celu uzyskania dostępu do konta uprzywilejowanego
4. wykonanie nieuprawnionych działań
5. zainstalowanie furtki dla bieżącego lub przyszłego wykorzystania
6. zatarcie śladów działalności (usunięcie zapisów z rejestrów systemowych)
7. ataki na inne komputery

Naruszenia bezpieczeństwa

Ewolucja typowych ataków na przestrzeni lat

1988÷1993:	❍ wykorzystanie trywialnych haseł i znanych słabych punktów systemu ❍ instalowanie programów przechwytujących ruch sieciowy (sniffer) ❍ analiza kodu źródłowego narzędzi systemowych w celu odkrycia nieznanych „dziur”
1994:	❍ ataki poprzez sniffery – przechwytywanie haseł ❍ podsuwanie koni trojańskich ❍ ataki na sendmail ❍ ataki poprzez NFS i NIS
1995:	❍ IP spoofing, DNS spoofing ❍ ataki na routery, nameservery, IP source routing, IP source porting
1996:	❍ fragmentacja pakietów IP, IP hi-jacking ❍ denial of service, ping of death, SYNC flood
1998:	❍ web-jacking, e-mail spamming
od 2002:	❍ web-phishing (*personal data fishing*), e-mail spamming

Błędy implementacji

Najczęstsze obszary błędów implementacyjnych

Sieciowe komponenty systemu, np. stos TCP/IP

- Bonk – atak na stos protokołu TCP/IP firmy Microsoft, który może spowodować awarię systemu zaatakowanego komputera

lub NetBIOS:

- WinNuke – atak umożliwiający wywołanie awarii starszych wersji systemu Windows przy użyciu usług systemu NetBIOS

albo RPC:

- RDS_Shell – Metoda wykorzystania składnika Remote Data Services, należącego do MDAC (Microsoft Data Access Components), umożliwiająca zdalnemu napastnikowi uruchamianie poleceń z uprawnieniami systemowymi

Wrażliwe usługi systemowe i narzędziowe

- usługi informacyjne: netstat, systat, nbtstat, finger, rusers, showmount, ident
- usługi konfiguracyjne: bootparam, dhcp, zdalne zarządzanie
- RPC (rpcinfo, rexec) → Secure RPC, DCE RPC
- NFS (export) → Secure NFS (ONC)
- SMB (netview, smbclient) → CIFS
- NIS → NIS+
- zdalny dostęp: rlogin, rsh (brak rejestracji dostępu), system zaufania → ssh
- transfer plików i przesyłek pocztowych: rcp → scp, ftp → aftpd , sendmail → postfix
- X window: xhost → ssh

Zdalna detekcja systemu operacyjnego

Metody rozpoznawania systemu

http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Aktywne

- poprzez inicjowanie a następnie analizowanie połączeń (na ogół specjalnie spreparowanych)
- skierowane wobec konkretnych stanowisk

Pasywne

- poprzez podsłuch pakietów pochodzących z analizowanego systemu
- mniej precyzyjne ale trudnowykrywalne

Zdalna detekcja systemu operacyjnego

Usługi informacyjne

- charakterystyczne powitanie (*banner*), np. ident, sendmail

Badanie RTO (*Retransmission Time-Out*)

- pomiar czasów pomiędzy retransmisjami pakietów SYN+ACK w trakcie nawiązywania odpowiednio spreparowanego połączenia
- ilość retransmisji i odstępy między nimi należą do „cech osobniczych” systemu operacyjnego
- np. MacOS X wysyła segment RST
- a Windows i Linux milcząco zamykają wpółotwarte połączenie

Zdalna detekcja systemu operacyjnego

Kamuflaż

- security trough obscurity !

stealth patch – łata na jądro Linuksa:

- blokuje nieprawidłowe pakiety ACK
- blokuje pakiety z flagami SYN i FIN
- blokuje pakiety z niepoprawnymi flagami i kombinacjami flag
- blokuje pakiety ICMP (za wyjątkiem echo)

IP Personality – zestaw łat na jądro Linuksa i iptables (nie rozwijany):

Uwierzytelnianie

Unix

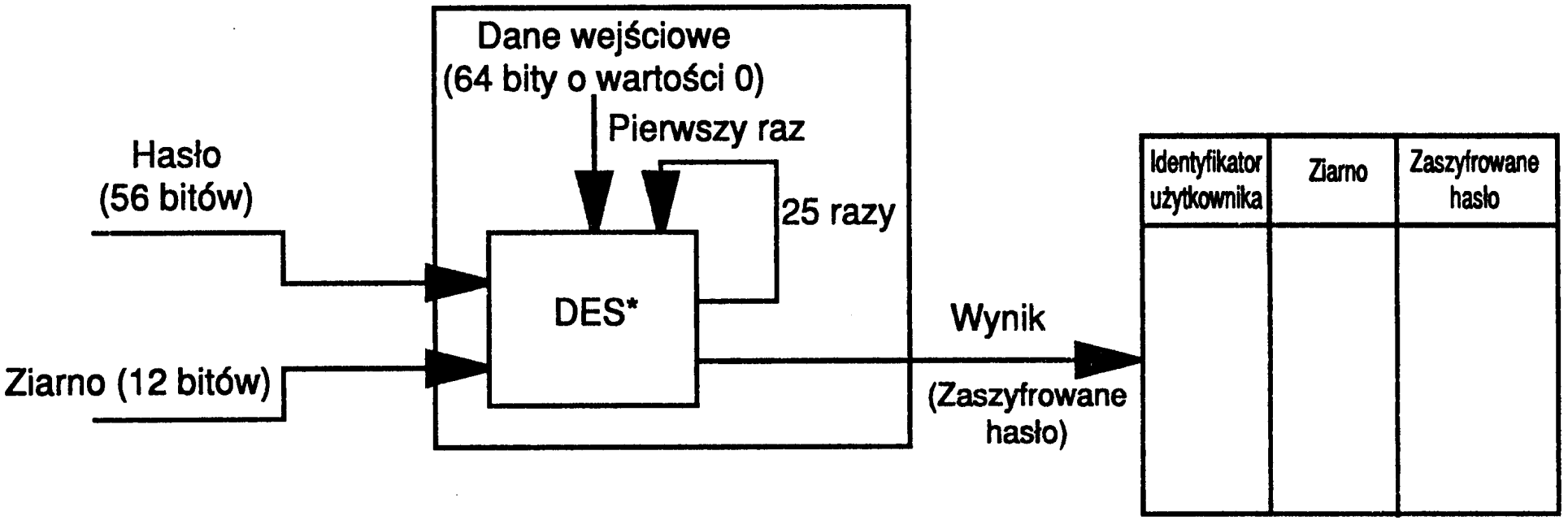
Przechowywanie haseł (mechanizm klasyczny):

- użytkownik wybiera 8 znakowe hasło
- hasło jest zamieniane na 56b ciąg za pomocą 7b kodu ASCII
- powstały 56b ciąg jest kluczem algorytmu DES
- e-blok algorytmu DES jest modyfikowany za pomocą 12b wartości – ziarna (ang. *salt*) ustalanego na ogół na podstawie bieżącego czasu
- tak zmodyfikowany algorytm DES jest wykonywany na 64b bloku złożonym z samych zer
- wyjście podaje się na wejście kolejnej iteracji (25 iteracji)
- 64b wynik transformowany na 11-znaków z alfabetu 64 znakowego (A-Z, a-z, 0-9, '.', '/')

Uwierzytelnianie

Unix

Zapamiętywanie hasła:



*DES: Data Encryption Standard

mapujemy wiele wartości w jedną – przekształcenie jest nieodwracalne

Uwierzytelnianie

Unix

Złamanie hasła

Metoda przeszukiwania wyczerpującego (*brut-force attack*):

- załóżmy moc obliczeniową o wydajności 6,4 miliona iteracji DES na sekundę
- 8 znaków z 36-znakowego alfabetu: 36^8 czyli ok. 2,8 biliona kombinacji
- 25 iteracji dla każdego kombinacji – 1 milion kombinacji w 4 sek.
- dowolne hasło zostanie złamane (2,8 biliona kombinacji) w 4 miesiące

Uwierzytelnianie

Unix

Usprawnienia

Ukrycie haseł poza dostępem zwykłego użytkownika:

- /etc/shadow (przeniesienie haseł do oddzielnej lokalizacji)
- np. pakiety shadow-in-a-box, Linux Shadow Password Suite, ...
- możliwe ataki na shadow (wykorzystują głównie luki w usługach, obrazy *core dump*)
- centralne bazy katalogowe, np. NIS, NIS+, LDAP

Testery jakości haseł:

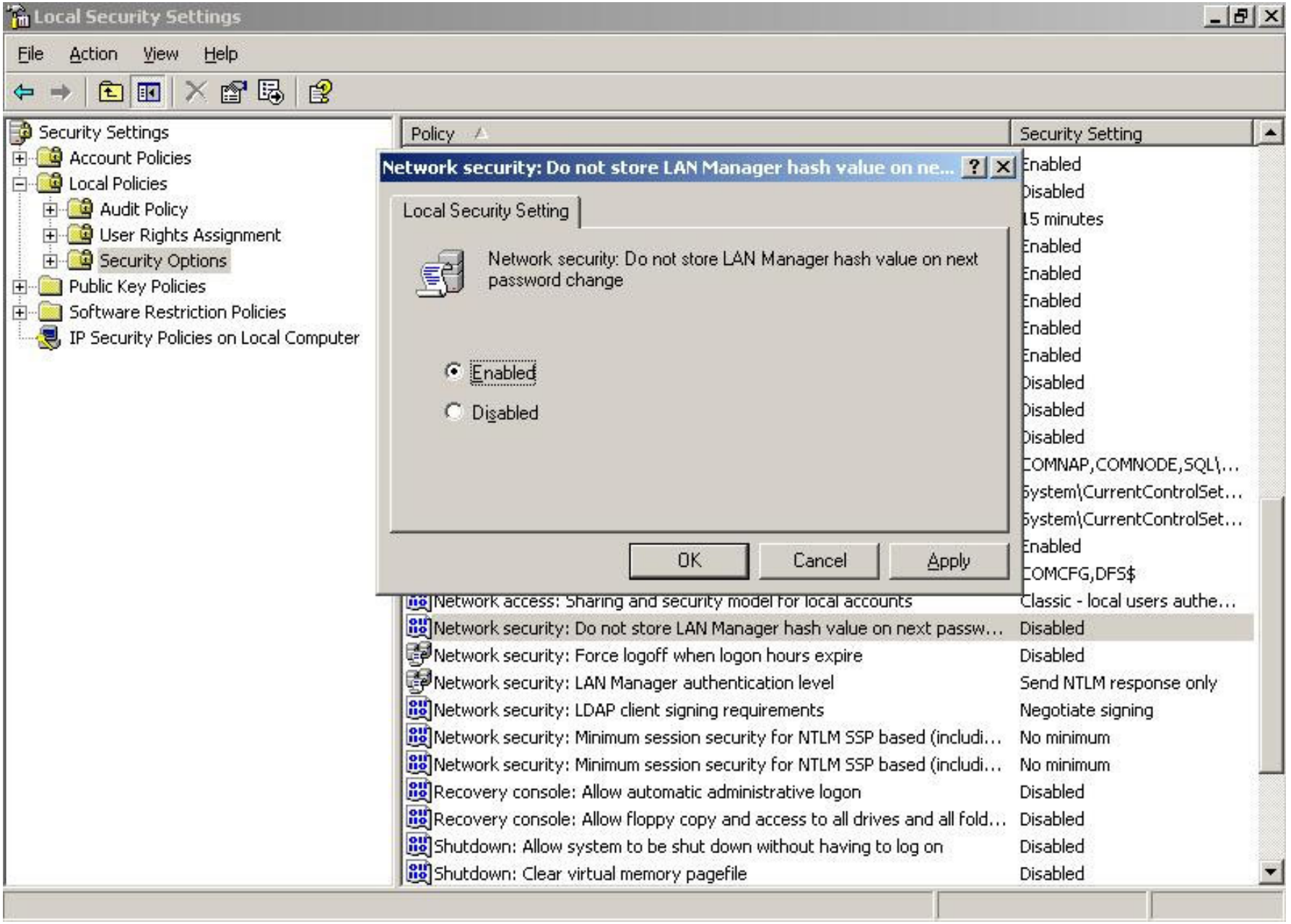
- passwd+ (zastępuje passwd), anlpasswd, npasswd

Uwierzytelnianie

MS Windows NT 4.0

- niezakodowane hasła przechowywane są w zastrzeżonym obszarze LSA (Local Security Authority) rejestru: `HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets` dostępnym tylko dla usługi Security Accounts Manager
- od wersji NT 4.0 hasła mogą być również zakodowane (tzw. *hash*) funkcją MD5 (z ziarnem i z 40b kluczem RSA) i przechowywane w rejestrze: `HKLM\SAM` oraz w zastrzeżonym obszarze systemu plików NTFS: `%SYSTEMROOT%\SYSTEM32\CONFIG\SAM`
- dla zachowania zgodności wstecz (z linią 9x) obok postaci hash MD5 umieszczane są kopie haseł – LM hash – zakodowane autorskim algorytmem LanMan (z protokołu NT LanManager, bez ziarna)
- przechowywanie postaci LM hash można wyłączyć

Uwierzytelnianie



Uwierzytelnianie

MS Windows NT 4.0

Usprawnienia (SP2 i SP3):

- opcja *syskey* (SP2) – postacie hash można zaszyfrować kluczem 128b kluczem RSA SYSTEM KEY (zamiast kluczem 40b) – to znacznie utrudnia łamanie haseł (ale i tak dostępne są gotowe narzędzia, choć bardziej złożone niż uprzednio)
- filtr słabych haseł Passfilt.dll (SP3)
- niestety ciągle wiele luk i gotowych narzędzi *exploit* (wykorzystują uprawnienia administracyjne do importu plików SAM, luki w systemie tworzenia kopii zapasowych systemu RDISK, itp.)

Uwierzytelnianie

MS Windows 2000, XP, 2003

- algorytm uwierzytelniania NTLM został zastąpiony przez Kerberos TGP
- przechowywanie postaci LM hash domyślnie wyłączone
- starsze wersje Windows nie obsługują Kerberosa, stąd mogą nadal wymagać LM hash
- opcja syskey domyślnie włączona

Uwierzytelnianie

Novell NetWare

- w bazie katalogowej NDS (*NetWare Directory Service*) przechowywany jest skrót hasła użytkownika z ziarnem (jest nim identyfikator użytkownika) oraz para kluczy RSA
- najpierw stacja sieciowa uwierzytelnia się w imieniu użytkownika wobec wybranego serwera (NetWare od wersji 4 stosuje SSO) metodą zawołanie-odzew
- następnie serwer przesyła klucz publiczny NDS (K_{NDS}), a stacja losuje klucz jednorazowy K_1, który prześle serwerowi zaszyfrowany kluczem K_{NDS}
- K_1 posłuży serwerowi do bezpiecznego przekazania klucza prywatnego RSA użytkownika
- stacja przekształca klucz RSA do postaci GQ – Gillou-Quisquater (asymetryczny algorytm uwierzytelniania – NetWare od wersji 4 nie stosuje RSA) i natychmiast usuwa z pamięci hasło wraz z kluczem RSA
- klucz GQ posłuży do uwierzytelniania użytkownika w dostępie do zasobów sieci

Prawa dostępu do zasobów

Unix

Standard POSIX (*Portable Operating System Interface*) 1003.1

- prawa: r w x
- kategorie użytkowników: u g o
- dodatkowo Set User Id, Set Group Id, Sticky
- niektóre aplikacje oferują własne rozszerzenia tego modelu uprawnień, np. ProFTPD (http://proftpd.linux.co.uk/)

Rozszerzone implementacje:

- ACL: AIX, Solaris
- MAC: Trusted Solaris, Trusted IRIX, Ultrix, HP-UX, Xenix

Prawa dostępu do zasobów

Unix

POSIX 1003.1e/1003.2c

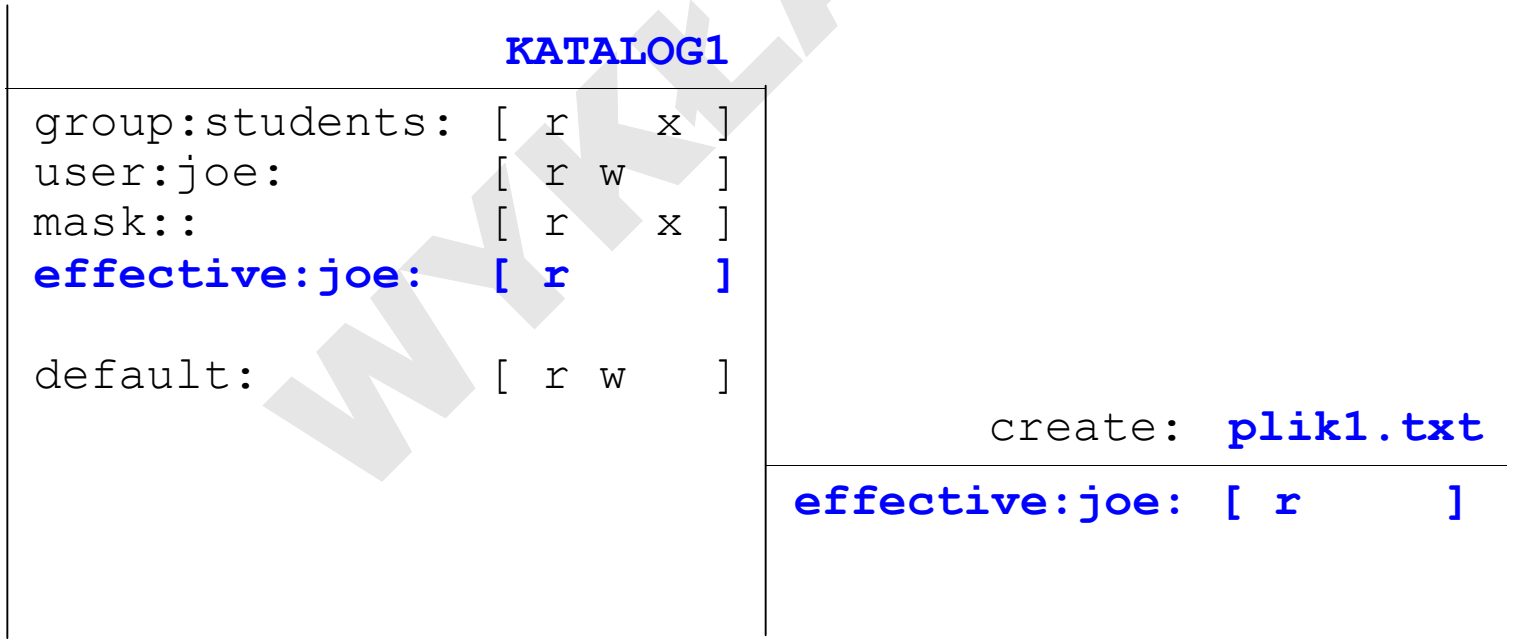
- ACL, CAP, MAC i audyt
- projekt TrustedBSD (http://www.trustedbsd.org/) – dla FreeBSD
- Linux – jądro od 2.5.46 (do jądra 2.4 jest patch ACL, np. SuSE Linux)
- systemy plików: Ext2, Ext3, IBM JFS, ReiserFS oraz SGI XFS
- wsparcie NFS – w NFSv3 możliwe przekazywanie ACL (chociaż brak definicji sposobu na przekazywanie – różne implementacje);
  NFSv4 ma mechanizm NFS ACL (niestety niezgodny z POSIX)
- archiwizacja i backup – narzędzie *pax* (*portable archive interchange*)

Prawa dostępu do zasobów

Prawa dostępu ACL

ACL w jądrze Linux:

- *minimal* ACL – prawa r w x dla u g o
- *extended* ACL – rozszerzone prawa i maski praw:



Prawa dostępu do zasobów

Prawa dostępu ACL

ACL w jądrze Linux:

- polecenia *getfacl* i *setfacl*

```
$ setfacl -d -m group:students:r-x katalog1
$ getfacl --omit-header katalog1
  user::rwx
  user:joe:r-x
  group::r-x
  mask::r-x
  other::---
  default:user::rwx
  default:group::r-x
  default:group:students:r-x
  default:mask::r-x
  default:other::---
```

http://www.trustedbsd.org/docs.html

Prawa dostępu do zasobów

Prawa dostępu ACL

Kolejność aplikacji praw z listy ACL:

1. *owner* (*user*)
2. wyszczególniony użytkownik (filtrowane przez maskę)
3. *owning group* (jeśli zawiera żądane prawa filtrowane przez maskę)
4. wyszczególniona grupa (jeśli zawiera żądane prawa filtrowane przez maskę)
5. jeśli żadna dopasowana grupa nie zawiera praw – żądanie jest odrzucane
6. *others* (bez maski)

Prawa dostępu do zasobów

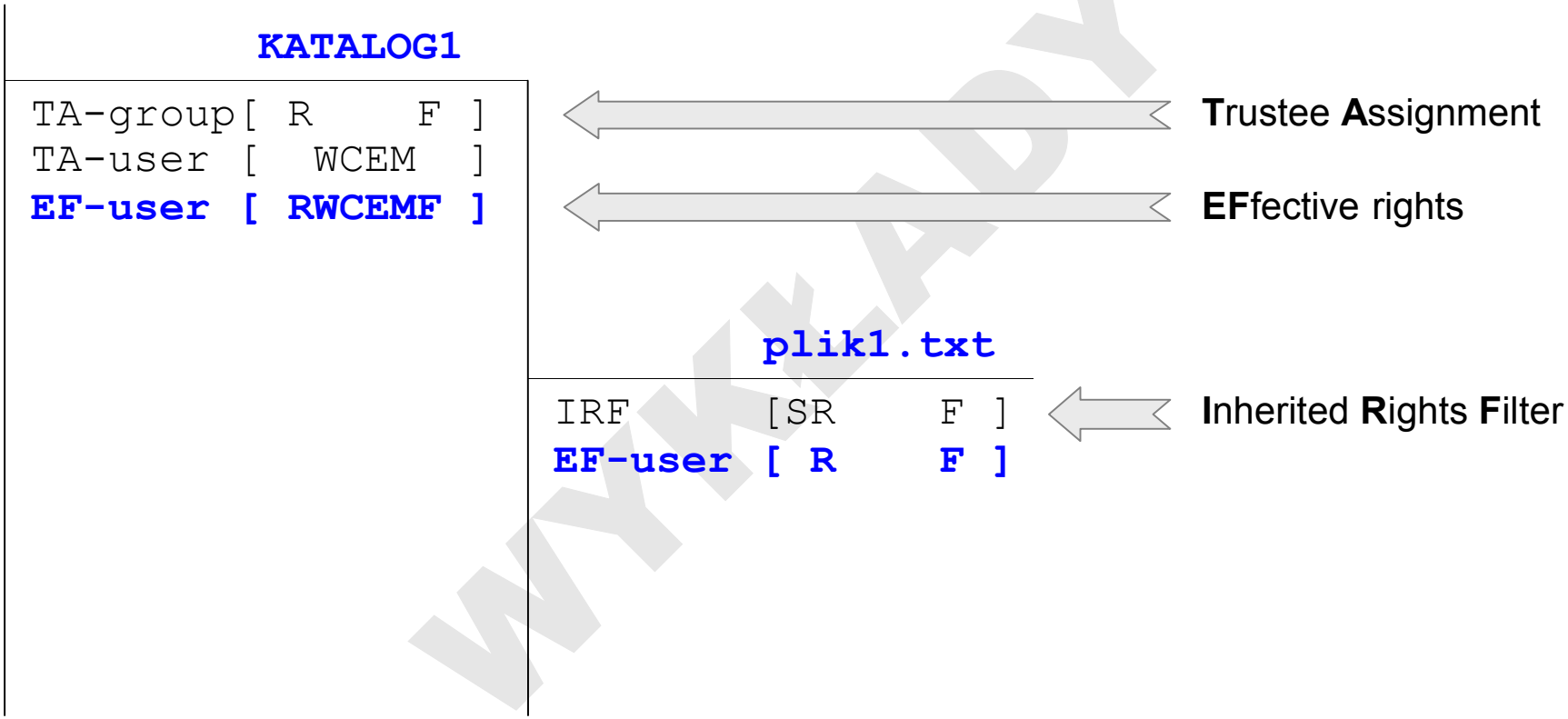
Prawa dostępu Trustees

NetWare (Novell), Linux (http://trustees.sourceforge.net/)

prawo	znaczenie w odniesieniu do **pliku**	znaczenie w odniesieniu do **katalogu**
R (Read)	prawo czytania danych z pliku	
W (Write)	prawo zapisywania danych do pliku	
C (Create)	prawo odzyskiwania pliku usuniętego (o ile nie miał atrybutu P)	prawo tworzenia plików i podkatalogów
E (Erase)	prawo usuwania pliku	prawo usuwania plików i podkatalogów
M (Modify)	prawo do zmiany nazwy i atrybutów pliku	prawo do zmiany nazwy i atrybutów katalogu
F (File scan)	prawo czytania nazwy pliku	prawo do czytania nazw plików w katalogu (umożliwia wylistowanie zawartości katalogu)
A (Access control)	prawo nadawania i odbierania posiadanych praw do pliku innym użytkownikom	prawo nadawania i odbierania posiadanych praw do katalogu innym użytkownikom
S (Supervisory)	prawo unieważniające wszystkie restrykcje w odniesieniu do pliku	prawo unieważniające wszystkie restrykcje w odniesieniu do katalogu

Prawa dostępu do zasobów

Prawa dostępu Trustees



Prawa dostępu do zasobów

Prawa dostępu ACL w MS Windows

- ACL (odbiegające od POSIX) w systemie plików NTFS
- wsparcie protokołu CIFS (*Common Internet File System*)
- oddzielnie przypisania GRANT i DENY
- dynamiczne grupy (np. *All Authenticated Users*)
- wiele atrybutów: append, delete, change permissions, take ownership, change ownership, …

SAMBA wspiera ACL POSIX (!)

istnieją podobne atrybuty dla niektórych systemów plików Unix
(*np.* chattr *dla Linux Ext2/Ext3*)

Prawa dostępu do zasobów

Implementacje ACL w systemie Unix

Przechowywanie list ACL:

- węzeł przysłonięty (*shadow inode*): wiele i-węzłów może być związanych z tym samym węzłem przysłoniętym (Solaris UFS file system)
- rozszerzone atrybuty EA obiektów (*Extended Attributes*): struktury informacyjne związane z plikami i in. obiektami w systemie (Linux)
- ilość pozycji na liście ACL

man 5 attr

File system	Max. entries
XFS	25
Ext2, Ext3	32
ReiserFS, JFS	8191

Prawa dostępu do zasobów

Implementacje ACL w systemie Unix

Efektywność kontroli ACL:

Seconds for Copying Files From Memory to a File System
(11351 files, 608 directories, total file size = 137 MB)

File system	Without EAs or ACLs		With ACLs		Overhead (%)
Ext2	18.3	0.2	18.8	0.2	+3
Ext3	22.0	2.4	22.7	0.5	+3
ReiserFS	9.0	0.1	12.8	0.1	+42
XFS-256	19.0	0.2	34.1	0.2	+80
XFS-512	20.1	0.4	21.4	0.2	+7
JFS	38.2	0.6	36.5	0.2	-4

Uprawnienia specjalne

Unix

Flaga *suid*

- *suid* = *Set User Id*
- takie programy jak *passwd*
- często spotykany atak to wyszukiwanie programów z atrybutem *stuid* należących do superużytkownika *root*, w celu wykorzystania błędów lub nieodpowiedniej konfiguracji systemu i uzyskania uprawnień administracyjnych
- programy z ustawionym bitem *suid* powinny być szczególnie chronione
- dla grupy – flaga *sgid* = *Set Group Id*

Redukcja przywilejów

(*privilege reduction*)

POSIX capabilities (CAP)

- mechanizm pozwalający na rozdzielenie uprawnień administracyjnych superużytkownika *root* na zbiór szczegółowych uprawnień
- powiązanie z systemem plików (dodatkowe bity praw dostępu)
- zmiany czasu systemowego, podwyższania priorytetów procesów, tworzenia i kontroli plików specjalnych, mechanizmów IPC, zarządzania pamięcią, modułami jądra, itp.
- Linux od jądra 2.2 ("requires root privilege" = "requires a capability")
- narzędzie LCAP (http://pweb.netcom.com/~spoon/lcap/)

Redukcja przywilejów

Linux capabilities

ftp://linux.kernel.org/pub/linux/libs/security/linux-privs

m.in.:

- administrowanie modułami jądra
- administrowanie siecią
- dowiązywanie do gniazd numerów portów zarezerwowanych dla serwisów systemowych
- realizacja komunikacji rozgłoszeniowej i grupowej w sieci
- omijanie ograniczeń dotyczących kontroli dostępu do plików
- zmiana informacji o właścicielu i grupie plików
- kontrola plików specjalnego rodzaju
- kontrola flag *suid* oraz *sgid*
- omijanie ograniczeń dotyczących wysyłania sygnałów do procesów
- blokowanie stron w pamięci fizycznej
- omijanie limitów zasobowych

Separacja przywilejów

(*privilege separation*)

- zadania wymagające wysokich uprawnień są odseparowane w postaci odrębnego procesu
- nadaje również się do systemów z klasycznym przydziałem uprawnień administracyjnych (bez CAP)
- ponadto błędy w programie procesu głównego (z uprawnieniami zwykłego użytkownika) nie umożliwiają już tak łatwego uzyskania wysokich uprawnień
- przykład sztandarowy: OpenSSH

Wirusy

Wirusy i inne robactwo

- wirus – kod samopowielający się w systemie operacyjnym
- wirusy skryptowe i w makrodefinicjach (np. w dokumentach edytorów tekstu)
- wirusy sieciowe (*worms*) – przenoszone poprzez sieć (uruchamiane lokalnie)
  - Internet Worm (© Robert Morris)
- wirusy w poczcie elektronicznej (jako załączniki)
  - wirusy atakujące programy pocztowe (np. Win.Redteam):
    - przenoszą się tradycyjnie (przez system operacyjny)
    - jako ofiary wybierają klientów poczty elektronicznej (np. Eudorę)
    - zarażają klienta i wykonują makrodefinicje (skrypty Eudory), rozsyła się do wszystkich osób z listy adresowej.

Wirusy

Wirusy i inne robactwo

- konie trojańskie – „udające” pożyteczne i znane programy (np. nazwą)
- wirusy bujdy (hoax-wirusy) – udające ostrzeżenia:

> Na wolności pojawił się nowy wirus gorszy od wszelkich znanych dotychczas. Przedostaje się wędrując po linii energetycznej, zamienia piny portu szeregowego i odwraca kierunek obrotu dysków. Aby zapobiec zarażeniu nie korzystaj z energii elektrycznej z sieci. Są pogłoski, że wirus zaatakował już prawie wszystkie większe fabryki akumulatorów zarażając dodatnie bieguny baterii i akumulatorów. Można próbować podłączyć tylko biegun ujemny.
>
> Prześlij natychmiast to strzeżenie do wszystkich swoich znajomych.

http://www.hoax-slayer.com/

Wirusy

Wirusy i inne robactwo

trendy

- 2002: wirusy wieloplatformowe: exe + elf
- 2003: *blended threats* – łączą cechy różnych klas: wirusów, koni trojańskich, robaków sieciowych (np. MS Blaster)
- 2004: wirus Ratter atakujący Windows CE
- 2004: wirus Cabir atakujący telefony komórkowe z systemem Symbian

Wirusy

Zaawansowane mechanizmy

kamuflaż i techniki anty-antywirusowe

- polimorfizm
- metamorfizm
- maskowanie (*stealth*) oraz opancerzenie (*armor*)
- retrowirusy (obrona agresywna)

moduły zdalnie aktualizowane

Wirusy

Efekty działania wirusów

- utrudnienie pracy systemu (zużywanie zasobów: CPU, pamięci, przestrzeni dyskowej, pasma sieci)
- część artystyczna: destrukcja danych
- wątek szpiegowski: „wyciek” danych na zewnątrz (zamaskowany kanał komunikacyjny)

Wirusy

Nakłady poniesione na usunięcie infekcji i zwalczanie skutków:

- w roku 1999: w sumie 12,1 mld USD
- w roku 2000: w sumie 17,1 mld USD

Jedne z najkosztowniejszych wirusów w historii:

- Love Bug (łącznie ok. 50 odmian), 2000 r.: 8,7 mld USD
- Code Red, 2001 r.: 2,6 mld USD
- Melissa, 1999 r.: 1,2 mld USD
- Explorer, 1999 r.: 1 mld USD
- SirCom, 2001 r.: 460 mln USD

(*wg Computer Economics, 2001*)

Ochrona przed wirusami

1. Programy wyszukujące programy zarażone (skanery)

- przed ich uruchomieniem
  - wymagające znajomości kodu wirusa
    - wyszukujące identyfikator wirusa (*fingerprint*)
    - wyszukujące fragment charakterystyczny wirusa
  - nie wymagające znajomości kodu wirusa
    - analiza zmian struktury programu
- na podstawie efektów
  - metoda "przynęty"
  - analiza zmian w systemie plików po zakończeniu programu

Ochrona przed wirusami

2. Programy wzbogacające możliwości ochronne systemu

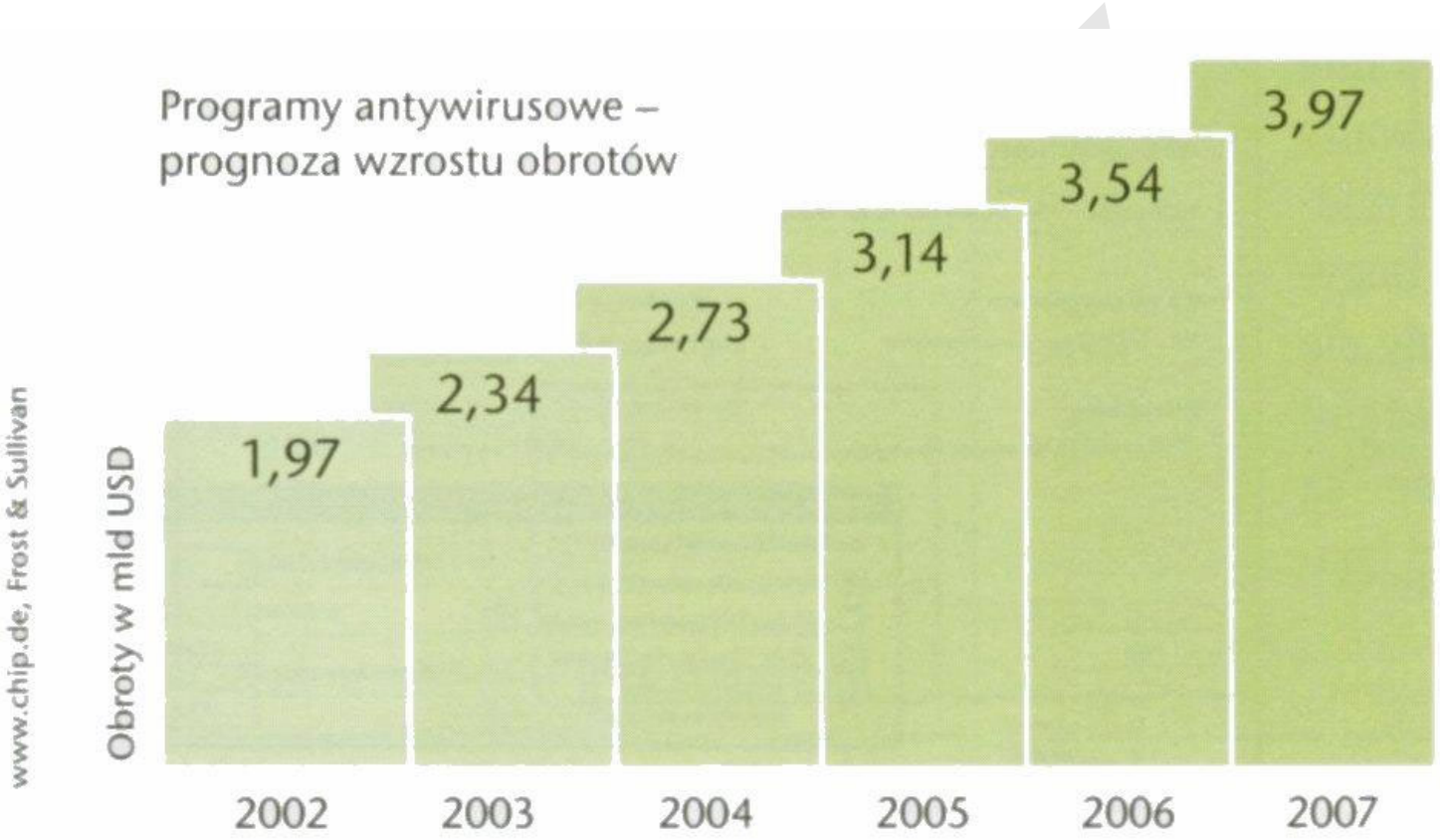
- monitory operacji wykonywanych w systemie wyszukujące działania wirusa
  - w trakcie instalacji lub powielenia wirusa
  - w trakcie wykonywania zadań
    - destruktywnych
    - demonstracyjnych
- skanery usług sieciowych (np. poczty elektronicznej, WWW)
- system plików zabezpieczony przed zapisem

3. Programy usuwające:

- kod wirusa
- skutki działania wirusa

Ochrona przed wirusami

Niezły biznes



Ochrona przed wirusami

Przykładowe narzędzia:

Zdalne skanery internetowe

- http://skaner.mks.com.pl
- http://hausesall.trendmicro.com
- http://www.bitdefender.com
- http://www.pandasoftware.com/activescan/pol/activatescan_principal.htm
- http://www.kaspersky.pl/serwices.html/s=online_vir_chk

Programy anty-spyware

- Ad-aware: http://www.snapfiles.com/get/adaware.html
- Spybot: http://www.safer-networking.org/pl/download/
- Bazooka: http://www.kephyr.com
- SpywareBlaster: http://javacoolsoftware.com/downloads.html

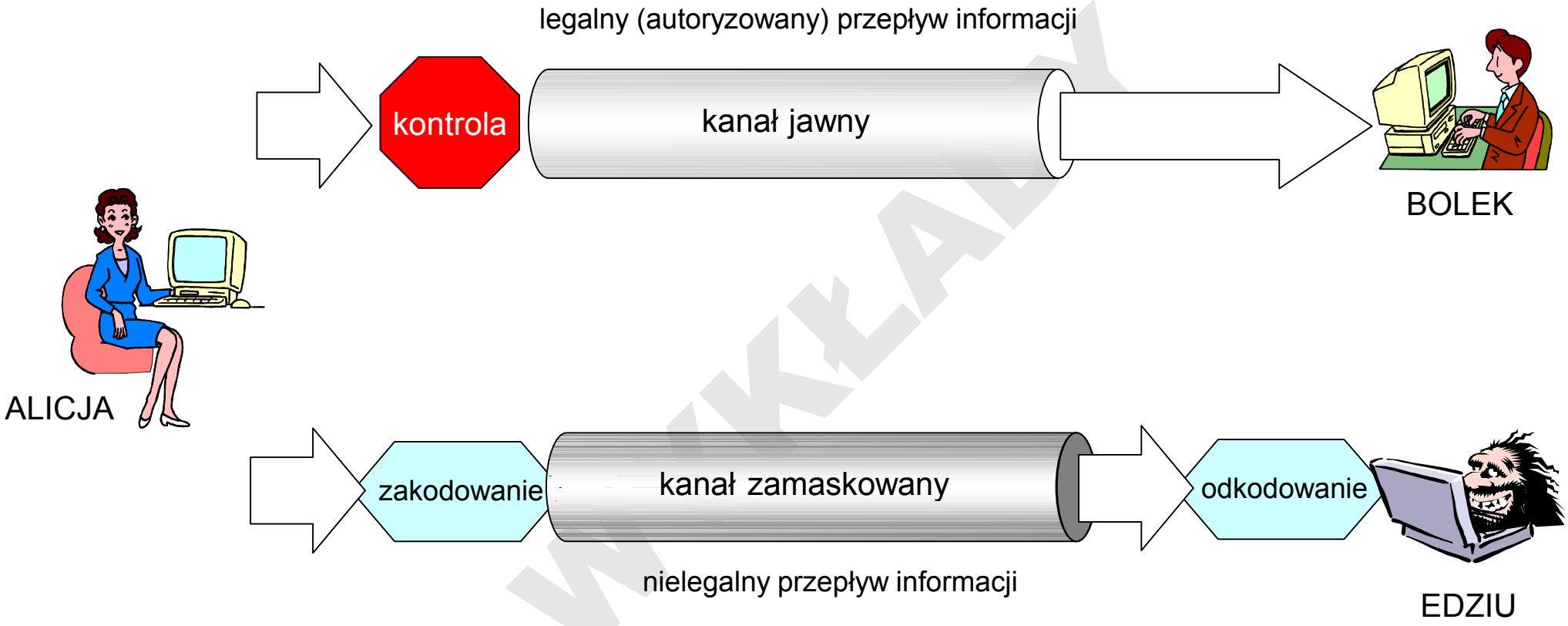
Skuteczność ochrony?

- 41% posiada oprogramowanie antywirusowe, jednak tylko 9% aktualizuje wzorce wirusów częściej niż raz w miesiącu
- 17% użytkowników poczty zawsze (lub automatycznie) otwiera załączniki pocztowe (w tym blisko 30% z czystej ciekawości)
- 61% nigdy nie modyfikowało domyślnych ustawień oprogramowania aplikacyjnego w celu podwyższenia poziomu bezpieczeństwa

(*szacunki wg badań internetowych Central Command, 2001)*

Zamaskowane kanały komunikacji

(*covered channels*)



Zamaskowane kanały komunikacji
(*covered channels*)

Przykłady

- kanał czasowy (obciążenie systemu)
- kanał poprzez kolejkę wydruku
- kanał poprzez tworzenie / usuwanie pliku

Obrona

- poprzez wprowadzanie szumu komunikacyjnego

  1 MB · 1b / 10s = 2 ½ roku

Narzędzia podnoszące poziom bezpieczeństwa

1. **Weryfikatory trywialności haseł (*proactive password checking*)**
   - np. passwd+, anlpasswd, npasswd (Unix), passfilt (Windows)

2. **Łamacze haseł (*reactive password checking*)**
   - np. crack (Unix), L0phtCrack (Windows), john (Unix i Windows)

3. **Generatory nietrywialnych haseł**
   - passwordsafe, PasswordGenerator (http://www.securesafepro.com), Atory Password Generator (http://www.atory.com), AnyPassword (http://www.anypassword.com), KeePass (http://keepass.sourceforge.net)

4. **Kompleksowe systemy testowania konfiguracji i nadzoru**
   - np. COPS (Computer Oracle and Password Checker), SATAN, STAT-NT (Windows)

Hasła jednorazowe – produkty

SecureID (Security Dynamics / RSA Security Inc.)

ACE token = *Access Control Encryption*

- karta generuje kod cyfrowy zmieniany co 60 sek. (funkcja czasu i tajnego numeru karty)
- jako hasło użytkownika występuje konkatenacja kodu i PIN użytkownika
  - SecureID Card – generator i wyświetlacz kodu (żywotność 3 lata)
  - SecureID PINPAD Card – z klawiaturą numeryczną do wprowadzania numeru PIN
  - SecureID Key Fobs – breloczek do kluczy

Hasła jednorazowe – produkty

ACM = ***Access Control Module*** – zastępuje moduł uwierzytelniania (*login*) systemu operacyjnego: SCO Unix, Solaris, HP-UX, Ultrix, IRIX, AIX, VMS, MVS, Cray Unicos, Windows 95 – 2003, Windows Mobille 2003, NetWare Connect, ...

- uwierzytelnianie wg własnego protokołu (tajny algorytm)

Hasła jednorazowe – produkty

Security Dynamics współpracuje z wieloma producentami sprzętu.

ACM mogą być również:

- serwery dostępowe m.in. 3Com, Cisco, Gandalf, Xylogic
- routery m.in. Cisco, Rockwell, Telebit
- palmtopy, telefony komórkowe

ACM hardware – dla innych platform operacyjnych

- serwer dostępowy w następujących modelach:
  - ACM/100 – 1 port, do 100 użytkowników
  - ACM/400 – 4 porty, do 200 użytkowników
  - ACM/1600 – 16 portów, do 400 użytkowników

Hasła jednorazowe – produkty

ACE/Server – centrum zarządzania systemem dla całej sieci

- zarządzanie użytkownikami SecurID i kartami
- zarządzanie dostępem do zasobów
- rejestrowanie zdarzeń
- realizacja: Progres RBDMS (4GL, SQL API for C)

ACE/Slave – serwer rezerwowy

- replika ACE/Servera

Hasła jednorazowe – produkty

Lintel Security, Belgia

- funkcjonalnie zbliżony do SecurID PINPAD Card
- serwer uwierzytelniania na dedykowanym lub niededykowanym NetWare

S/Key (Fill Carn, Bellcore – Bell Communication Research)

- prosty pakiet uwierzytelnienia za pomocą generowanych programowo haseł jednorazowych (z wykorzystaniem MD4) głównie dla środowisk Uniksowych

OPIE (NRL – US Naval Research Laboratory)

- odpowiednik S/Key wykorzystujący MD5
- OPIE = One-time Passwords In Everything

S/Key i OPIE

Programowa generacja haseł

- na początku lat 1980-tych Lamport zaproponował schemat nie wymagający kryptografii
- klient ustala tajne hasło P, z którego generuje się za pomocą funkcji jednokierunkowej h sekwencję t haseł jednorazowych:

  $P,\ h(P),\ h(h(P)),\ ...,\ h^{t-1}(P)$

- hasła wykorzystuje w kolejności od końca: i-te hasło to $h^{t-i}(P)$, $1 \leqslant i \leqslant t$
- jako funkcję jednokierunkową h pakiet S/Key używa MD4, a OPIE – MD5
- parametr P można wzbogacić ziarnem salt : $h^{t-i}(P||\text{salt})$
  - można używać tego samego hasła w różnych kontach z różnym ziarnem
  - nie trzeba zmieniać hasła P po t-krotnym wykorzystaniu go