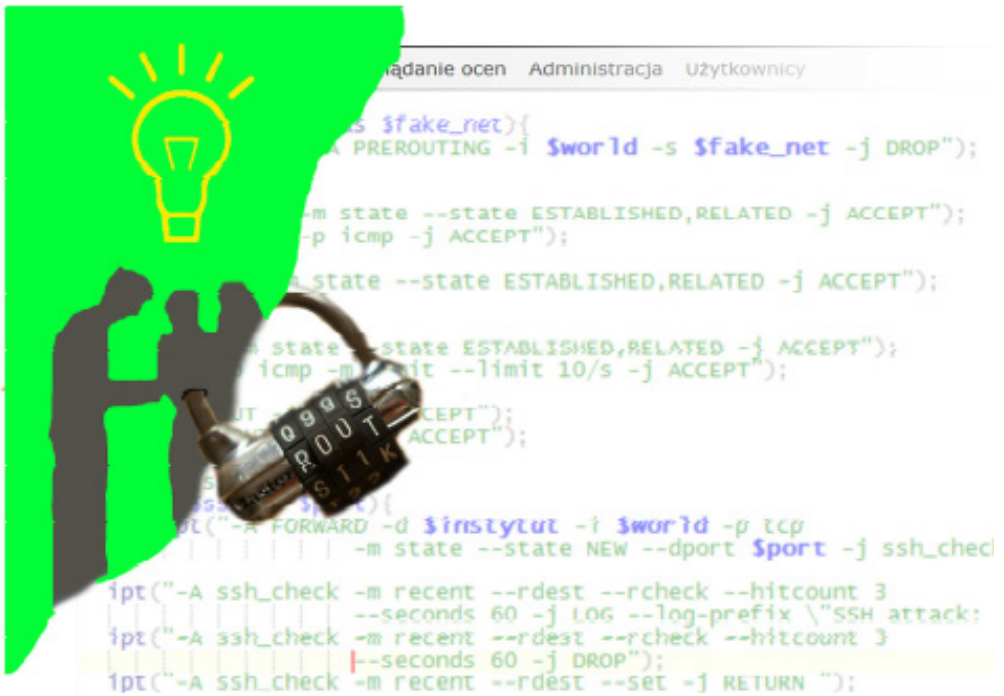
Maciej Laskowski



Współczesne Technologie Informatyczne

Bezpieczeństwo systemów informatycznych

Partnerzy:

WSPÓŁCZESNE TECHNOLOGIE INFORMATYCZNE

BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt Absolwent na miarę czasu współfinansowany przez Unię Europejską
w ramach Europejskiego Funduszu Społecznego

Wydział Elektrotechniki i Informatyki

Politechnika Lubelska
Wydział Elektrotechniki i Informatyki
ul. Nadbystrzycka 38A
20-618 Lublin

WSPÓŁCZESNE TECHNOLOGIE INFORMATYCZNE

BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

Maciej Laskowski

Politechnika Lubelska
Lublin 2013

Recenzenci:
dr inż. Piotr Kopniak, Politechnika Lubelska
dr inż. Grzegorz Kozieł, Politechnika Lubelska

Projekt okładki: Maciej Laskowski

Skład komputerowy: Maciej Laskowski

Publikacja finansowana z projektu „Absolwent na miarę czasu"

Projekt „Absolwent na miarę czasu" współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego. Nr umowy UDA-POKL.04.01.01-00-421/10-01

Publikacja dystrybuowana bezpłatnie

Publikacja wydana za zgodą Rektora Politechniki Lubelskiej

ISBN: 978-83-7947-006-8

Wydawca: Politechnika Lubelska
ul. Nadbystrzycka 38D, 20-618 Lublin
Realizacja: Biblioteka Politechniki Lubelskiej
Ośrodek ds. Wydawnictw i Biblioteki Cyfrowej
ul. Nadbystrzycka 36A, 20-618 Lublin
tel. (81) 538-46-59, email: wydawca@pollub.pl
www.biblioteka.pollub.pl
Druk: TOP Agencja Reklamowa Agnieszka Łuczak
www.agencjatop.pl

Elektroniczna wersja książki dostępna w Bibliotece Cyfrowej PL www.bc.pollub.pl
Nakład: 100 egz.

Spis treści

Wstęp

Systemy informatyczne stanowią obecnie podstawowe narzędzie do operowania na danych. Służą ich gromadzeniu, przetwarzaniu oraz przesyłaniu. Stanowią platformę pracy wielu ludzi, a także sterują urządzeniami, wspomagają pracę człowieka czy nawet zastępują go. Są niezastąpione w wielu dziedzinach, zaś ich awaria uniemożliwia pracę, powoduje wymierne straty finansowe, zaś skutki awarii niekiedy usuwane są przez bardzo długi czas.

Ze względu na istotność zastosowań systemów informatycznych i ich nieodzowność w wielu dziedzinach życia, konieczne jest zapewnienie właściwego poziomu bezpieczeństwa. Przez bezpieczeństwo rozumiemy całokształt zagadnień związanych z szacowaniem i kontrolą ryzyka wynikającego z korzystania z systemów informatycznych i towarzyszącej im infrastruktury.

Zagadnienia bezpieczeństwa systemów informatycznych są jednym z najważniejszych priorytetów ich twórców oraz administratorów. Badania

prowadzone w dziedzinie bezpieczeństwa zaowocowały opracowaniem metod identyfikacji i oceny ryzyka oraz kontroli zagrożeń (Kwiecień, 2012). Pomimo ciągłego rozwoju tej dziedziny nauki oraz pracy wielu zespołów specjalistów z różnych dziedzin, zagadnienia bezpieczeństwa pozostają wciąż nierozwiązanym do końca problemem. Sytuację w tej dziedzinie możemy przedstawić, jako wyścig pomiędzy twórcami zabezpieczeń oraz tymi, którzy próbują je złamać. Wyścig ten skutkuje rozwojem coraz doskonalszych metod zabezpieczeń, które w coraz lepszy sposób chronią systemy informatyczne oraz zawarte w nich dane.

W książce zamieszczono przegląd najważniejszych zagadnień z dziedziny szeroko pojętego bezpieczeństwa systemów informatycznych. Ma ona pomóc czytelnikowi uświadomić sobie, na jakie zagrożenia są narażone systemy informatyczne, w jaki sposób się przed nimi zabezpieczać oraz jak minimalizować skutki naruszenia bezpieczeństwa w momencie, gdy już do niego dojdzie.

Korzystając z okazji, chciałbym przekazać swoje podziękowania recenzentom, dr inż. Grzegorzowi Koziełowi oraz dr inż. Piotrowi Kopniakowi z Zakładu Bezpieczeństwa Informacji w Instytucie Informatyki Politechniki Lubelskiej, których cenne uwagi niewątpliwie przyczyniły się do ostatecznego kształtu tej książki.

Autor

1

Zagrożenia systemów informatycznych

Cel

W niniejszym rozdziale przedstawione zostały czynniki, jakie mogą zagrażać systemom informatycznym. Przeprowadzono również próbę określenia skutków poszczególnych zagrożeń. Celem niniejszego rozdziału jest uświadomienie czytelnikowi występujących zagrożeń oraz ich konsekwencji.

Plan

1. Pojęcie bezpieczeństwa systemu informatycznego
2. Czynniki ryzyka systemów informatycznych
3. Skutki wystąpienia poszczególnych zagrożeń

1.1. BEZPIECZEŃSTWO SYSTEMÓW INFORMATYCZNYCH

Pod ogólnym pojęciem *bezpieczeństwo systemu informatycznego* będziemy rozumieli zapewnienie bezpieczeństwa ochranianemu systemowi informatycznemu oraz zawartym w nim danym. System informatyczny jest bowiem jedynie narzędziem do przetwarzania danych, gdyż to one stanowią najistotniejszy obiekt, który powinien podlegać szczególnej ochronie. Ich utrata, uszkodzenie lub niedostępność często stanowią poważną przeszkodę w dalszym funkcjonowaniu organizacji korzystającej z systemu informatycznego.

Aby móc mówić o bezpieczeństwie konieczne jest zapewnienie bezpieczeństwa zarówno danym jak również systemowi informatycznemu stanowiącemu narzędzie do gromadzenia, przesyłania i przetwarzania danych. Poprzez zapewnienie bezpieczeństwa rozumiemy zapewnienie:

- integralności,
- poufności,
- dostępności.

Integralność to zapewnienie, że dane dostępne poprzez system informatyczny będą zawsze prawdziwe i kompletne, nie będą zawierać błędów oraz będą w jak najlepszy sposób przedstawiały sytuację rzeczywistą, którą powinny opisywać.

Poprzez **poufność** rozumiemy zapewnienie, że dane będą dostępne jedynie dla osób upoważnionych, w takim zakresie, jaki został przewidziany. Osoby postronne i nieupoważnione nie będą mogły uzyskać dostępu do danych.

Dostępność definiujemy jako zapewnienie, że wszystkie upoważnione osoby będą mogły uzyskać dostęp do danych zawsze, gdy zaistnieje

taka potrzeba oraz w każdym miejscu, pod warunkiem, że jest to zgodne z ustalonymi zasadami i procedurami dostępu (Kozieł 2011a).

Osiągnięcie poziomu bezpieczeństwa gwarantującego zachowanie integralności, poufności i dostępności wymaga podjęcia działań zapewniających wyeliminowanie zagrożeń w takich aspektach, jak (Szychowiak 2006):

- awarie sprzętu wchodzącego w skład systemu informatycznego,
- awarie oprogramowania,
- awarie zasilania poszczególnych elementów systemu,
- nieupoważniony odczyt danych,
- nieupoważnione zmodyfikowanie danych,
- zniszczenie danych,
- fabrykowanie danych,
- podsłuchanie transmisji,
- podszywanie się pod innego użytkownika,
- włamanie do systemu informatycznego,
- nieupoważniony dostęp do systemu,
- działanie szkodliwego oprogramowania,
- kradzież danych,
- kradzież elementów sytemu informatycznego,
- udostępnianie danych osobom postronnym przez osoby uprawnione do dostępu do danych,
- naruszenia bezpieczeństwa informacji przez pracowników organizacji.

1.2. AWARIE SPRZĘTU

Awaria sprzętu jest jedną z najczęstszych przyczyn naruszenia bezpieczeństwa systemu informatycznego. Najczęściej narusza dostępność zasobów, ze względu na to, że jakieś urządzenie przestaje działać.

Niekiedy jednak może być przyczyną naruszenia poufności lub integralności danych. Problem ten może wystąpić w przypadku, gdy awarii ulegną urządzenia zabezpieczające dane przed niepowołanym dostępem przy jednoczesnym działaniu urządzeń dostarczających dane. W praktyce taka sytuacja rzadko ma miejsce, ponieważ do rzadkości należą rozwiązania, w których urządzenie dostarczające dane nie posiada żadnych zabezpieczeń (Szychowiak 2006).

Jak wiemy, uniknięcie awarii sprzętu jest niemożliwe. Każde urządzenie psuje się, podlega procesom starzenia, niszczy się w trakcie użytkowania. Nawet w najlepszym sprzęcie, nowym i dokładnie przetestowanym może wystąpić awaria – nie ma możliwości jej wykluczenia. Obrona przed awariami sprzętu polega na budowaniu systemów informatycznych w sposób, który umożliwia zachowanie ciągłości pracy nawet podczas awarii któregoś z elementów.

Idealne byłoby osiągnięcie sytuacji, w której awaria nie zaburza pracy systemu informatycznego. Jednak implementacja rozwiązań pozwalających na zachowanie ciągłości pracy może być bardzo kosztowna.

Przed dobraniem zabezpieczeń należy więc przeprowadzić analizę ryzyka wystąpienia awarii oraz koszty, jakie zostaną poniesione w przypadku jej wystąpienia.

W wielu przypadkach koszty zabezpieczenia ciągłości pracy będą niewspółmiernie wysokie w stosunku do kosztów, jakie zostaną poniesione w przypadku awarii powodującej naruszenie ciągłości pracy systemu. Rodzaj dobranego zabezpieczenia zależy głównie od urządzenia, jakie może ulec awarii.

1.2.1. DYSK TWARDY

Dyski twarde to najczęściej wykorzystywane nośniki informacji. Montowane są w każdym komputerze i to one stanowią podstawę wszelkich magazynów danych. Awaria dysku jest bardzo częstą przyczyną utraty danych. Konieczne jest więc zabezpieczenie się przed tym zagrożeniem.

Pierwszą linią obrony przed utratą danych jest **kopia zapasowa** (często określana angielskim terminem backup), zwana również archiwizacją. Jest to wykonanie kopii danych, przed których utratą chcemy się chronić. Umieszczamy ją na innym nośniku niż ten, który przechowuje „oryginalne" dane. Podczas wykonywania kopii zapasowej najczęściej stosujemy również kompresję w celu zmniejszenia objętości zapisywanych danych. Niekiedy backup jest również szyfrowany w celu ochrony przed niepowołanym dostępem.

Ze względu na dużą objętość kopii zapasowej zostały opracowane jej modyfikacje w celu zmniejszenia objętości zapisywanych danych. Wyróżniamy następujące rodzaje kopii zapasowej (Microsoft 2012):

- Kopia pełna;
- Kopiowane są wszystkie pliki i katalogi ze zbioru, który podlega archiwizacji. Ten rodzaj kopii zużywa najwięcej miejsca. Odtworzenie danych z kopii zapasowej polega na skopiowaniu ich w miejsce uszkodzonych danych oryginalnych.

- Jest to operacja prosta, lecz zazwyczaj dość czasochłonna.
- Kopia przyrostowa;
- W kopii umieszczane są tylko te pliki, które zostały zmodyfikowane od momentu wykonania ostatniej kopii pełnej lub przy-rostowej. Wykonanie kopii przyrostowej wymaga wcześniejszego utworzenia kopii pełnej. Musi bowiem istnieć zbiór danych, z którego będziemy w stanie odtworzyć komplet danych, do którego następnie dołączymy wprowadzone później modyfikacje.
- Wykonanie kopii przyrostowej pozwala na znaczną oszczędność miejsca przeznaczonego na kopię zapasową. wymaga jednak porównania zbioru archiwizowanych danych z poprzednio wykonaną kopią w celu określenia, które zbiory danych uległy zmianie i powinny zostać skopiowane.
- Odtwarzanie danych wymaga wgrania danych z kopii pełnej, a następnie wgrywania danych z kopii przyrostowych w kolejności ich wykonywania – oznacza to więc konieczność posiadania ostatniej kopii pełnej oraz kompletnego zestawu kopii przyrostowych.
- W przypadku, gdy jedna z kopii jest uszkodzona niemożliwe jest odzyskanie ostatecznej poprawnej wersji danych.
- Kopia różnicowa;
- Kopia, w której umieszczane są pliki, które zmieniły się od czasu wykonania ostatniej kopii pełnej lub przyrostowej. Stanowi więc kompromis pomiędzy kopią pełną a przyrostową. Wymaga wcześniejszego wykonania kopii pełnej, po której mogą być wykonane kopie przyrostowe. Wykonując kolejne kopie różnicowe, nie uwzględniamy plików, które zostały umieszczone w poprzednich kopiach różnicowych.
- Kopiujemy je ponownie jako pliki zmodyfikowane.

- Odtwarzanie kopii różnicowej wymaga odtworzenia ostatniej kopii pełnej oraz wykonanych po niej kopii przyrostowych. Następnie odtwarzamy ostatnią kopię różnicową.
- Warto podkreślić, że tylko jedna kopia różnicowa jest odtwarzana, a nie wszystkie wykonane od momentu ostatniej kopii pełnej lub przyrostowej. Kopia różnicowa zawiera więc więcej danych niż kopia przyrostowa, lecz pozwala na zaoszczędzenie dużej ilości czasu podczas odtwarzania danych, ponieważ odtwarzana jest tylko jedna kopia różnicowa po odtworzeniu ostatniej kopii pełnej lub przyrostowej.

Przykład automatycznego wykonywania kopii zapasowej został przedstawiony w rozdziale 3.3.

Kolejnym narzędziem pozwalającym na ochronę przed utratą danych w przypadku awarii dysku jest **obraz dysku**, czyli plik lub zestaw plików zawierający zapisaną całą zawartość dysku twardego, wraz ze strukturą partycji oraz plików i katalogów.

Jest to o tyle istotne, że w momencie, gdy awarii ulegnie np. systemowy dysk twardy, tracimy zainstalowany na nim system operacyjny wraz ze wszystkimi programami, jakie były zainstalowane. Ponowna instalacja systemu, sterowników urządzeń oraz programów zajmuje dużo czasu, zwłaszcza w przypadku intensywnie użytkownych komputerów wyposażonych z bogaty zestaw oprogramowania. O wiele poważniej przedstawia się sytuacja w przypadku serwerów. Niedostępność usług serwera często jest przyczyną powstania poważnych strat. Obraz dysku pozwala na szybkie przywrócenie sprawności serwera, co następnie pozwala na zredukowanie poziomu strat. Przy pomocy utworzonego obrazu dysku możemy odtworzyć jego strukturę na innym dysku twardym.

Obraz dysku oraz procedura jego tworzenia zostały szczegółowo opisane w rozdziale 3.1.

Kopia zapasowa oraz obraz dysku to narzędzia, które wymagają kopiowania dużej ilości danych, a niekiedy również wymagające specjalistycznej wiedzy informatycznej. Wykonanie jednej z tych operacji często przekracza możliwości zwykłego użytkownika. Kopię zapasową danych należy jednak wykonywać również na stacjach roboczych i komputerach domowych, które z reguły nie są wyposażone w specjalistyczne oprogramowanie. Z pomocą zwykłym użytkownikom przychodzi **synchronizacja** katalogów. Jest to narzędzie pozwalające na porównanie zawartości dwóch wskazanych katalogów. Każdy z katalogów umieszczany jest w jednym z sąsiadujących okien. Określa się je mianem stron. Jeżeli plik istnieje tylko w jednym z katalogów mówimy, ze plik istnieje tylko po jednej stronie. W zależności od położenia okna z katalogiem: po stronie prawej lub lewej. Porównanie wykazuje różnice pomiędzy katalogami pokazując takie informacje jak:

- Pliki i katalogi istniejące tylko po jednej stronie – informacja podawana jest z rozróżnieniem na pliki istniejące po prawej oraz po lewej stronie.
- Pliki o tej samej nazwie, lecz różnej zawartości – najczęściej dotyczy to plików, które zostały zmodyfikowane przez użytkownika od czasu ostatniej synchronizacji.
- Pliki identyczne – pliki, które nie uległy zmianie i w obydwu katalogach są identyczne.

Synchronizacja jest prostą operacją pozwalającą na wykonanie kopii danych znajdujących się w jednym katalogu i umieszczenie jej w drugim katalogu. Przy czym możliwe jest wykonanie synchronizacji tak by kopiowane były tylko te pliki i katalogi, które różnią się między sobą. Operacja ta pozwala zarówno na łatwe wykonanie kopii zapasowej, jak również na ujednolicenie zawartości katalogów, jeżeli użytkownik naprzemiennie korzysta ze swojego zbioru danych umieszczonych

na dwóch różnych nośnikach. Opis tworzenia kopii zapasowej został zamieszczony w rozdziale 3.4.

Wszystkie prezentowane dotychczas rozwiązania pozwalały odzyskać utracone dane. Jednak możliwe było odzyskanie ich w wersji, która istniała w momencie wykonania ich kopii. Jako, że niemożliwe jest wykonywanie opisanych operacji na bieżąco, po każdej zmianie wprowadzonej do zbioru danych, zawsze część danych zostanie utracona. Istnieje jednak rozwiązanie pozwalające na tworzenie kopii zapasowej w czasie rzeczywistym. Jest to **macierz dysków**. Macierz dysków to kilka dysków pracujących tak jakby były jednym urządzeniem – wolumin tworzony jest na kilku dyskach jednocześnie. Dyski macierzy podłączane są do tego samego komputera lub do urządzenia macierzy dyskowej. Mają więc wspólne zasilanie i znajdują się w tym samym miejscu. Pociąga to za sobą zagrożenia identyczne z występującymi w przypadku przechowywania kopii zapasowej na innym dysku tego samego komputera, przed którymi należy się odpowiednio zabezpieczyć. Istnieje sześć różnych rodzajów macierzy dyskowych zwanych w skrócie RAID (ang. *Redundant Array of Independent Disks*) oraz ich kombinacje. Rodzaje macierzy nazywamy również poziomami i oznaczamy je kolejnymi dodatnimi liczbami całkowitymi. Z punktu widzenia bezpieczeństwa istotne są tylko niektóre poziomy RAID.

Są to (na podstawie: Microsoft, 2012):

- RAID 1 – macierz dwóch dysków. Każdy z dysków macierzy jest wierną kopią drugiego. Oznacza to, że dane zapisywane są jednocześnie na dwóch dyskach. Na każdym z nich zapisywana jest ich niezależna kopia. Odbywa się to oczywiście kosztem niewielkiego zmniejszenia prędkości zapisu. Kosztem utworzenia macierzy jest również utrata pojemności. Rozmiar macierzy dostępny do wykorzystania będzie

równy pojemności mniejszego z dysków użytych do utworzenia macierzy. Macierz RAID 1 zachowuje sprawność nawet wtedy, gdy jeden z dysków zostanie uszkodzony.

- RAID 5 – macierz tworzona przy użyciu, co najmniej trzech dysków. Ten rodzaj macierzy jest również odporny na uszkodzenie jednego z dysków. Jednak zapis jest tu zorganizowany w zupełnie inny sposób. Zastosowany jest tu zapis paskowy (ang. *stripping*) polegający na tym, że w macierzy składającej się z n dysków dane są zapisywane na n-1 dyskach. Na każdym z nich zostaje zapisany fragment danych. Na ostatnim dysku zapisywane są sumy kontrolne pozwalające na odtworzenie zawartości dowolnego z dysków macierzy. Macierz RAID 5 wymaga większej liczby dysków od macierzy RAID 1. Koszty jej utworzenia będą więc większe. Jednak w macierzach o dużej pojemności rozwiązanie to jest o wiele lepsze. Po pierwsze pozwala na zwiększenie prędkości zapisu i odczytu. Po drugie utrata pojemności macierzy jest relatywnie mniejsza. W macierzy RAID 1 utrata pojemności wynosi 50%. W macierzy RAID 5 utrata ta wynosi 1/n, gdzie n jest liczbą dysków w macierzy i nie może być mniejsze niż 3. W małych macierzach zawierających trzy dyski relatywna utrata miejsca wynosi 33%. Współczynnik ten wygląda o wiele lepiej w dużych macierzach składających się z wielu dysków.
- RAID 6 – macierz tworzona przy użyciu, co najmniej czterech dysków. Jest to zapis paskowy wykorzystujący pojemność dwóch dysków na zapisywanie sum kontrolnych. Macierz RAID 6 odporna jest na jednoczesne uszkodzenie dwóch dysków. Nawet w takiej sytuacji dane nie są tracone. Macierze poziomu szóstego używane są w systemach wymagających największej odporności na uszkodzenia. Koszt tworzenia tego typu macierzy jest największy, ze względu na koniecz-

ność zakupienia minimum 4 dysków. Utrata miejsca w macierzy wynosi 2/n, gdzie n jest liczbą dysków w macierzy i nie może być mniejsze niż 4. Jak widzimy utrata miejsca w małej macierzy złożonej z czterech dysków wynosi 50%. Jest to analogiczna wartość jak w macierzy RAID 1, jednak tu otrzymujemy w zamian wyższy poziom bezpieczeństwa. Ponadto zwiększenie liczby dysków zainstalowanych w macierzy poprawia ten współczynnik.

Macierze RAID zazwyczaj tworzone są poprzez podłączenie dysków do fizycznego kontrolera, który odpowiada za skonfigurowanie macierzy oraz zorganizowanie zapisu i odczytu danych. Rolą użytkownika jest jedynie podłączenie dysków oraz określenie w programie konfiguracyjnym rodzaju tworzonej macierzy. Niekiedy należy jeszcze skonfigurować dodatkowe ustawienia macierzy. Są jednak one łatwo dostępne poprzez interfejs udostępniany przez kontroler, najczęściej podczas uruchamiania komputera.

Możliwe jest również tworzenie macierzy programowych. W tym przypadku dyski podłączane są bezpośrednio do kontrolera dysków komputera. Macierz RAID tworzona jest poprzez odpowiednie zorganizowanie zapisu przez dodatkową usługę działającą w systemie operacyjnym.

Takie rozwiązanie jest popularne w mniej wymagających wydajnościowo macierzach dyskowych. Pozwala bowiem uniknąć kosztów zakupu kontrolera sprzętowego macierzy.

Ograniczeniem tego rozwiązania jest liczba złącz dysków na płycie głównej komputera limitująca liczbę dysków, jakie mogą być włączone do macierzy. Opis tworzenia programowej macierzy RAID został zawarty w rozdziale 3.5.

1.2.2. KOMPUTER

Kolejnym newralgicznym elementem systemu informatycznego jest komputer. Awaria jednego z podzespołów może spowodować przerwę w pracy komputera lub nawet uszkodzenie innych podzespołów. Istotność tej awarii zależy od roli, jaką uszkodzony komputer pełni w systemie informatycznym. Jeżeli jest to jedna ze stacji roboczych to problem zazwyczaj nie jest zbyt istotny, bowiem użytkownik może skorzystać z innego komputera. Warunkiem jest jednak możliwość uzyskania przez niego dostępu do własnego zbioru danych oraz wszystkich niezbędnych aplikacji. Aby spełnić ten warunek należy przedsięwziąć odpowiednie środki bezpieczeństwa. Poszukiwane jest rozwiązanie, które pozwalałoby użytkownikowi uzyskiwać dostęp do danych z dowolnego komputera podłączonego do systemu informatycznego przy zachowaniu bezpieczeństwa danych. Rozwiązanie takie możliwe jest do zrealizowania za pomocą kontrolera domeny obsługującego profile mobilne, pełniącego jednocześnie rolę serwera plików.

O wiele poważniejsza jest awaria serwera. Najczęściej prowadzi ona do niedostępności zasobów i usług, z których korzysta wielu użytkowników jednocześnie. Bardzo często zdarza się, że użytkownicy nie są w stanie wykonywać swojej pracy do momentu usunięcia awarii. W zależności od uszkodzenia efektem awarii może być:

- Naruszenie dostępności – najczęstszy efekt awarii serwera. Dane nie są dostępne, ponieważ serwer nie pracuje.
- Naruszenie integralności – dane mogą zostać uszkodzone w wyniku przerwania operacji zapisu lub nawet utracone, jeżeli awaria dotyczy dysku.

Zabezpieczenie przed awarią serwera może zostać zrealizowane poprzez zbudowanie klastra niezawodnościowego. Jest to zespół niezależnych komputerów pracujących i zachowujących się tak, jakby były jedną maszyną. Klaster jest przezroczysty dla użytkowników, którzy nie widzą jego struktury, co oznacza, że z punktu widzenia użytkownika klaster jest widoczny jako pojedyncza maszyna. Jednak struktura klastra pozwala na przejmowanie funkcji przez dowolną z maszyn w przypadku awarii jednej z nich. Tematyka klastrów została opisana szerzej w rozdziale 2.

1.3. ATAKI INTRUZÓW

Wiele z zagrożeń wymienionych w rozdziale 2.1 spowodowanych jest działaniem intruzów uzyskujących nieupoważniony dostęp do systemu informatycznego. Aby zabezpieczyć się przed tego typu incydentami należy zadbać o wszystkie aspekty bezpieczeństwa systemu informatycznego, takie jak(Szychowiak 2006):

- ochrona przed szkodliwym oprogramowaniem,
- filtrowanie niepożądanego ruchu,
- odpowiednie zabezpieczenie elementów systemu informatycznego przed dostępem fizycznym,
- właściwy poziom dostępu do elementów systemu informatycznego możliwy do uzyskania przez poszczególnych użytkowników,
- aktualność oprogramowania i jego konfiguracji,
- edukacja użytkowników systemu informatycznego w zakresie jego obsługi i bezpieczeństwa.

  O ile zabezpieczenia sprzętowe, fizyczne i programowe najczęściej są realizowane w sposób poprawny i wystarczający to edukacja użytkowników przeważnie jest zaniedbywana. Stanowi to doskonałe pole

do popisu dla włamywaczy. Ataki prowadzone przy pomocy technik socjotechnicznych stanowią większość wszystkich ataków, gdyż przełamanie zabezpieczeń programowych nie jest łatwe. O wiele łatwiej jest wykorzystać nieświadomość użytkownika i przekonać go do przekazania dostępu do systemu informatycznego. Użytkownik może to zrobić zarówno świadomie, jak i nieświadomie.

Przykładem takiego ataku, w którym użytkownik w sposób nieświadomy pozwala atakującemu uzyskać dostęp do systemu jest atak prowadzony przy pomocy telefonu. Atakujący dzwoni do niczego nie podejrzewającego użytkownika systemu informatycznego i przedstawia się jako administrator systemu lub tester oprogramowania. Najczęściej uprzedza użytkownika o konieczności zachowania bezpieczeństwa systemu udzielając mu przy tym wartościowych wskazówek. Tym sposobem zyskuje zaufanie użytkownika, a następnie prosi użytkownika o przeprowadzenie procedury testowej – może to być np. prośba o zmianę hasła na podane przez atakującego. Po zmianie hasła atakujący uzyskuje możliwość zalogowania się do systemu, a niekiedy nawet utworzenia dodatkowego konta dającego mu stały dostęp do systemu. Nawet jeżeli użytkownik zmieni ponownie hasło atakujący zazwyczaj zdąży wykonać zaplanowane działania kradnąc dane lub je uszkadzając czy też niszcząc.

Innym przykładem prób przejęcia dostępu do systemu jest wysyłanie maili z prośbą o podanie loginu i hasła. Często takie wiadomości opatrzone są podpisem administratora systemu oraz informacją o konsekwencjach, jakie grożą użytkownikowi, jeżeli nie udostępni swoich danych. Ten typ ataku stał się w ostatnich czasach plagą w systemach kont poczty elektronicznej. Poniżej przykład takiego maila:

Uwaga: Abonent
Miło nam poinformować, że nasz Admin Center jest zamknięcie wszystkich nieużywanych kont ze względu na zatory w naszym server. To poczty potwierdzić swoje konto aktywne. Również Aktualnie modernizacji naszej bazy danych i e-mail centrum konta, które są wymagane, aby zakończyć swoje dane i wysłać go do nas. Informacje te będą wymagane w celu sprawdzenia konta, aby uniknąć zamknięte.
* Imię i nazwisko:
* Nazwa użytkownika:
* E-mail:
* Hasło:
UWAGA: Jeśli zrobiłeś tego wcześniej, możesz zignorować tę wiadomość.

Dziękujemy za zrozumienie.
Copyright © Admin 2013 Wszelkie prawa

Jak widać na powyższym przykładzie, atakujący grozi usunięciem konta e-mail użytkownika, jeżeli nie dostanie żądanych danych. Charakterystyczną cechą wszystkich informacji dotyczących próby wyłudzenia dostępu jest żądanie podania hasła. Należy uświadomić wszystkich użytkowników systemu informatycznego, że administrator systemu nigdy nie prosi o podanie hasła. Administrator może uzyskać dostęp do wszystkich niezbędnych mu danych przy pomocy swojego własnego konta. Nie będzie więc prosił o dostęp do kont innych użytkowników.

Dosyć jaskrawym przykładem kampanii edukacyjnej jest prowadzona przez banki akcja informacyjna. Konta bankowe dostępne przez stronę internetową stwarzają atakującym dogodną możliwość przeprowadzenia ataku, który w przypadku powodzenia kończy się utratą środków pieniężnych zgromadzonych na koncie właściciela.

Poniżej zamieszczona została treść takiej informacji udostępnianej klientom przez bank Pekao S.A. – stanowi ona zarówno doskonały przykład, jak też źródło wiadomości o właściwym i bezpiecznym użytkowaniu konta internetowego (za: Pekao S.A., 2012):

Przypominamy o zasadach bezpiecznego korzystania z bankowości elektronicznej

Pamiętaj!

- Podczas kontaktów telefonicznych Bank nigdy nie prosi o podanie numeru PIN do Pekao24. Logowanie do usług telefonicznych odbywa się ZAWSZE w serwisach automatycznych.
- Bank nigdy nie prosi o podanie pełnego hasła do serwisu internetowego, aplikacji mobilnej i serwisu mobilnego.
- Bank nigdy nie prosi o wykonywanie w ramach testów przelewów lub innych operacji związanych z Pekao24.
- Bank nigdy nie prosi o podanie nazwy producenta, modelu i numeru telefonu do PekaoSMS podczas logowania do serwisu internetowego, aplikacji mobilnej i serwisu mobilnego.

- Bank nigdy nie wysyła żadnych certyfikatów bezpieczeństwa poprzez wiadomość SMS.
- Nigdy nie używaj do logowania adresu lub linku podesłanego w wiadomości e-mail lub SMS, jeśli nie jesteś pewien jej źródła.
- Zawsze sprawdzaj czy wiadomość SMS z kodem autoryzacyjnym jest zgodna z wykonywaną przez Ciebie operacją.
- Jeśli korzystasz z tokena sprzętowego, pamiętaj aby nigdy nie podawać numeru seryjnego urządzenia, numeru PIN oraz kodów generowanych przez token osobom nieuprawnionym.
- Bank nigdy nie wysyła wiadomości e-mail z prośbą o podanie tych informacji oraz nigdy nie prosi o wprowadzenie ich na stronie do logowania.
- Numery PIN do PekaoTokena, tokena sprzętowego i aplikacji mobilnej powinny różnić się od PIN-u, którego używasz w serwisie telefonicznym i inne niż PIN do telefonu. Wybierz kombinację cyfr trudniejszą do odgadnięcia niż data Twoich urodzin.
- Zachowaj ostrożność i ograniczone zaufanie w stosunku do wiadomości e-mail pochodzących od nieznanych nadawców. Zalecamy nie odpowiadać na takie wiadomości i nie otwierać przesłanych załączników lub linków oraz nie podawać poufnych informacji na stronach przypominających swoim wyglądem strony Banku.

- Nie ufaj nadawcy wiadomości e-mail. Oszuści mają możliwość spreparowania wiadomości tak, by sprawiała wrażenie, że wysłała ją osoba, lub instytucja, której ufasz.
- Regularnie aktualizuj system operacyjny i przeglądarki internetowe zainstalowane na Twoim komputerze i urządzeniach mobilnych.
- Nie instaluj na komputerze i na urządzeniu mobilnym oprogramowania ze źródeł, do których nie masz zaufania. Niektóre aplikacje mogą umożliwić osobom niepowołanym śledzenie danych wpisywanych w przeglądarce, np. numeru klienta, hasła, a także rejestrować działania podejmowane w Internecie.
- Zabezpiecz komputer i urządzenia mobilne profesjonalnym oprogramowaniem antywirusowym!

Informuj niezwłocznie Bank o wszelkich podejrzanych sytuacjach!

1.4. ZAGROŻENIA DANYCH

Mówiąc o bezpieczeństwie systemów informatycznych nie sposób pominąć kwestii bezpieczeństwa danych. Tak naprawdę ochronie podlegają przede wszystkim dane. System informatyczny jest jedynie narzędziem, które służy do przetwarzania, przesyłania i gromadzenia danych. Oczywiście awaria systemu informatycznego powoduje najczęściej naruszenie bezpieczeństwa danych.

Wśród zagrożeń danych możemy wyróżnić kilka podstawowych rodzajów (Szychowiak 2006):

- uszkodzenie nośnika,
- usunięcie danych,
- sformatowanie dysku,
- uniemożliwienie właścicielowi dostępu do danych,
- dostęp innych użytkowników do prywatnych danych, który może skutkować podglądem, modyfikacją, usunięciem lub wykonaniem dowolnej innej nieautoryzowanej operacji,
- kradzież danych (z nośnikiem lub bez),
- naruszenie integralności danych poprzez usunięcie ich, zmodyfikowanie, utworzenie nowych danych,
- podszywanie się pod innego posiadacza danych.

Pierwsze cztery z wymienionych zagrożeń skutkują niedostępnością danych, niezależnie od tego, które z wymienionych zagrożeń wystąpiło. Sposoby zabezpieczania danych przed ich niedostępnością opierają się na tworzeniu kopii zapasowych, stosowaniu macierzy RAID czy też rozwiązań klastrowych. Szczegółowo te rozwiązania zostały omówione w rozdziale 3.

Zagrożenia wymienione w punktach od 5 do 8 wymagają zastosowania zupełnie innych mechanizmów ochrony. Oczywiście istnieje wiele sposobów ograniczenia dostępu do danych, które możemy zastosować w celu uniemożliwienia dostępu osobom niepowołanym. Jednak nie rozwiązują one w pełni problemu. Często bowiem dane dostają się w niepowołane ręce podczas ich przesyłania. A jak wiemy najpopularniejszym kanałem komunikacyjnym jest obecnie Internet. Jest on dostępny dla

wszystkich. Istnieje więc uzasadnione ryzyko, że osoba niepowołana uzyska dostęp do danych przesyłanych przez nas.

Intruz atakujący dane może mieć różne cele. Od zwykłego poznania zawartości danych po ich nieautoryzowaną modyfikację, sfabrykowanie innych danych czy też ich usunięcie.

Zabezpieczenie się przed tak różnorodnymi zagrożeniami wymaga zastosowania technik kryptograficznych lub steganograficznych. Obecnie najszerzej stosowane są techniki kryptograficzne. Stosowane są do:

- szyfrowania danych,
- sprawdzania poprawności i oryginalności danych,
- sprawdzania autentyczności danych oraz tożsamości ich nadawcy.

Szyfrowanie danych to nic innego jak ich przekształcenie do postaci niezrozumiałej dla osób postronnych. Nawet, jeżeli atakującemu uda się przechwycić dane, stoi przed nim jeszcze jeden problem. Jest nim złamanie szyfru. Nie jest to operacja prosta, a już na pewno nie jest to operacja, którą można wykonać w krótkim czasie – zwłaszcza, jeżeli szyfrujący zadbał o zastosowanie współczesnego algorytmu szyfrującego oraz odpowiednio długiego klucza kryptograficznego. Jeżeli natomiast osoba szyfrująca nie zastosowała się do wymogów bezpieczeństwa, kryptoanalityk może w krótkim czasie złamać szyfr i odczytać dane. Często również możliwe jest poznanie klucza kryptograficznego. Może to skutkować użyciem tego klucza przez osoby nieupoważnione do fabrykowania danych.

Zaszyfrowanie danych jest skutecznym zabezpieczeniem przed ujawnieniem danych. Dosyć istotną kwestią jest również czas niezbędny do złamania szyfru. Jeżeli jest on dłuższy niż czas, w którym zaszyfrowane dane posiadają wartość to zabezpieczenie uznajemy za wystarczająco dobre. Dobrym przykładem jest tu rozkaz wojskowy. Zaszyfrowany

rozkaz zawiera informacje o akcjach podejmowanych przez oddziały wojska. Dla wroga wiadomość ta ma wartość dopóki jest aktualna. Informacja o planowanym ataku, czy też pozycja oddziału wojska może stanowić cenną wskazówkę. Odszyfrowanie tej wiadomości po przeprowadzonym ataku czy też zmianie pozycji oddziału wojska nic nie da wrogowi. Wyjątkiem jest tu sytuacja, w której w wyniku kryptoanalizy skompromitowany zostanie klucz kryptograficzny (np. kryptoanalityk pozna jego budowę). W takim przypadku, jeżeli skompromitowany klucz zostanie użyty ponownie, przeciwnik będzie mógł natychmiast odczytać zaszyfrowane dane.

W przypadku zagrożeń, w których atakujący ingeruje w postać danych – modyfikuje je lub fabrykuje, istotne jest potwierdzenie oryginalności danych i ich pochodzenia. Do tego celu stosowane są funkcje skrótu oraz podpisy cyfrowe.

Funkcja skrótu generuje skrót danych, czyli ciąg bitów wyliczony przy pomocy wybranego algorytmu. Osoba odbierająca dane może ponownie obliczyć skrót danych. Jeżeli jest on taki sam jak skrót dostarczony przez nadawcę to będzie to oznaczać, że dane nie zostały zmodyfikowane ani uszkodzone. Innymi słowy: potwierdzać to będzie ich oryginalność.

Podpis cyfrowy to rozwiązanie bazujące na niesymetrycznym algorytmie kryptograficznym. Nadawca podpisuje dane przy pomocy swojego klucza prywatnego, który jest znany jedynie nadawcy. Odbiorca może odszyfrować dane jedynie przy pomocy klucza publicznego nadawcy, który jest powszechnie dostępny. Jeżeli operacja deszyfrowania się uda, oznacza to, że nadawcą jest osoba, której kluczem publicznym udało się odszyfrować dane. W ten sposób możliwe jest potwierdzenie tożsamości nadawcy (Kozieł, Harbarchuk 2005).

2

Klastry komputerowe

Cel

Niniejszy rozdział poświęcony został przedstawieniu klastrów komputerowych, czyli jednemu z narzędzi o największych możliwościach niezawodnościowych oraz wydajnościowych, jak również posiadających duże możliwości skalowania.

Warto więc poznać nieco bliżej stosowane rozwiązania i ich właściwości. Zgodnie z przyjętą zasadą, w książce prezentowane będą rozwiązania powszechnie dostępne, w miarę możliwości darmowe.

Plan

1. Podstawowe pojęcia związane z tworzeniem klastrów.
2. Wykorzystywane technologie.
3. Przełączanie usług.
4. Współdzielenie danych.

2.1. RODZAJE KLASTRÓW KOMPUTEROWYCH

Klaster komputerowy to grupa niezależnych komputerów połączonych ze sobą oraz skonfigurowanych tak, aby mogły wspólnie pracować. Z punktu widzenia użytkownika klaster komputerowy widziany jest jako pojedyncza maszyna. Komputery wchodzące w skład klastra nazywamy węzłami. Najczęściej połączone są one ze sobą za pomocą szybkiej sieci komputerowej. Każdy z węzłów posiada własny system operacyjny. Działanie w klastrze umożliwia dodatkowe oprogramowanie instalowane w systemie operacyjnym węzła. Klastry są budowane w celu poprawy wydajności, niezawodności, dostępności i skalowalności rozwiązań informatycznych. Istotnym czynnikiem jest niższy koszt budowy klastra w porównaniu do pojedynczego komputera o tych samych parametrach wydajnościowych czy niezawodnościowych (Wikipedia 2012).

Klastry mogą być budowane poprzez wykorzystanie tanich komputerów PC, czego przykładem może być klaster Beowulf, którego idea tworzenia opiera się na łączeniu powszechnie dostępnych komputerów PC, często pełniących rolę maszyn roboczych (Wikipedia 2012a). Realizacja klastra Beowulf przedstawiona została na rysunku 2.1.

Profesjonalne rozwiązania bazują na maszynach przeznaczonych do pracy, jako serwery. Najczęściej są to maszyny przystosowane do umieszczenia w szafach serwerowych, co pozwala na umieszczenie dużej liczby maszyn na relatywnie małej powierzchni, przy jednoczesnym zapewnieniu im odpowiednich warunków zasilania i chłodzenia.
Przykładowa realizacja tego typu klastra została zaprezentowana na rysunku 2.2.

Klastry znajdują zastosowanie w wielu dziedzinach. Najczęściej tam gdzie wymagana jest wysoka niezawodność lub duża wydajność. Będą to więc zarówno organizacje, których działalność opiera się na wysokiej dostępności usług lub zasobów w sieci (np. banki, firmy ubezpieczeniowe, profesjonalne firmy hostingowe), jak i jednostki potrzebujące dużej mocy obliczeniowej, takie jak centra badawcze lub uczelnie.

Rys 2.1 Klaster Beowulf
(źródło: Wikipedia, 2012a)

Ze względu na zadania, jakie ma pełnić klaster należy odpowiednio dobrać jego typ. Podział klastrów ze względu na funkcję, jaką pełnią obejmuje trzy grupy:

- Klastry wydajnościowe – tak zwane klastry przetwarzania równoległego. Ich zadaniem jest osiągnięcie wysokiej mocy obliczeniowej, umożliwiającej wykonywanie wymagających obliczeń lub przetwarzania danych wymagającego dużej mocy procesora. Zadania uruchamiane na klastrach muszą być zaprojektowane w specjalny sposób, umożliwiający podział zadania na mniejsze części możliwe do równoległego, niezależnego wykonania na różnych procesorach.

Rys 2.2 Klaster komputerowy
(źródło: Krutul, 2003)

- Klastry niezawodnościowe (o wysokiej dostępności) – klastry, których zadaniem jest zapewnienie jak największego poziomu niezawodności działania usług lub dostępu do zasobów.
- Najczęściej realizowane są poprzez połączenie maszyn, które wzajemnie przejmują swoje role w przypadku awarii jednego z komputerów wchodzących w skład klastra czy też usługi zainstalowanej na jednym z komputerów.
- Klastry równoważące obciążenie – zwane też często klastrami serwerowymi. Stosowane są do obsługiwania usług o dużym obciążeniu. Mogą to być zarówno usługi sieciowe, jak i obliczeniowe. Ich rola polega na równoważeniu obciążenia maszyn wchodzących w skład klastra. W praktyce oznacza to, że nowe żądanie zostanie skierowane do maszyny o najmniejszym obciążeniu w celu optymalizacji szybkości odpowiedzi na nie.

W praktyce najczęściej konstruowane są klastry o charakterze mieszanym. Kładzie się w nich nacisk zarówno na wydajność, jak i niezawodność działania (Wikipedia, 2012).

2.2. NIEZAWODNOŚĆ KLASTRA KOMPUTEROWEGO

Na niezawodność klastra komputerowego, która jest szczególnie interesująca z punktu widzenia bezpieczeństwa, ma wpływ ciągłość działania na nim usług oraz ochrona danych przed ich utratą lub uszkodzeniem.

Ciągłość działania usług realizowana jest przez przełączanie działania usługi z uszkodzonego węzła klastra do sprawnego. Jednak, aby było to możliwe, usługa uruchomiona na dowolnym węźle musi mieć dostęp do aktualnego zbioru danych. W przypadku wielu usług zbiór danych zmienia się dynamicznie podczas ich działania. Utworzenie jednego

wspólnego zbioru danych, do którego będą miały dostęp wszystkie węzły klastra oczywiście jest możliwe i jest to jedno z rozwiązań. Jednak stosując je tracimy na niezawodności systemu, gdyż uszkodzenie urządzenia przechowującego dane lub brak jego dostępności powoduje awarię całego systemu i powiązanych z nim usług korzystających z danych.
Konieczne jest więc przechowywanie pełnego zbioru danych na wszystkich węzłach klastra, przy czym zbiór ten musi być aktualizowany w czasie rzeczywistym, tak aby wszystkie węzły miały aktualne dane. Rozwiązaniem pozwalającym na zrealizowanie tego założenia jest DRBD, czyli Distributed Replicated Block Device (Linbit, 2012).

DRBD jest rozwiązaniem technicznym pozwalającym na replikację danych (również w czasie rzeczywistym) pomiędzy dwoma dowolnymi punktami na świecie. DRBD jest używane, jako technologia zapewniająca wysoką dostępność oraz pozwalająca utrzymanie ciągłości działania po awarii jednego z węzłów klastra. Technologia ta jest rozwijana przez firmę Linbit na licencji GPL (Linbit, 2012).

DRBD jest rozwinięciem technologii RAID na poziomie węzłów klastra komputerowego. Realizuje mirroring węzłów przy użyciu sieci komputerowej. Zadaniem DRBD jest utrzymywanie na wszystkich węzłach klastra jednakowych zbiorów danych. Pozwala to na jednoczesne korzystanie ze wszystkich węzłów klastra lub też na korzystanie z dowolnego z węzłów po awarii węzła podstawowego. W każdej z tych sytuacji możliwa będzie poprawna praca z aktualnym zbiorem danych, dzięki uaktualnianiu danych w czasie rzeczywistym przez DRBD. DRBD możemy rozumieć jako RAID 1 działający na węzłach klastra przy wykorzystaniu sieci jako kanału transmisji danych. Istotnym czynnikiem jest również to, że DRBD działa na dowolnej liczbie węzłów klastra, nie tylko na dwóch. Organizację zasobów DRBD oraz schemat działania przedstawiono

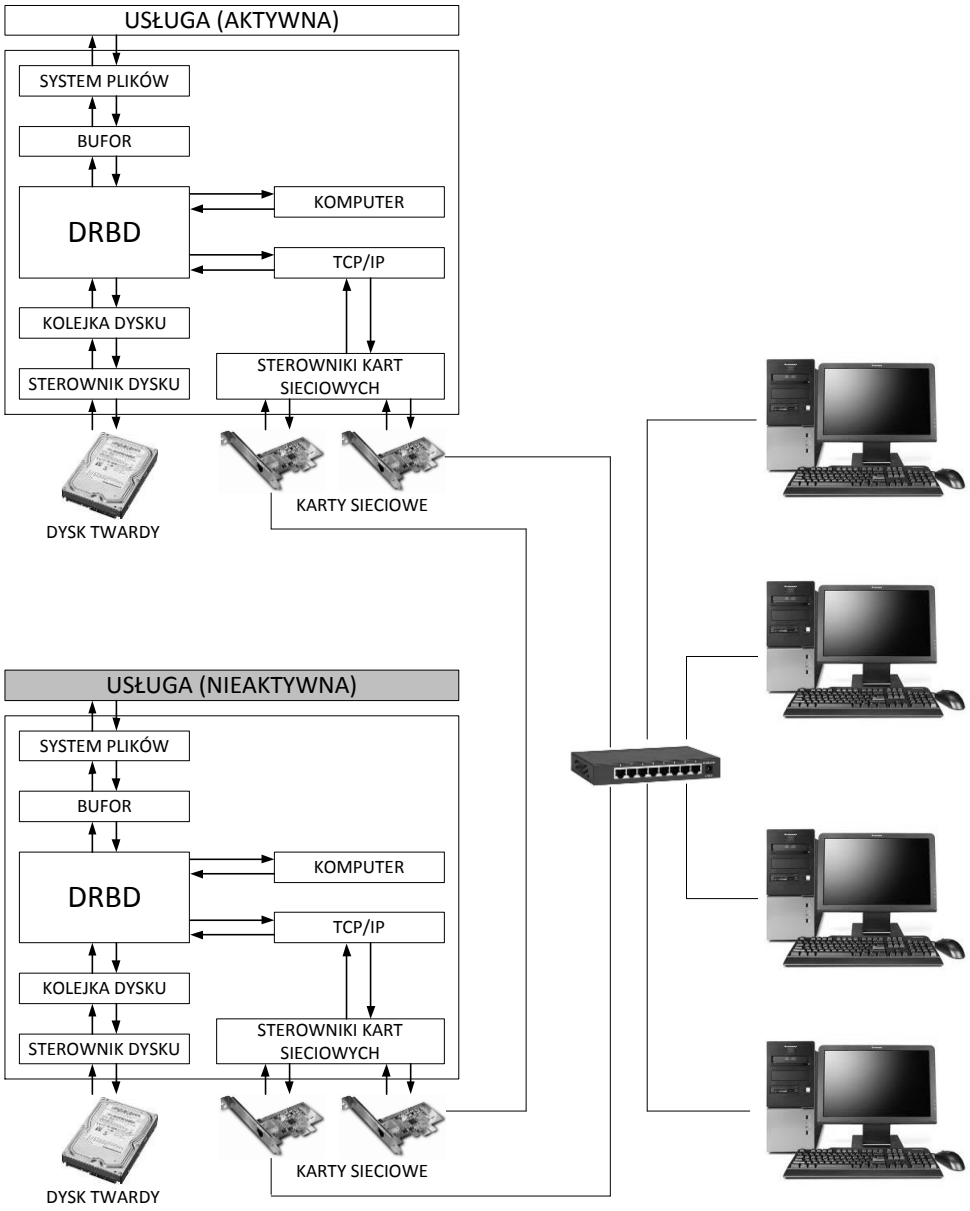
na rysunku 2.3. Usługa DRBD musi być zainstalowana na wszystkich węzłach klastra. Instancje usługi komunikują się pomiędzy sobą poprzez sieć komputerową. Warto zauważyć, że najczęściej budowana jest niezależna sieć komputerowa wykorzystywana jedynie przez usługę DRBD. Dzięki temu możliwe jest utrzymanie wysokiej wydajności systemu oraz uniknięcie opóźnień synchronizacji węzłów spowodowanych dużym ruchem w sieci (DRBD, 2012).

Usługa DRBD stanowi warstwę pośredniczącą pomiędzy podzespołami komputera, kolejką dysku oraz usługami działającymi na danym węźle klastra. Jest nadrzędna w stosunku do urządzeń blokowych, czyli dysków, partycji, woluminów LVM czy macierzy RAID. Dane zapisywane w jednym z węzłów są wiernie przenoszone na pozostałe węzły klastra pracujące pod kontrolą DRBD (DRBD, 2012).

Usługa DRBD oferuje dwa tryby pracy (na podstawie: DRBD, 2012):

- Synchroniczny – system plików w aktywnym węźle klastra (maszynie z uruchomionymi usługami) otrzymuje potwierdzenie zakończenia zapisu bloku danych dopiero wtedy, kiedy zostanie on zapisany na wszystkich węzłach klastra. Tryb synchroniczny przeznaczony jest do pracy w klastrach o wysokiej dostępności, gdzie istotne jest zapisanie każdego pojedynczego bloku danych w przypadku awarii aktywnego węzła. W praktyce ten tryb pracy dobrze się sprawdza w klastrach wyposażonych w dodatkową sieć, o wysokiej przepustowości, dedykowaną do pracy z usługą DRBD. Istotne jest, aby węzły klastra znajdowały się w niewielkiej odległości od siebie, pozwalającej na połączenie ich siecią lokalną.

- Asynchroniczny – węzeł klastra zapisujący dane otrzymuje potwierdzenie zapisania ich na dysku natychmiast po zapisaniu ich na dysku lokalnym. Ten tryb pracy jest niezbędny do tworzenia klastrów, w których węzły znajdują się w dużej odległości od siebie, często ze sobą połączone jedynie siecią Internet. Wykorzystywany jest również wszędzie tam gdzie opóźnienie związane z zapisaniem danych przez inne węzły wprowadzane przez tryb synchroniczny nie jest akceptowalne.
  W tym trybie usługa DRBD synchronizuje dane pomiędzy węzłami już po zapisaniu ich w aktywnym węźle. Pozwala to na sprawne działanie pomimo opóźnień w transferze danych pomiędzy węzłami, jednak może skutkować niewielką utratą danych w przypadku awarii aktywnego węzła.
  Sama synchronizacja danych nie jest wystarczająca do utworzenia klastra niezawodnościowego. Dodatkowym warunkiem oprócz zabezpieczenia przed utratą i niedostępnością danych jest utrzymanie ciągłości działania usług. Konieczne jest monitorowanie ich działania w węźle aktywnym. Odnosi się to zarówno do kontrolowania czy usługa działa, jak również do sprawdzania poprawności działania usługi. W momencie, gdy usługa przestanie działać lub też ulegnie uszkodzeniu stanowiącemu przeszkodę w poprawnym działaniu, usługa powinna zostać przełączona na inny węzeł klastra. Oznacza to, że w innym węźle klastra ta usługa powinna zostać włączona, w taki sposób by kontynuowała działanie usługi z węzła, który uległ awarii.



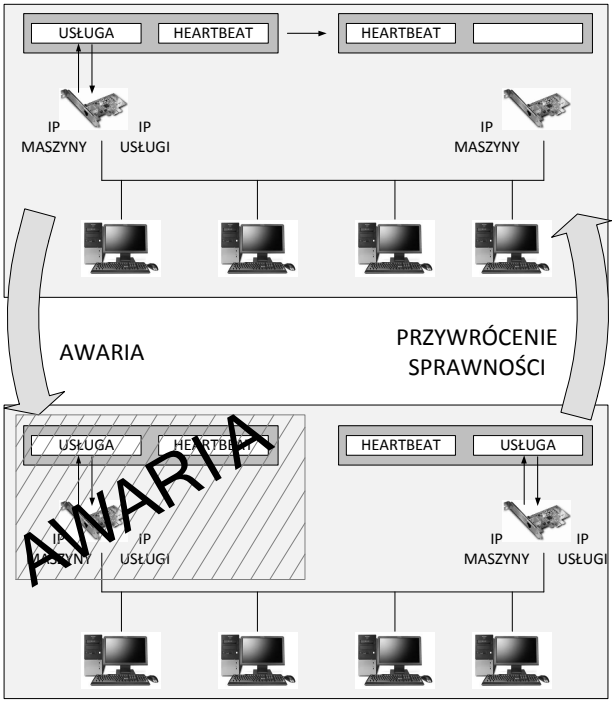
Rysunek 2.3. Schemat działania DRBD w klastrze złożonym z dwóch komputerów (opracowanie własne na podstawie DRBD 2012)

Aby było to możliwe konieczne jest bieżące monitorowanie usług. Możliwe jest wykonanie tego przy pomocy samodzielnie opracowanych programów i skryptów, jednak trudno jest w ten sposób osiągnąć wysoką sprawność działania oraz krótkie czasy przełączania usług. Z pomocą przychodzi tu usługa Heartbeat, której schemat działania został zaprezentowany na rysunku 2.4. Jej zadaniem jest monitorowanie pracy usług oraz utrzymywanie komunikacji pomiędzy węzłami klastra (DRBD, 2012).

W momencie wystąpienia awarii usługi lub węzła, Heartbeat wykrywa zaistniałą sytuację i informuje o niej węzeł, który ma zastąpić uszkodzoną maszynę. W przypadku, jeżeli uszkodzeniu uległa jedynie usługa, zostaje ona wyłączona w węźle aktywnym. W węźle, który ma zastąpić węzeł aktywny uruchamiana jest kopia usługi, która przestała działać w węźle aktywnym. Aby możliwe było kierowanie żądań do usługi modyfikowany jest również adres IP usługi, tak by wskazywał na węzeł, na którym ona aktualnie działa. Po usunięciu awarii i przywróceniu poprawnego działania węzła oraz zainstalowanych usług Heartbeat ponownie przenosi działanie usługi do węzła podstawowego. Dezaktywuje jednocześnie działającą usługę w węźle zapasowym oraz dokonuje ustawienia adresu IP usługi tak, aby wskazywał na maszynę, na której usługa ta jest w danej chwili uruchomiona (DRBD, 2012).

Górna część rysunku 2.4 przedstawia poprawnie działający klaster. Węzeł aktywny stanowi platformę działania usługi. W momencie wystąpienia awarii (dolna część rysunku) Heartbeat uruchamia usługę w węźle zapasowym, ustawiając jednocześnie jej adres IP tak, aby wskazywał na węzeł zapasowy. Usunięcie usterki skutkuje ponownym przeniesieniem usługi do węzła podstawowego. Przeniesienie może być zainicjowane zarówno automatycznie przez usługę Heartbeat, jak również przez administratora.

Oczywiście przed przeniesieniem usługi konieczne jest wcześniejsze zsynchronizowanie danych przechowywanych przez poszczególne węzły klastra. Dokonuje tego DRBD. Po usunięciu awarii automatycznie przystępuje do synchronizacji danych uaktualniając dane czasowo niedostępnego węzła do ich najnowszej wersji. Operacja przeprowadzana jest w tle, bez zakłócania działania usług. W przypadku, gdy awarii uległy wszystkie węzły (np. wskutek rozległej awarii zasilania) DRBD wykryje, który z węzłów działał najdłużej i uaktualni dane pozostałych węzłów do wersji przechowywanej na nim (DRBD, 2012).



Rysunek 2.4 Schemat działania usługo Heartbeat
(opracowanie własne na podstawie DRBD 2012)

Przerwa w działaniu sieci wykorzystywanej podczas synchronizacji również może spowodować utratę spójności danych przechowywanych w różnych węzłach klastra. Nie stanowi to jednak przeszkody w działaniu. Po przywróceniu sprawności sieci DRBD przeprowadza automatyczną synchronizację danych poszczególnych węzłów.

DRBD zapewnia również odporność na awarię lokalnego magazynu danych (dysku, macierzy dysków). Pomimo awarii dane są na bieżąco dostępne. Możliwe jest to poprzez wykorzystanie węzłów zapasowych, jako źródła danych. Po ponownym podłączeniu urządzenia dane przechowywane na nim są synchronizowane z danymi klastra. Jeżeli urządzenie zostało wymienione na inne, możliwe jest jego podłączenie bez zakłócania pracy klastra, a następnie zsynchronizowanie go.

DRBD obsługuje również sytuacje, w których węzły klastra tracą połączenie pomiędzy sobą. Sytuacja taka możliwa jest jedynie w przypadku, gdy wszystkie łącza pomiędzy węzłami zostaną uszkodzone lub przestaną działać. W takim wypadku oddzielone od siebie węzły przechodzą do roli węzłów aktywnych i obsługują przychodzące do nich żądania. Jest to najbardziej kłopotliwa sytuacja, istnieje bowiem możliwość niezależnego zmodyfikowania zbiorów danych w różnych węzłach klastra. Na każdym z nich możliwe jest zmodyfikowanie tego samego zbioru danych w inny sposób, co w rezultacie uniemożliwi poprawną synchronizację danych po ponownym połączeniu węzłów klastra (DRBD, 2012).

3

Bezpieczeństwo danych – receptury

Cel

W niniejszym rozdziale przedstawione zostały praktyczne przykłady wykonywania różnego rodzaju czynności prowadzących do podniesienia poziomu bezpieczeństwa systemów informatycznych.

W opisanych w niniejszym rozdziale rozwiązaniach autor starał się bazować przede wszystkim na oprogramowaniu darmowym, dostępnym powszechnie w Internecie. Ma to na celu promowanie używania legalnego oprogramowania oraz przedstawienie rozwiązań alternatywnych dla oprogramowania komercyjnego.

Plan

1. Przedstawienie zagadnienia tworzenia obrazu dysku oraz jego zastosowań.
2. Tworzenie obrazu sektora rozruchowego.
3. Zastosowania kopii zapasowej i techniki jej wykonania.
4. Synchronizacja katalogów.
5. Macierze dysków – rodzaje i zastosowania.

3.1. OBRAZ DYSKU

Obraz dysku to plik (lub zestaw plików), w którym zapisana jest struktura dysku twardego komputera. Obejmuje ona w szczególności:

- opis struktury dysku,
- zapis wszystkich partycji istniejących na dysku,
- zapis wszystkich danych istniejących na dysku.

Przy pomocy obrazu dysku możliwe jest odtworzenie jego struktury na dowolnym dysku twardym o wystarczającej pojemności (najczęściej jest to dysk o pojemności nie mniejszej niż dysk, z którego tworzony był obraz). Struktura odtwarzana jest w taki sposób, że podmienienie dysków w komputerze nie powoduje żadnych zmian. System operacyjny uruchomi się prawidłowo, wszystkie aplikacje będą zainstalowane w systemie, a dane będą na swoim miejscu tak, jak to miało miejsce na dysku oryginalnym.

Opisując obraz dysku należy wspomnieć również o **obrazie partycji**. Jest to bowiem obraz jednej z partycji znajdującej się na dysku twardym. Obraz partycji zawiera jedynie zapis struktury i danych znajdujących się na wybranej partycji. Wiąże się z tym poważne ograniczenie. Aby możliwe było odtworzenie obrazu partycji, partycja musi istnieć na dysku twardym. Jeżeli więc utraciliśmy tablicę partycji, konieczne jest ponowne ręczne utworzenie partycji na dysku, a dopiero później możliwe będzie odtworzenie jej obrazu. Należy jednak mieć świadomość, że jeżeli jest to partycja zawierająca system operacyjny najprawdopodobniej nie będzie się on uruchamiał po takim odtworzeniu obrazu partycji. Najważniejszą przyczyną tego stanu rzeczy jest brak boot managera w master boot record (MBR). Odpowiada on bowiem za inicjalizację procesu urucha-

miania systemu operacyjnego. Kolejną przeszkodą może być niewłaściwe położenie partycji uniemożliwiające boot managerowi odnalezienie partycji systemowej.

Ważnym ograniczeniem podczas tworzenia obrazu (zarówno partycji, jak i dysku) jest konieczność posiadania miejsca na zapis obrazu. Miejsce to musi posiadać odpowiednią pojemność dostępną do zapisu oraz znajdować się poza strukturą, której obraz wykonujemy. Wykonując obraz dysku nie możemy go zapisać na tym samym dysku. Konieczne jest więc posiadanie innego dysku twardego, pamięci flash, dysku sieciowego, nagrywarki płyt czy innego nośnika pamięci, który może być obsłużony przez program wykonujący obraz dysku. W przypadku wykonywania obrazu partycji sytuacja jest mniej skomplikowana. Obraz możemy po prostu zapisać na innej partycji. Jeżeli nie dysponujemy dodatkowym nośnikiem pamięci, zaś chcemy wykonać obraz dysku, to musimy wykonać go w częściach. Wykonujemy obrazy wszystkich partycji, a następnie tworzymy obraz master boot recoru. Sposób tworzenia obrazu master boot record został opisany w kolejnym podrozdziale.

Do tworzenia obrazów dysków lub partycji możemy użyć takich programów jak:

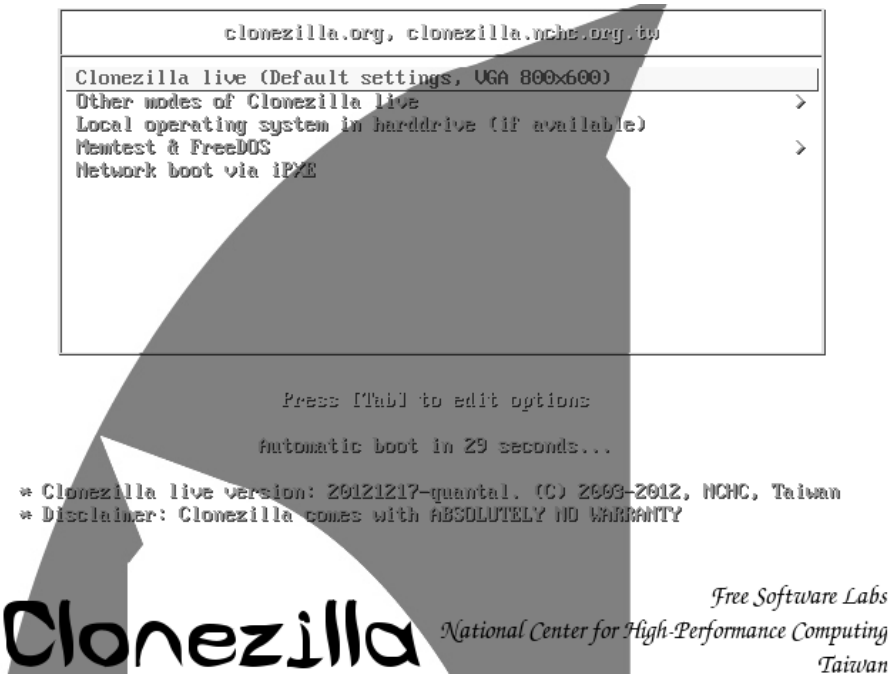
- Clonezilla,
- Partition Image,
- Norton Ghost (komercyjny),
- Acronis True Image (komercyjny),
- poleceń `dump` oraz `dd` systemu Linux.

Darmowym programem pozwalającym tworzyć obrazy dysków i partycji jest Clonezilla. Jest to program działający w środowisku Linux. Korzystamy z niego uruchamiając live CD z załączonym programem Clonezilla. Obraz ISO płyty z programem można pobrać ze strony producent. Wybieramy link pozwalający pobrać najnowszą wersję stabilną (ang. *stable*). Fragment strony z podkreślonym linkiem pozwalającym na pobranie live CD zawierającego program Clonezilla przedstawiony został na rysunku 3.1.

Pobrany obraz nagrywamy na płytę CD. Uruchamiamy komputer startując system operacyjny z płyty CD. Zostaniemy powitani ekranem startowym przedstawionym na rysunku 3.2, umożliwiającym wybór trybu uruchamiania live CD. Domyślna opcja „Clonezilla live" pozwala na uruchomienie systemu operacyjnego zapisanego na płycie CD oraz korzystanie z oprogramowania do tworzenia obrazów dysków. Poniżej znajdziemy opcję „Other modes of Clonezilla Live" Pozwala ona na przejście do kolejnego menu, w którym mamy możliwość wybrania innych rozdzielczości ekranu używanych podczas pracy z live CD i programem Clonezilla. Opcji tej należy używać wówczas, gdy nie odpowiada nam rozdzielczość ekranu oraz wtedy, gdy system operacyjny lub program Clonezilla nie wyświetlają prawidłowo ekranów. Pozostałe opcje pozwalają na uruchomienie systemu operacyjnego z dysku twardego, uruchomienie „memtest" lub uruchomienie komputera z użyciem iPXE.

Rys 3.1. Strona pobierania programu Clonezilla

(źródło: opracowanie własne)



Rys 3.2. Ekran powitalny live CD Clonezilla

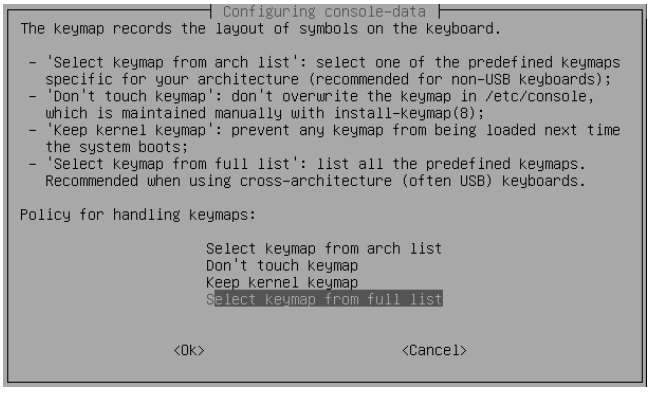
(źródło: opracowanie własne)

Kolejny ekran, który ujrzymy po uruchomieniu systemu operacyjnego z liveCD (rysunek 3.3) pozwoli na wybór języka. Program nie obsługuje języka polskiego.

Rys 3.3. Menu wyboru języka programu

(źródło: opracowanie własne)

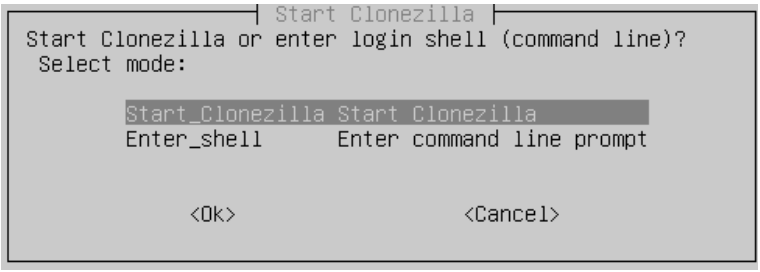
Po wybraniu języka zostaniemy poproszeni o wybranie układu klawiatury (rysunek 3.4). Domyślny układ klawiatury nie wymaga zmiany, jeżeli używamy znaków z alfabetu języka angielskiego.



Rys 3.4. Menu wyboru układu klawiatury

(źródło: opracowanie własne)

Na następnym ekranie (rysunek 3.5) zostaniemy poproszeni o wybranie czy chcemy uruchomić program Clonezilla czy też pracować w powłoce systemu Linux. Jako, że naszym celem jest tworzenie obrazu dysku, wybieramy opcję pierwszą.



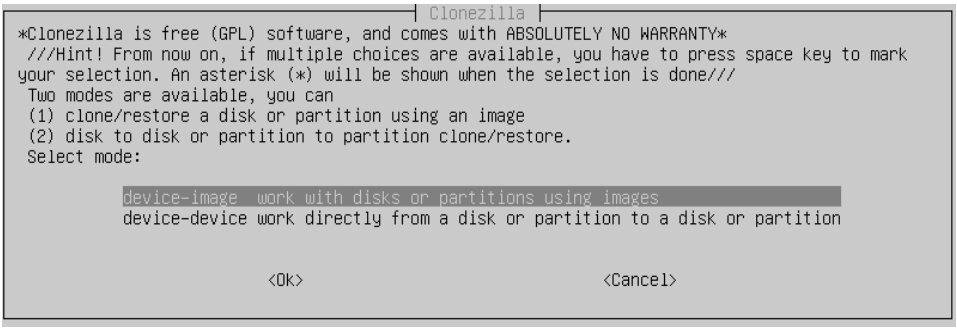
Rys 3.5. Menu wyboru uruchamiania

(źródło: opracowanie własne)

Kolejnym pytaniem, jakie zostanie postawione przed odtwarzającym system jest określenie trybu pracy programu (rysunek 3.6). Mamy do wyboru dwa tryby:

- device-image – tryb pracy, w którym tworzony jest obraz dysku zapisywany w pliku (podczas odtwarzania obraz dysku jest źródłem danych, które zostaną zapisane na dysku),
- device-device – ten tryb pracy umożliwia skopiowanie zawartości jednego dysku na drugi (jednej partycji na drugą) bez zapisywania danych w pliku obrazu. Dane są kopiowane bezpośrednio pomiędzy urządzeniami – warunkiem jest jednoczesne podpięcie obu dysków do komputera.

W przedstawianym przykładzie wybieramy najczęściej używana opcję zapisu obrazu dysku w pliku (device-image).



Rys 3.6. Menu wyboru trybu pracy
(źródło: opracowanie własne)

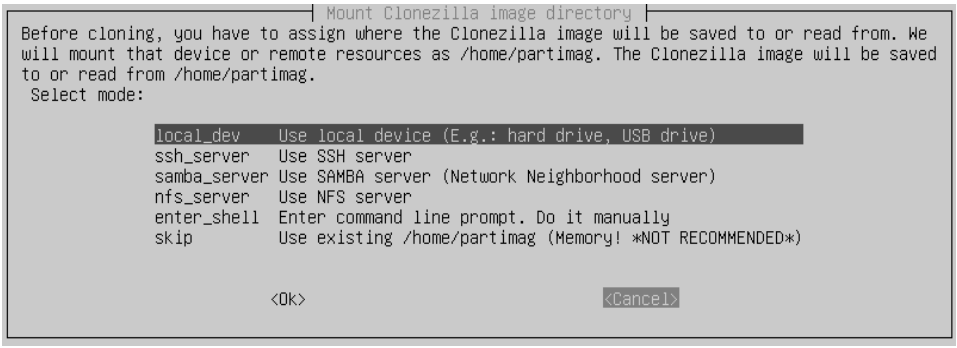
Kolejny ekran (rysunek 3.7) pozwala na dokonanie wyboru rodzaju magazynu danych, jaki chcemy użyć do zapisania pliku obrazu. Do wyboru mamy następujące opcje:

- local_dev – obraz zostanie zapisany na urządzeniu podłączonym lokalnie do komputera, może to być dysk twardy lub inny nośnik pamięci pozwalający na zapis danych; jest to rozwiązanie pozwalające na najszybsze utworzenie pliku obrazu,
- ssh_server – plik obrazu zostanie przesłany na serwer z wykorzystaniem protokołu ssh. Rozwiązanie to wymaga podłączenia komputera do sieci komputerowej lecz pozwala tworzyć obrazy dysków bez podłączania dodatkowych nośników danych. Zaletą tego rozwiązania jest możliwość umieszczenia pliku obrazu na dowolnym serwerze w sieci posiadającym uruchomioną usługę ssh.
- Transmisja jest zaszyfrowana, jednak wadą tego rozwiązania jest powolny transfer danych wynikający zarówno z ograniczeń nakładanych przez przepustowość łącza, jak i algorytm szyfrujący,
- samba_serwer – opcja pozwalająca na umieszczenie pliku obrazu na serwerze plików lub dysku sieciowym; opcja ta współdziała z dyskami

udostępnianymi przez systemy Windows oraz Linuxową usługę samba; serwer plików musi być podłączony do tej samej sieci lokalnej; tworzenie obrazu jest nieco szybsze niż podczas przesyłania obrazu protokołem SSH, lecz nadal dużo wolniejsze od kopiowania na urządzenie podłączone bezpośrednio do komputera; transmisja nie jest zabezpieczona,

- nfs_server – opcja analogiczna do przedstawionej w poprzednim punkcie, współdziałająca z sieciowym systemem plików NFS (ang. *Network File System*) pozwalającym na wymianę plików pomiędzy komputerami pracującymi pod kontrolą systemu Linux,
- enter_shell – opcja pozwalająca na przejście do powłoki systemu Linux i samodzielnego zdefiniowania miejsca przeznaczonego na zapis pliku obrazu; umożliwia wykorzystanie niestandardowych magazynów plików,
- skip – powoduje zrezygnowanie z wyboru miejsca zapisu pliku obrazu, domyślnym miejscem będzie wówczas katalog udostępniony na RAM-dysku, rozwiązanie to nie jest polecane, ponieważ zawartość RAM-dysku tracona jest podczas restartu komputera; rozwiązanie to nie pozwala na trwałe zapisanie obrazów dysków.

Wybieramy opcję local_dev ze względu na najszybsze działanie i powszechną dostępność przenośnych urządzeń do zapisu danych (dyski USB, pendrive).



Rys 3.7. Menu wyboru rodzaju magazynu danych

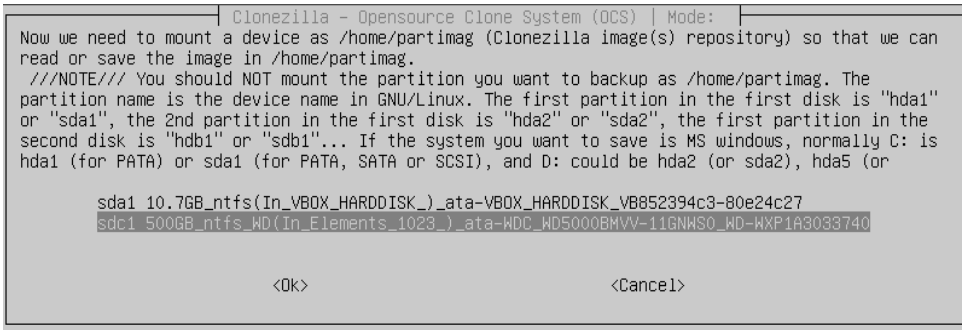
(źródło: opracowanie własne)

Wybranie opcji local_dev spowoduje wyświetlenie komunikatu, informującego użytkownika, że ma jeszcze możliwość podłączenia zewnętrznych urządzeń USB, które mogą zostać użyte jako repozytorium obrazów (rysunek 3.8). Jeżeli użytkownik zdecyduje się na podłączenie dodatkowego urządzenia powinien odczekać pewien czas aż urządzenie zostanie wykryte przez system operacyjny.

Rys 3.8. Komunikat o możliwości podłączenia urządzenia USB

(źródło: opracowanie własne)

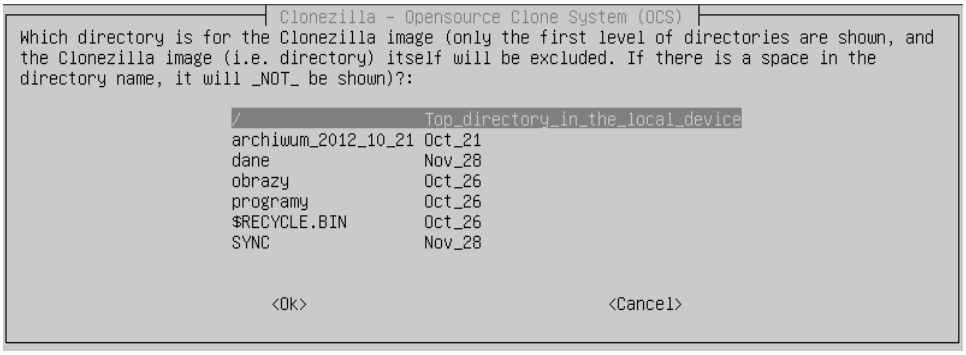
Wszystkie nośniki danych dostępne w systemie operacyjnym zostaną wylistowane przez program (rysunek 3.9). Z pośród nich wskazujemy ten, na którym zostanie zapisany plik obrazu. Należy zwrócić uwagę na to, że wylistowane zostały woluminy (w przykładzie na rysunku partycje). Wybrana partycja nie może znajdować się na dysku, którego obraz będziemy wykonywać.



Rys 3.9. Wybór urządzenia do zapisu pliku obrazu

(źródło: opracowanie własne)

Po zatwierdzeniu wyboru woluminu program wyświetli jego zawartość (rysunek 3.10), prosząc o wybranie katalogu repozytorium, w którym ma zostać zapisany plik obrazu. Do wyboru mamy katalog główny woluminu oraz jego katalogi pierwszego poziomu. Nie ma możliwości wybrania podkatalogów znajdujących się głębiej w strukturze.



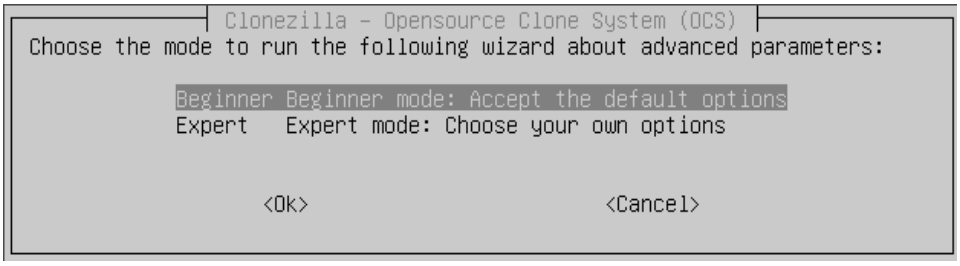
Rys 3.10. Wybór katalogu repozytorium

(źródło: opracowanie własne)

W kolejnym kroku system prosi o wybranie trybu pracy (rysunek 3.11). Do wyboru mamy:

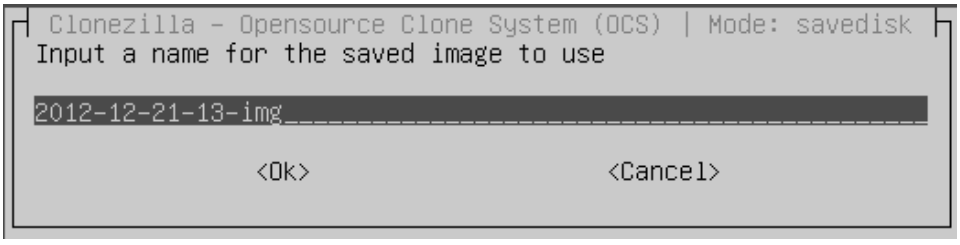
- Beginner – tryb, w którym przyjmowane są domyślne opcje tworzenia obrazu,
- Expert – tryb pozwalający użytkownikowi samodzielnie dostosowywać zaawansowane ustawienia pracy programu.

Wybieramy tryb Beginner. Domyślne opcje są dostosowane do tworzenia obrazów powszechnie używanych systemów plików i nie ma konieczności ingerowania w nie.



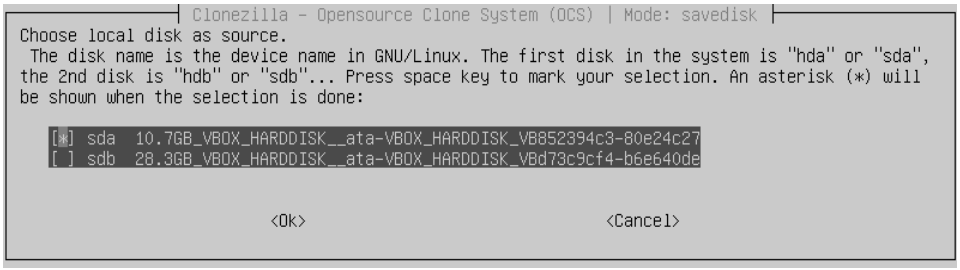
Rys 3.11. Wybór trybu pracy
(źródło: opracowanie własne)

Następnie zostaniemy poproszeni o podanie nazwy pliku obrazu (rysunek 3.12).



Rys 3.12. Określenie nazwy pliku
(źródło: opracowanie własne)

Na kolejnym ekranie (rysunek 3.13) pojawi się lista dysków, z której wybieramy nośnik, którego obraz będziemy wykonywać. Warto zauważyć, że na liście nie zostało wyświetlone urządzenie zawierające wolumin wybrany jako repozytorium obrazów.



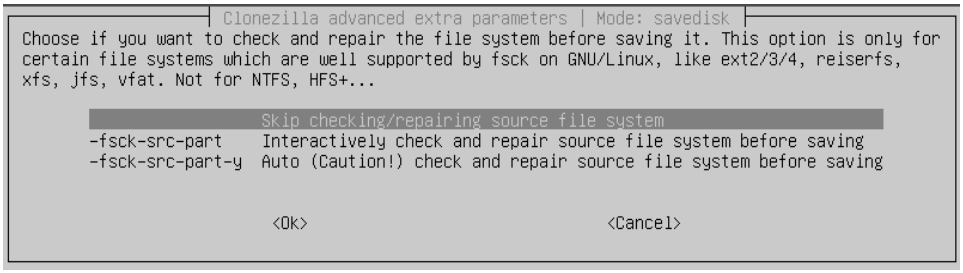
Rys 3.13. Wybór dysku źródłowego
(źródło: opracowanie własne)

Przed przystąpieniem do tworzenia obrazu dysku program proponuje użytkownikowi sprawdzenie poprawności struktury systemu plików i jego ewentualne naprawienie (rysunek 3.14). Użytkownik ma możliwość wybrania jednej z trzech opcji:

- Skip checking/repairing source file system – pominięcie sprawdzania i naprawiania systemu plików,
- -fsck-src-part – interaktywne naprawianie systemu plików, użytkownik potwierdza wykonanie napraw,
- -fsck-src-part-y – automatyczne wykonanie napraw systemu plików, użytkownik nie ma możliwości ingerowania w proces.

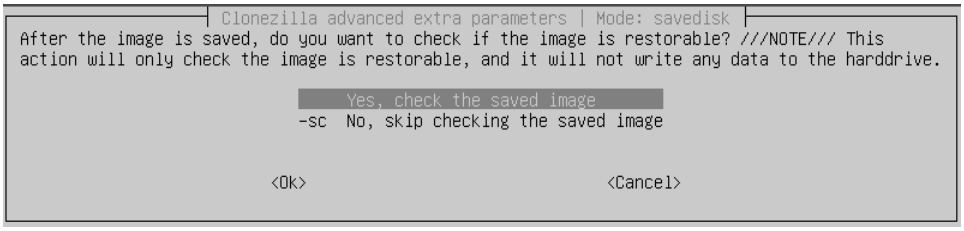
Naprawianie struktury systemu plików zawsze jest operacją ryzykowną. Nawet twórcy oprogramowania Clonezilla zalecają stosowanie jej jedynie dla systemów plików wspieranych przez polecenie `fsck` systemu Linux. Niewskazane jest wykonywanie tej operacji dla Windowsowych

systemów plików. W prezentowanym przykładzie również pominięto naprawianie systemu plików. Jeżeli sprawdzanie jest konieczne najlepiej jest wykonywać je za pomocą narzędzi dołączonych do systemu operacyjnego, który w pełni wspiera dany system plików.



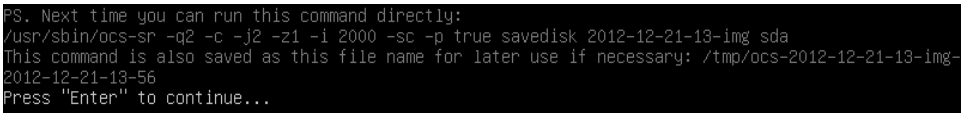
Rys 3.14. Wybór opcji sprawdzania poprawności struktury systemu plików
(źródło: opracowanie własne)

W kolejnym kroku program pyta, czy po utworzeniu obrazu przystąpić do sprawdzenia czy możliwe jest przywrócenie struktury dysku przy użyciu utworzonego obrazu (rysunek 3.15). Operację tą można pominąć lub wybrać sprawdzanie pliku obrazu. Operacja ta pozwala upewnić się, że utworzony plik pozwala na poprawne przywrócenie struktury dysku. Sprawdzenie nie ma wpływu na sam proces tworzenia obrazu. Wymaga jednak dosyć dużo czasu.



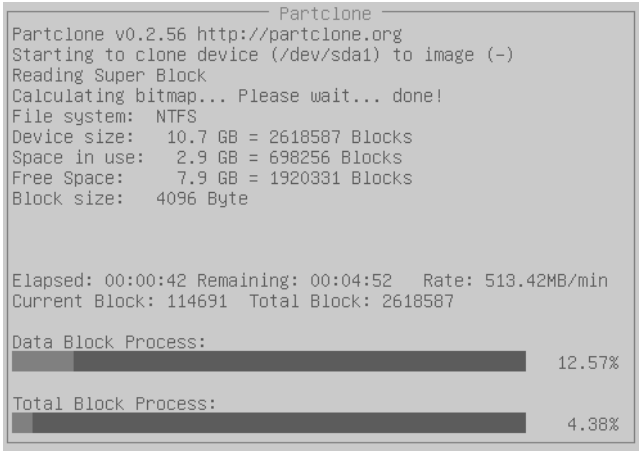
Rys 3.15. Sprawdzanie poprawności pliku obrazu
(źródło: opracowanie własne)

Po przejściu dalej otrzymamy jeszcze informację, że możliwe jest uruchomienie tworzenia obrazu bezpośrednio z konsoli bez konieczności wybierania poszczególnych opcji oraz zostanie zaprezentowana składnia polecenia wykonującego polecenie, które zostało wygenerowane poprzez wybieranie poszczególnych opcji (rysunek 3.16).



Rys 3.16. Składnia polecenia tworzącego obraz dysku
(źródło: opracowanie własne)

Ostatnim etapem jest potwierdzenie chęci wykonania obrazu dysku poprzez wciśnięcie klawisza „y". W tym momencie rozpocznie się tworzenie obrazu dysku, którego postęp możemy obserwować na ekranie komputera (rysunek 3.17). Wyświetlany jest postęp tworzenia obrazów poszczególnych partycji oraz całkowity postęp operacji.

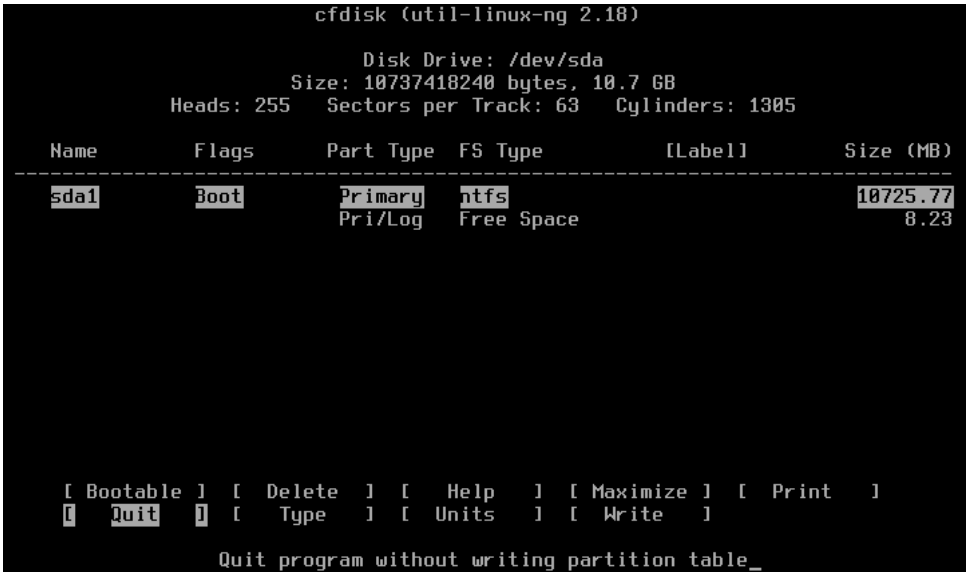


Rys 3.17. Okno prezentujące postęp tworzenia obrazu dysku
(źródło: opracowanie własne)

3.2. OBRAZ MASTER BOOT RECORD

Master boot record (MBR) to struktura obejmująca pierwsze 512B dysku twardego. Zapisane są w niej takie informacje, jak program rozruchowy czy tablica partycji. Uszkodzenie MBR skutkuje zarówno brakiem możliwości uruchomienia systemu operacyjnego, jak i brakiem możliwości zidentyfikowania położenia partycji, a co za tym idzie brakiem możliwości odczytania danych zapisanych na dysku. W obu tych przypadkach warto posiadać kopię MBR. Kopia taka znajduje się w obrazie dysku – o ile taki wykonaliśmy – lecz nie wszystkie programy potrafią odtworzyć jedynie zawartość MBR. Warto więc stworzyć niezależną kopię zawartości MBR. Można to wykonać przy pomocy dowolnej dystrybucji systemu Linux. Jako, że nie na każdym komputerze jest zainstalowany system Linux warto użyć live CD. Do przygotowania przykładu prezentowanego w niniejszym rozdziale użyto RescueCD, dostępnego pod adresem http://www.sysresccd.org/Download.

Tworzenie kopii MBR rozpoczynamy od uruchomienia komputera z live CD. Istotne by możliwe było użytkowanie uruchomionego systemu Linux z prawami użytkownika root. Po uruchomieniu warto sprawdzić, jakie partycje są na dysku i czy na pewno prawidłowo udało się określić nazwę, pod jaką widoczny jest dysk w systemie. Możemy do tego celu użyć polecenia `cfdisk`. Jako argument polecenia podajemy ścieżkę do pliku urządzenia dysku: `cfdisk /dev/sda`. Jako rezultat dostaniemy informację o dysku i jego strukturze. Przykładowy wygląd ekranu polecenia `cfdisk` został przedstawiony na rysunku 3.18.



Rys 3.18. Okno programu cfdisk
(źródło: opracowanie własne)

Na podstawie informacji dostarczonych przez polecenie `cfdisk` wiemy, że na dysku istnieje partycja z systemem plików ntfs. Wykorzystamy ją do zapisania kopii master boot recordu. Aby jej użyć należy ją najpierw podmontować w systemie. Tworzymy katalog, w którym podmontujemy partycję: `mkdir /a`. Następnie montujemy partycję w nowo utworzonym katalogu: `mount -t ntfs-3g /dev/sda1 /a`. W tym momencie możemy przystąpić do wykonania kopii MBR. Wykorzystamy do tego polecenie `dd`. Jest to polecenie pozwalające na kopiowanie zawartości jednego pliku do drugiego. Jako że urządzenia są reprezentowane w systemie Linux jako pliki, możemy na nich operować tak jak na plikach. W poleceniu `dd` plik źródłowy podajemy poprzedzając jego ścieżkę znakami „`if=`", jest to skrót od angielskich słów „input file". Kopię MBR zapiszemy w pliku, który umieścimy na podmontowanej partycji.

Ścieżkę do pliku wynikowego poprzedzamy znakami „of=" będącymi skrótem od angielskich słów „output file". Pozostaje jeszcze kwestia liczby skopiowanych bajtów danych. Nie możemy bowiem kopiować całego dysku. Zależy nam tylko na pierwszych 512 bajtach. Polecenie `dd` pozwala na kopiowanie danych blokami danych. Rozmiar bloku danych podajemy w bajtach i poprzedzamy go znakami „`bs=`". Możemy również określić liczbę kopiowanych bloków poprzedzając ją znakami „`count=`". Polecenie kopiujące MBR do pliku o nazwie mbr umieszczonego na podmontowanej partycji będzie miało więc postać: dd if=/dev/sda of=/a/mbr bs=512 count=1. Prawidłowe wykonane polecenie skopiuje pierwsze 512 bajtów dysku i zapisze je w pliku. Komunikaty zwracane przez polecenie oraz listing zawartości partycji wraz z utworzonym plikiem zawierającym kopię MBR zostały zaprezentowane na rysunku 3.19.

Rys 3.19. Wykonanie kopii MBR dysku /dev/sda
(źródło: opracowanie własne)

Odtworzenie obrazu MBR przebiega analogicznie jak wykonanie kopii. Nie musimy jednak dbać o ilość skopiowanych danych, bowiem w pliku znajduje się jedynie 512 bajtów danych. Wystarczy więc wykonanie polecenia: `dd if=/a/mbr of=/dev/sda`. Polecenie to skopiuje całą zawartość pliku do pierwszego sektora dysku.

3.3. KOPIA ZAPASOWA

Kopia zapasowa (inaczej backup lub archiwizacja danych) to kopia danych wykonana w celu zabezpieczenia się przed ich utratą. Oczywiście zamierzony efekt uda się uzyskać tylko wtedy, gdy kopia będzie nieuszkodzona. Używamy jej bowiem do skopiowania z powrotem danych do ich oryginalnej lokalizacji po ich uszkodzeniu lub zniszczeniu. Z tego warunku wynikają dwa następne: kopia musi być odpowiednio zabezpieczona przed uszkodzeniem oraz odpowiednio oznakowana, tak aby można było ją jednoznacznie zidentyfikować.

Odpowiednie zabezpieczenie to zapewnienie kopii bezpieczeństwa ochrony przed wszystkimi czynnikami mogącymi ją uszkodzić a zwłaszcza przed tymi, które mogą uszkodzić oryginalne dane – jednoczesne uszkodzenie oryginalnych danych oraz ich kopii bezpieczeństwa powoduje nieodwracalną utratę danych. Czynnikami ryzyka najczęściej są tu:

- Niewłaściwe parametry zasilania – najczęściej przepięcia niszczące dane zapisane na dyskach twardych podłączonych do działających urządzeń. Zabezpieczenie przed tym czynnikiem ryzyka polega na umieszczeniu kopii na nośniku, który nie może zostać uszkodzony w wyniku niewłaściwych parametrów zasilania. Może być to dowolny nośnik odizolowany od elementów mogących przenieść na niego niebezpieczne ładunki. Może to również być nośnik niewrażliwy na działania prądu elektrycznego np. dysk optyczny.
- Przypadkowe lub celowe usunięcie danych – działanie człowieka często jest przyczyną utraty lub uszkodzenia danych. Zabezpieczenie polega na umieszczeniu danych w lokalizacji niedostępnej dla osób mogących uszkodzić dane. W przypadku zabezpieczania przed zwykłymi użytkownikami wystarczy zmiana uprawnień dostępu do kopii bezpie-

czeństwa poprzez pozostawienie im jedynie prawa odczytu. Jeżeli jednak chcemy zabezpieczyć się przed umyślnym działaniem administratora systemu poziom trudności znacznie wzrasta. Konieczne jest bowiem umieszczenie kopii danych w miejscu niedostępnym dla administratora. Możliwe jest to poprzez umieszczenie kopii bezpieczeństwa w sejfie, do którego dostęp możliwy jest jedynie przy udziale dwóch osób. Realizowane jest to w ten sposób, że sejf wyposaża się w dwa zamki. Do każdego z nich klucz posiada inna osoba. Aby otworzyć sejf konieczna jest obecność dwóch osób. Niemożliwe jest więc zniszczenie kopii bez udziału drugiej osoby.

- Katastrofa niszcząca lub czyniąca niedostępnym wszystko, co znajduje się w ograniczonym obszarze. Do takich katastrof należą pożary, powodzie lub zalania, trzęsienia ziemi, skażenia chemiczne i biologiczne, huragany, wybuchy oraz itp.
- Jako, że działanie wymienionych czynników polega na niszczeniu lub uniedostępnianiu pewnych obszarów, kopie zapasowe będą skutecznym zabezpieczeniem jedynie wtedy, gdy znajdą się poza obszarem katastrofy i jednocześnie możliwe będzie uzyskanie dostępu do nich. Konieczne jest więc określenie rodzaju katastrof, jakie mogą się zdarzyć oraz ich potencjalnego zasięgu. Kopie bezpieczeństwa muszą zostać umieszczone poza zasięgiem wszelkich katastrof, które mogą zagrozić danym zgromadzonym w systemie informatycznym.
- Zabezpieczenie przed tego rodzaju zagrożeniami w warunkach występujących w Polsce realizujemy najczęściej poprzez umieszczenie nośnika z kopią zapasową w innym budynku, a przynajmniej w innym pomieszczeniu niż przechowywane są dane systemu informatycznego. Kopia bezpieczeństwa umieszczana jest w sejfie spełniającym funkcje zabezpieczenia przeciw kradzieżowego, przed niepowołanym

dostępem, przeciwpożarowego oraz przeciw zalaniu. Dodatkowo niektóre kopie umieszczane są w skrytkach bankowych. Pozwala to na uzyskanie dodatkowej ochrony przeciw kradzieżowej oraz na umieszczenie kopii w odległym budynku, posiadającym odpowiedniej klasy zabezpieczenia.

- Dobrym, często stosowanym rozwiązaniem w większych firmach jest przechowywanie danych w innym oddziale firmy, położonym w oddalonej lokalizacji. Dane do archiwizacji są kopiowane kompresowane oraz szyfrowane. Następnie w postaci zaszyfrowanej przesyłane są do innego oddziału w celu umieszczenia ich na nośniku odpowiednim do przechowywania w sejfie.

Jak łatwo zauważyć, trudno jest dobrać uniwersalny typ kopii zapasowej, tak by zabezpieczała przed wszystkimi zagrożeniami i jednocześnie była szybko dostępna w razie awarii. W praktyce wykonuje się różnego rodzaju kopie. Najczęściej tworzy się harmonogram wykonywania kopii zapasowej. Popularnym rozwiązaniem jest codzienne tworzenie kopii zapasowej przechowywanej w sejfie firmowym oraz dodatkowo raz w tygodniu lub w miesiącu tworzenie kopii zapasowej umieszczanej w bezpiecznym miejscu (np. w skrytce bankowej).

W niniejszym rozdziale przedstawione zostanie rozwiązanie automatycznego tworzenia kopii zapasowej na dodatkowym dysku zainstalowanym w komputerze oraz przenoszenia tej kopii na inny komputer. Tak wykonaną kopię możemy w dowolnym momencie przenieść na nośnik zewnętrzny. Wymaga to jednak ingerencji użytkownika, którego zadaniem jest podłączenie nośnika do nagrania kopii oraz wydanie polecenia kopiowania.

Wykonanie kopii na innym dysku ma dodatkową zaletę, która nie została wcześniej wspomniana. Otóż najczęstszym problemem występującym podczas pracy z danymi jest ich przypadkowe usunięcie lub uszkodzenie czy też modyfikacja. Utworzenie kopii zapasowej na dysku umieszczonym w komputerze pozwala na udostępnienie tej kopii użytkownikom systemu w trybie tylko do odczytu. Dzięki temu użytkownicy systemu będą mogli samodzielnie odzyskać utracone pliki, bez konieczności angażowania w to administratora systemu. Konieczne jest oczywiście ustawienie odpowiednich uprawnień dostępu do poszczególnych zbiorów danych umieszczonych w archiwum.

W przykładzie przedstawione jest tworzenie archiwum na serwerze plików pracującym pod kontrolą systemu Linux. Na serwerze znajduje się kilka katalogów zawierających dane różnych grup użytkowników. Z założenia archiwum danego katalogu ma być dostępne tylko dla grupy, która ma prawa do archiwizowanego katalogu.

Struktura partycji na serwerze może zostać wyświetlona przy pomocy polecenia `df -h`. W naszym przykładzie ma ona postać przedstawioną na listingu 3.1.

W przedstawionej strukturze dane umieszczone są na urządzeniu oznaczonym jako `/dev/md0`. Jest to macierz RAID zamontowana w katalogu `/home`, w którym znajdują się podkatalogi zawierające poszczególne zbiory danych. Listing zawartości katalogu /home został przedstawiony na listingu 3.2.

```
Filesystem Size Used Avail Use% Mounted on
/dev/root  9.4G 4.0G 5.4G 43% /
/dev/sdb4  1.7T 1.7T  65G 97% /mnt/archiwum
/dev/sdc5  382G 351G  32G 92% /mnt/archiwum_dzienne
/dev/sdc7  1.2T 1.1T 54G 96% /mnt/archiwum_miesieczne
/dev/sdc6  196G 5.3G 191G  3% /mnt/instalki
/dev/md0   94G  67G  27G 72% /home
```

Listing 3.1 Lista partycji

(źródło: opracowanie własne)

```
root@ALFA:~# ls -l /home/
total 3
drwxrwx--- 60 root biuro 2168 2012-10-18 10:51 dane/
drwxrwx--- 198 ksg ksieg 8224 2012-10-03 13:02
ksiegowosc/
drwxrwx--- 14 root proj 1224 2012-09-06 11:43 projek-
ty/
drwxrwx--- 14 root dyr 1414 2012-08-07 13:25 dyrek-
cja/
drwxrwxrwx 14 root users 2078 2012-11-02 08:56 wymia-
na/
```

Listing 3.2 Listing zawartości katalogu `/home`

(źródło: opracowanie własne)

Jak widać na powyższym listingu, do każdego katalogu przypisana jest inna grupa użytkowników, która ma do niego pełne prawa. Kopia zapasowa każdego z katalogów musi zostać wykonana niezależnie i umieszczona w oddzielnym katalogu, do którego prawa odczytu i wyko-

nania nadane zostaną tylko grupie mającej prawo do katalogu oryginalnego. Wyjątek stanowi katalog wymiana, do którego wszyscy użytkownicy mogą mieć prawo odczytu.

Archiwa będą zapisywane w katalogu `/mnt/archiwum`. Jak widać na listingu 3.1 jest to partycja `/dev/sdb4` o rozmiarze 1,7 TB.

Jako, że rozmiar partycji jest dosyć duży pozwala na przechowywanie wielu archiwów. Jest to korzystne ze względu na możliwość podejrzenia starszych wersji danych. Przyjmiemy więc rozwiązanie pozostawiania wszystkich archiwów jakie się zmieszczą. Przed przystąpieniem do wykonania kolejnego archiwum konieczne jest zwolnienie miejsca na dysku poprzez usunięcie najstarszych archiwów. Operacja ta będzie najłatwiejsza do wykonania jeżeli nazwy katalogów zawierających archiwa będą ułożone w kolejności alfabetycznej. Wówczas możliwe będzie usunięcie pierwszych katalogów na liście, które zawierają najstarsze archiwa. Konieczne jest określenie ilości dostępnego miejsca na partycji `/dev/sdb4`. Skorzystamy tu z polecenia `df -m`, które zwraca ilość wolnego miejsca na wszystkich zamontowanych partycjach, podając je w megabajtach. Jako, że interesuje nas określenie ilości wolnego miejsca na jednej z partycji, skorzystamy z operatora strumieniowego, by przekazać wynik zwrócony przez `/dev/sdb4` do polecenia `grep sdb4`, które wybierze linię zawierającą ciąg znaków „sdb4”. Będzie to linia zawierająca informacje o interesującej nas partycji. Wynik działania polecenia `df -m|grep sdb4` zaprezentowany został na listingu 3.3.

```
/dev/sdb4  1781741  1715610 66131 97% /mnt/archiwum
```

Listing 3.3 Wynik działania polecenia `df -m|grep sdb4`

(źródło: opracowanie własne)

Trzecia liczba pokazuje ilość wolnego miejsca na dysku. Jako, że format odpowiedzi zwracanej przez polecenie jest stały, możemy sprawdzić, że ostatnia cyfra trzeciej liczby zawsze jest pięćdziesiątym znakiem w linii. Aby uzyskać trzecią liczbę wystarczy wyciąć poleceniem `cut` interesującą nas liczbę znaków. Jako, że partycja zawierająca dane do archiwizacji ma rozmiar 94 GB, to możemy mieć pewność, że ilość wolnego miejsca potrzebnego na archiwum nie będzie większa niż 100 000 MB. Wystarczy więc pobrać siedem ostatnich znaków liczby wolnych megabajtów, aby uzyskać potrzebną informację. Polecenie oraz jego rezultat działania będą miały postać przedstawioną na listingu 3.4.

```
root@ALFA:~# df -m|grep sdb4|cut -c 44-50
66131
```

Listing 3.4 Określenie ilości wolnego miejsca na partycji `/dev/sdb4`

(źródło: opracowanie własne)

Uzyskaną liczbę porównujemy z liczbą niezbędnych megabajtów wolnego miejsca niezbędną do wykonania archiwum. Liczbę tą możemy uzyskać przy pomocy polecenia `du`, które zwraca rozmiar wybranego katalogu. Jednak wykonanie polecenia trwa dosyć długo i obciąża dysk. O wiele łatwiej będzie skorzystać ze znanego nam rozmiaru partycji przechowującej dane. Jeżeli zapewnimy na partycji `/dev/sdb4` nieco ponad 94 GB wolnego miejsca na zapis danych, na pewno uda się tam zmieścić wykonanie archiwum.

Aby zwolnić niezbędną ilość wolnego miejsca skorzystamy z pętli `while`, w której umieścimy operację usuwania pierwszego katalogu z listy. Aby uzyskać nazwę katalogu do usunięcia użyjemy polecenia `ls` wyświetlającego listing katalogu `/mnt/archiwum`. Otrzymany wynik

przy pomocy operatora strumieniowego przekażemy do polecenia `head -1`, które zwróci jedynie pierwszą linię z podanego listingu. Wynik wykonania tego polecenia podamy jako parametr do polecenia `rm -rf`, które usunie podany katalog. Pełna postać pętli zwalniającej niezbędną ilość miejsca na partycji docelowej została przedstawiona na listingu 3.5.

```
cd /mnt/archiwum
while [ `df -m|grep sdb4|cut -c 44-50` -le 100000 ]
do
rm -rf `ls /mnt/archiwum|head -1`
done
```

Listing 3.5 Zwolnienie co najmniej 100 000 MB miejsca na partycji `/dev/sdb4`
(źródło: opracowanie własne)

Jak już wcześniej wspomniano, każda kopia zapasowa musi zostać oznaczona w sposób umożliwiający jej jednoznaczne zidentyfikowanie. Kopie będą umieszczane w katalogach. Konieczne jest więc opracowanie nazewnictwa katalogów, które pozwoli na odpowiednie oznaczenie kopii, jak również pozwoli na prawidłowe działanie fragmentu skryptu przedstawionego na listingu 3.5. Aby to osiągnąć nazwę każdego z katalogów rozpoczniemy ciągiem tekstowym z bieżącą datą, zawierającym rok, miesiąc oraz dzień. Do tego ciągu dołączymy nazwę katalogu, którego archiwum zostanie umieszczone wewnątrz. Pozwoli to na jednoznaczne określenie zawartości katalogu oraz daty utworzenia archiwum. W przypadku, gdy archiwum wykonujemy częściej niż raz dziennie, do nazwy można dołączyć również godzinę wykonania archiwum.

Ciąg zawierający wymagane składniki daty można uzyskać przy pomocy polecenia `date`. Pełną postać tego polecenia oraz efekt jego wykonania prezentuje listing 3.6.

```
root@ALFA:/mnt/archiwum# date +%Y_%m_%d
2012_12_29
```

Listing 3.6 Tworzenie ciągu daty

(źródło: opracowanie własne)

Uzyskany ciąg daty wykorzystujemy do utworzenia zmiennej zawierającej ścieżkę od katalogu, w którym zostanie zapisane archiwum. Zawartość katalogu źródłowego kopiujemy do katalogu docelowego przy pomocy polecenia `cp – Rp`. Zastosowanie opcje pozwalają na skopiowanie podkatalogów wraz z ich zawartością oraz zachowanie informacji o właścicielu pliku, dacie jego modyfikacji oraz utworzenia, a także oryginalnych praw dostępu. Jednak katalog, który jest tworzony przez skrypt będzie własnością użytkownika root.

Jeżeli chcemy dać innym użytkownikom prawo do tego katalogu należy zmienić jego właściciela i grupę. Możemy tego dokonać za pomocą polecenia `chown`, do którego jako parametry podajemy nazwę użytkownika nowego właściciela oraz nazwę grupy.

Dodatkową operacją konieczną od wykonania jest zmiana praw dostępu do zawartości katalogu. Zmiany tej dokonujemy za pomocą polecenia `chmod –R`. Opcja `R` jest konieczna, aby polecenie zadziałało również dla podkatalogów i ich zawartości. Fragment skryptu wykonujący kopię zapasową katalogu `/home/ksiegowosc` został zaprezentowany na listingu 3.7.

```
cd /home/ksiegowosc
echo zrodlowy `pwd`
kat_docelowy=/mnt/archiwum/`date +%Y_%m_%d`_ksiegowosc
echo docelowy $kat_docelowy
mkdir $kat_docelowy
cp -Rp * $kat_docelowy
chmod -R 550 $kat_docelowy
chown ksg:ksieg $kat_docelowy
```

Listing 3.7 Wykonanie kopii zapasowej katalogu `/home/ksiegowosc`

(źródło: opracowanie własne)

Przedstawiony kod należy umieścić w skrypcie, rozbudowując go o fragmenty kopiujące wszystkie wymagane zbiory danych. Tak przygotowany skrypt może być w prosty sposób wywoływany, bez konieczności każdorazowego wpisywania wszystkich instrukcji. Ponadto skrypty mogą być uruchamiane automatycznie przez cron. Cron jest Linuxowym demonem zajmującym się okresowym uruchamianiem programów. Informacje o programach, jakie mają być wykonane znajdują się w tabelach crontab (Camou, 2000).

Proces cron uruchomiony jest w systemie Linux w tle. Monitoruje on co minutę tablice crontab sprawdzając, czy jest jakieś zadanie do wykonania. Jednocześnie monitoruje zmiany, jakie administrator wprowadził do tablic crontab. Aby zaplanować uruchomienie zaprojektowanego skryptu wykonującego kopię zapasową skorzystamy z polecenia `crontab -e`. Polecenie to pozwala na edytowanie jednej z tablic crontab, w której możliwe jest umieszczanie zadań wykonywanych w czasie określonym przez dodającego wpis. Po wyedytowaniu tablicy zobaczymy jej przykładową zawartość, która została przedstawiona na listingu 3.8.

Analizując zawartość umieszczoną w tablicy crontab możemy zauważyć linie komentarza rozpoczynające się znakiem # oraz wpisy powodujące uruchomienie zadań umieszczonych w innych tablicach. Każdy wpis składa się z pięciu pól określających czas uruchomienia zadania podanego w polu szóstym. Pola oddzielone są od siebie spacjami. Każde z pól może zawierać wartość liczbową lub gwiazdkę. Gwiazdka oznacza cały zakres z możliwego przedziału, czyli zadanie będzie uruchamiane niezależnie od tego, jaka będzie wartość odpowiadająca danemu polu. Poszczególne pola oznaczają (za: Zespół PLD, 2012):

- Pole pierwsze: minuty – przyjmuje liczby z zakresu 0-59.
- Pole drugie: godziny – przyjmuje liczby z zakresu 0-23.
- Pole trzecie: dzień miesiąca – przyjmuje liczby z zakresu 0-31.
- Pole czwarte: miesiąc – przyjmuje liczby z zakresu 1-12.
- Pole piąte: dzień tygodnia – przyjmuje liczby z zakresu 0-7, przy czym zarówno zero jak i siedem oznaczają niedzielę.

```
# Run hourly cron jobs at 47 minutes after the hour:
47 * * * * /usr/bin/run-parts /etc/cron.hourly 1> /dev/null
#
# Run daily cron jobs at 4:40 every day:
40 4 * * * /usr/bin/run-parts /etc/cron.daily 1> /dev/null
#
# Run weekly cron jobs at 4:30 on the first day of the
week:
30 4 * * 0 /usr/bin/run-parts /etc/cron.weekly 1> /dev/null
#
```

```
# Run monthly cron jobs at 4:20 on the first day of the
month:
20  4  1  *  *  /usr/bin/run-parts  /etc/cron.monthly  1>
/dev/null
```

Listing 3.8 Przykładowa zawartość tablicy crontab

(źródło: opracowanie własne)

Posiadając te informacje możemy przyjrzeć się bliżej przykładowi z listingu 3.8. Zadanie pierwsze będzie uruchamiane co godzinę, czterdzieści siedem minut po każdej godzinie, niezależnie od daty i dnia tygodnia – na wszystkich pozostałych polach zostały umieszczone gwiazdki. Zadanie drugie będzie wykonywane raz dziennie o czwartej czterdzieści. Trzecie zadanie będzie wykonywane o czwartej trzydzieści, ale tylko w niedzielę – na polu oznaczającym dzień tygodnia umieszczono wartość „zero" określającą dzień tygodnia, w którym zadanie ma być wykonywane. Ostatni wpis dotyczy zadań uruchamianych raz w miesiącu, pierwszego dnia miesiąca o godzinie czwartej dwadzieścia. Oprócz przedstawionych standardowych harmonogramów uruchamiania zadań istnieją możliwości bardziej złożonego definiowania wykonywania zadań. Możemy definiować zakresy wartości lub ich listy. Możemy na przykład umieścić w harmonogramie zapis: "`1 10-16 * * *`". Oznaczać to będzie, że zadanie będzie wykonywane codziennie od godziny dziesiątej do szesnastej minutę po pełnej godzinie. Zapis "`*/5 * * * *`" oznacza wykonywanie danego zadania co pięć minut. Natomiast określenie czasu w postaci "`1 10,16 * * *`" będzie oznaczało wykonanie zadania dwa razy dziennie. Pierwszy raz minutę po godzinie dziesiątej. Drugi raz minutę po szesnastej.

Wszystkie przedstawione przykłady określenia czasu możemy łączyć w celu określenia precyzyjnego harmonogramu wykonywania kopii zapasowych. W każdej organizacji harmonogram ten będzie musiał być dobrany indywidualnie. Będzie on zależał przede wszystkim od wartości danych oraz od tego, jak często będzie musiała być wykonywana kopia zapasowa. Należy również uwzględnić, w jakich przedziałach czasu modyfikowane są dane oraz kiedy możliwe jest zwiększenie obciążenia serwera plików w celu wykonania archiwizacji plików. W typowych organizacjach praca odbywa się przez pięć dni w tygodniu w godzinach 8-16. Nie ma więc potrzeby tworzenia kopii zapasowych w weekendy. Ponadto wiemy, że nikt nie pracuje w godzinach nocnych, co jest równoznaczne z możliwością dodatkowego obciążenia serwera wykonaniem kopii zapasowej. Wystarczy więc wykonanie archiwizacji pięć razy w tygodniu, po każdym zakończonym dniu roboczym. Wywołanie skryptu tworzącego kopię zapasową może mieć postać:

```
10 2 * * 2-6 /skrypty/backup
```

Wpis tej postaci spowoduje wywołanie skryptu pięć razy w tygodniu, od wtorku do soboty. Każdego dnia skrypt będzie uruchamiany o godzinie drugiej dziesięć w nocy. Dlatego wywoływanie skryptu nie kończy się w piątek, gdyż konieczne jest jeszcze zarchiwizowanie danych zgromadzonych w piątek, a to zgodnie z harmonogramem nastąpi dopiero w sobotę o drugiej dziesięć rano.

3.3.1. PRZENIESIENIE DANYCH NA INNY KOMPUTER

Opisaną powyżej procedurę automatycznego tworzenia kopii zapasowej można wykonywać z wykorzystaniem dodatkowego komputera, na którym zostaną umieszczone archiwizowane dane. Zwiększa to poziom bezpieczeństwa kopii zapasowej ze względu na uniezależnienie danych od jednej maszyny oraz możliwość umieszczenia ich w odległej lokalizacji. Do wykonania takiej kopii niezbędne jest połączenie sieciowe pomiędzy komputerem z danymi oraz maszyną przechowującą kopie zapasowe. Ponadto na komputerze zawierającym dane musi być uruchomiona odpowiednia usługa udostępniająca te dane w sieci.

O ile w przypadku kopiowania danych z serwera plików przy wykorzystaniu sieci lokalnej nie musimy się o to martwić, gdyż istnieje na komputerze usługa udostępniająca dane w sieci, to w przypadku kopiowania danych przy wykorzystaniu innych sieci musimy zapewnić usługę pozwalającą na realizację tego zadania. Taką usługą może być FTP, SSH lub dowolna inna. Oczywiście każda z tych usług ma swoje wady i zalety, których należy być świadomym. Przykładowo FTP przesyła dane w postaci jawnej, podczas gdy ssh szyfruje wszystko przed przesłaniem przez sieć. Na korzyść FTP przemawia natomiast większa prędkość przesyłu danych. Przed zastosowaniem jakiejkolwiek usługi należy przeprowadzić analizę jej właściwości i upewnić się, że zapewni odpowiednio wysoki poziom bezpieczeństwa oraz będzie w stanie spełnić swoją rolę, czyli przesłać dane w odpowiednio krótkim czasie, bez zakłócania pracy systemu informatycznego.

W przedstawionym poniżej przykładzie opisana zostanie procedura kopiowania danych przez sieć lokalną, z wykorzystaniem usługo samba. Analogicznie do przedstawionego w tym rozdziale przykładu kopiowania

danych na inny dysk komputera, również w tym przypadku korzystamy z komputerów wyposażonych z system operacyjny Linux.

Aby wykonać kopię danych przez sieć należy zaprojektować skrypt, który będzie mógł być automatycznie uruchomiony. Skrypt możemy umieścić na dowolnym komputerze. Jeżeli jest to komputer z danymi to na drugim z komputerów należy udostępnić przy pomocy usługi samba dysk z prawem zapisu, na którym zostaną umieszczone pliki kopii zapasowej. Jeżeli skrypt uruchamiamy na komputerze, na którym zostaną umieszczone pliki kopii zapasowej, jedynym wymaganiem jest zainstalowanie na nim klienta usługi samba, umożliwiającego korzystanie z zasobów udostępnianych przez inne komputery za pomocą usługi samba.

Skorzystamy z najprostszego rozwiązania. Skrypt tworzący kopię zapasową umieszczamy na komputerze, na którym zostanie ona umieszczona. Postać skryptu oraz jego wywołanie będą identyczne, jak w przypadku wykonywania kopii na tym samym komputerze przedstawionego wcześniej w tym rozdziale. Jedyną różnicą jest konieczność wcześniejszego podmontowania archiwizowanego katalogu z serwera plików. W tym celu tworzymy pusty katalog w systemie plików. W tym katalogu będzie każdorazowo podmontowywany archiwizowany dysk. Montowanie dysku odbywa się przy użyciu polecenia `mount`. Jako parametry polecenia podajemy typ montowanego systemu plików, uprawnienia (odczyt, zapis), nazwę użytkownika oraz jego hasło, montowany zasób oraz katalog, w którym zostanie on podmontowany. Przykładowa składnia polecenia ma postać:

```
mount -t smbfs -o ro,username=user,password=haslo
//192.168.1.1/dane /mnt/dane/
```

W przedstawionej składni polecenia poszczególne opcje oznaczają:

- -t smbfs: typ systemu plików, w tym przypadku jest to dysk sieciowy udostępniany przez usługę samba,
- -o: oznaczenie, po którym zostaną podane dodatkowe opcje montowania,
- ro: dysk będzie podmontowany w trybie tylko do odczytu,
- username=user: określenie nazwy użytkownika, jaka zostanie użyta do nawiązania połączenia z usługą samba; w miejsce słowa user należy wstawić nazwę użytkownika,
- password=haslo: podanie hasła użytkownika wykorzystywanego do połączenia z serwerem plików; hasło wstawiamy w miejsce słowa haslo; opcja ta niesie ze sobą potencjalne zagrożenie, ponieważ jawnie podajemy w skrypcie hasło użytkownika, należy zadbać o odpowiednie ustawienie uprawnień do pliku, by osoby nieuprawnione nie mogły go odczytać,
- `//192.168.1.1/dane`: określenie adresu serwera plików oraz nazwy zasobu, do którego nawiązujemy połączenie,
- `/mnt/dane/`: wskazanie katalogu na komputerze lokalnym, w którym zostanie podmontowany dysk sieciowy.

3.4. SYNCHRONIZACJA

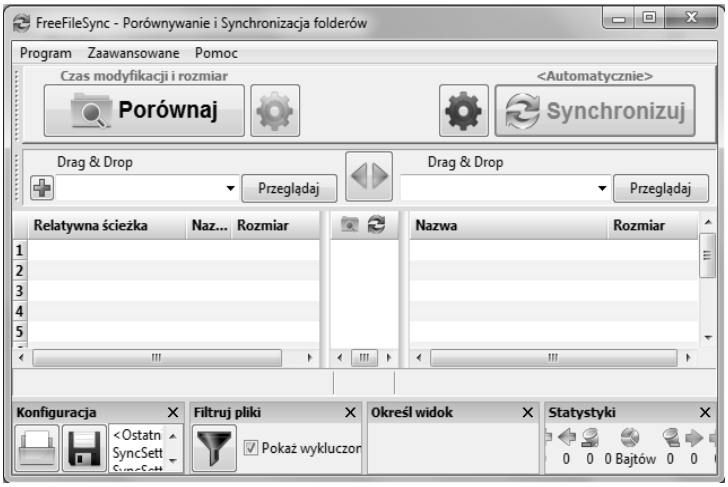
Synchronizacja jest operacją pozwalającą na ujednolicenie zawartości dwóch katalogów. Struktura każdego z katalogów jest porównywana, a następnie przedstawiane są różnice pomiędzy nimi. Użytkownik ma możliwość wyboru, które pliki będą podlegały kopiowaniu. Domyślnie kopiowane są wszystkie pliki. W przypadku plików o tej samej nazwie, różniących się datą modyfikacji, istnieje możliwość nadpisania starszego

pliku lub też utworzenia kopii pliku nowszego przy jednoczesnej zmianie nazwy starszego pliku.

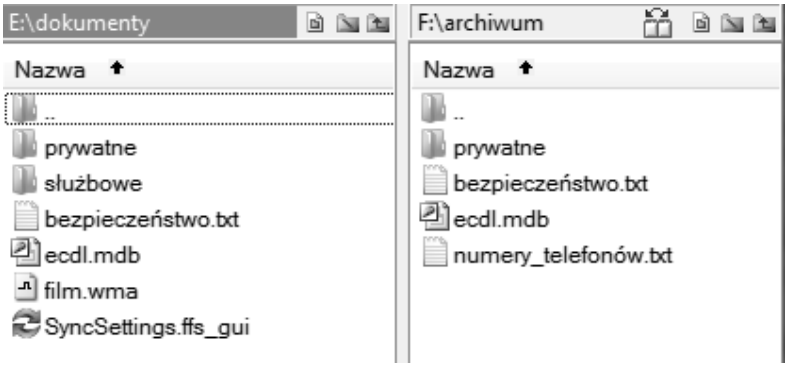
Narzędzie synchronizacji może być używane do ujednolicania zawartości dwóch katalogów przez użytkowników pracujących jednocześnie na dwóch zbiorach danych (np. na dwóch komputerach) lub też do wykonywania kopii zapasowej.

Do wykonania synchronizacji niezbędne jest odpowiednie narzędzie. W Internecie można znaleźć wiele narzędzi służących do synchronizacji katalogów. W niniejszej książce przedstawione zostanie udostępniany na licencji GNU GPL program FreeFileSync dostępny na stronie http://freefilesync.sourceforge.net/. Program ten wymaga instalacji. Po uruchomieniu pokazuje się okno programu. Zostało ono zaprezentowane na rysunku 3.20.

Interfejs programu przypomina interfejs dwupanelowego menedżera plików. W każdym z paneli umieszczamy katalog. Wykorzystujemy do tego technikę *przeciągnij i upuść* lub korzystamy z przycisku przeglądaj. W celu zaprezentowania możliwości, jakie daje synchronizacja przygotowane zostały dwa katalogi: dokumenty oraz archiwum. Ich zawartość została zaprezentowana na rysunku 3.21. Przedstawione katalogi różnią się zarówno liczbą katalogów oraz plików jak też pliki o nazwie „bezpieczeństwo.txt" różnią się zawartością.



Rys 3.20. Okno programu FreeFileSync
(źródło: opracowanie własne)



Rys 3.21. Struktura porównywanych katalogów
(źródło: opracowanie własne)

Po dodaniu katalogów do okna programu wybieramy przycisk „Porównaj". Uruchomi to procedurę porównania zawartości katalogów. Znalezione różnice zostaną wyświetlone w oknie programu tak jak to zostało przedstawione na rysunku 3.22.

Rys 3.22. Porównanie zawartości katalogów
(źródło: opracowanie własne)

Na rysunku 3.22 możemy zauważyć, że po wykonaniu porównania w prawym i lewym panelu programu pojawiły się nazwy plików, a pomiędzy nimi w środkowym panelu symbole oraz pola wyboru. Symbole oznaczają operację, jaka zostanie wykonana na pliku pokazanym w panelu obok. Pole wyboru umożliwia użytkownikowi rezygnację z wykonania zaproponowanej operacji. Podczas synchronizacji mogą być wykonywane takie operacje, jak:

- Kopiowanie plików z prawej strony na lewą – pliki, które istnieją po stronie prawej a nie istnieją po stronie lewej są kopiowane.
- W przypadku, gdy po obu stronach istnieją pliki o tej samej nazwie, lecz ich data modyfikacji lub zawartość różnią się, wówczas operacja kopiowania jest również wykonywana, a starszy plik w zależności od ustawień użytkownika może zostać usunięty lub zapisany jako starsza wersja.

- Kopiowanie plików z lewej strony na prawą - analogicznie, jak zostało opisane powyżej.
- Usuwanie plików – w przypadku, gdy chcemy by pierwszy katalog był wierną kopią drugiego, konieczne jest usunięcie zbędnych plików i katalogów z katalogu archiwum.

W przypadku, gdy pliki po obu stronach są identyczne synchronizacja pozostawi je bez zmian.

Po wykonaniu porównania możliwe jest przystąpienie do synchronizacji. W przedstawionym programie inicjujemy ją przyciskiem „Synchronizuj".

3.5. MACIERZ DYSKÓW

Macierzą dysków nazywamy zespół dysków działających tak, jakby tworzyły jeden wolumin. W niniejszym rozdziale zaprezentowane zostanie tworzenie macierzy programowej RAID 1, czyli tak zwanego mirroringu. Działanie takiej macierzy polega na jednoczesnym umieszczaniu danych na dwóch dyskach. Na każdym z nich zapisywana jest pełna kopia danych, w związku z czym uszkodzenie jednego z dysków nie powoduje uszkodzenia danych. Macierz może nadal poprawnie funkcjonować, z tym, że nie zapewnia już bezpieczeństwa w przypadku uszkodzenia dysku. Aby odbudować jej strukturę należy wymienić uszkodzony dysk, a następnie dołączyć go do macierzy przywracając jej pełną sprawność.

Przedstawiana w przykładzie macierz zostanie utworzona w systemie Linux Slackware 14 i będzie obejmować dwie partycje. Macierze programowe są tworzone przez połączenie partycji utworzonych na dyskach, w przeciwieństwie macierzy sprzętowych, które operują na całych urządzeniach.

Pierwszą czynnością jaką należy wykonać jest upewnienie się czy w systemie zainstalowany jest pakiet `mdadm`, który odpowiada za tworzenie macierzy i ich działanie. W tym celu w konsoli wydajemy polecenie `mdadm`. Jeżeli pakiet jest zainstalowany otrzymamy odpowiedź od polecenia w postaci:

```
usage: mdadm –help
for help
```

Taka odpowiedź świadczy o tym, że usługa `mdadm` jest zainstalowana w systemie operacyjnym. Jeżeli natomiast nie uzyskamy odpowiedzi, konieczne będzie zainstalowanie pakietu `mdadm` w wersji odpowiadającej używanemu systemowi operacyjnemu.

Kolejnym krokiem jest utworzenie partycji, które zostaną użyte do budowy macierzy RAID. Partycje te muszą posiadać ten sam rozmiar. Do utworzenia partycji możemy wykorzystać polecenie `fdisk (Valkor, 2012)`. Jako argument podajemy dysk, którego tablicę partycji będziemy modyfikować. W przypadku operacji na urządzeniu sda polecenie będzie miało postać: `fdisk /dev/sda`. Aby wyświetlić listę partycji istniejących na dysku wpisujemy „p”, co spowoduje wyświetlenie zawartości tablicy partycji aktualnie wybranego dysku. Efekt będzie miał postać przedstawiona na rysunku 3.23.

```
Command (m for help): p

Disk /dev/sda: 19.3 GB, 19327352832 bytes
255 heads, 63 sectors/track, 2349 cylinders, total 37748736 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x26a0c046

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1              63     1992059      995998+  82  Linux swap
/dev/sda2   *      1992060    21527099     9767520   83  Linux
```

Rys 3.23. Informacje o dysku udostępniane przez polecenie `fdisk`

(źródło: opracowanie własne)

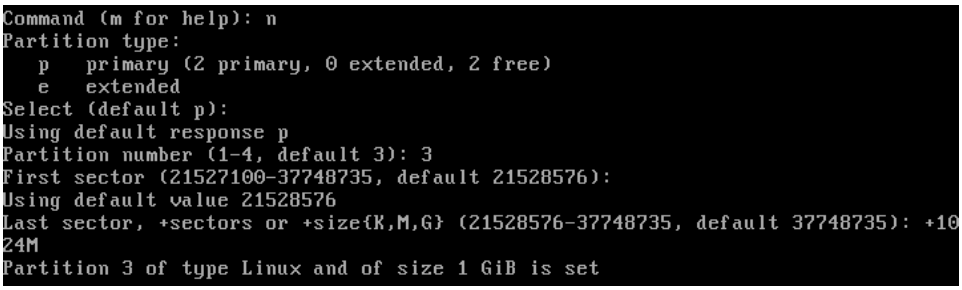
Polecenie `fdisk` wyświetla podstawowe informacje o dysku oraz listę partycji wraz z ich typem. Rozmiary partycji wyświetlane są w blokach. Dla każdej partycji podawane są:

- Nazwa partycji,
- Sektor będący pierwszym sektorem partycji,
- Sektor będący ostatnim sektorem partycji,
- Liczba bloków wchodzących w skład partycji,
- Identyfikator partycji,
- Nazwa rodzaju partycji.

Na podstawie wyświetlonych informacji możemy stwierdzić, czy na dysku istnieje niewykorzystane miejsce umożliwiające utworzenie nowej partycji. Porównujemy w tym celu numer ostatniego bloku ostatniej partycji z liczbą bloków na dysku twardym. Jeżeli istnieje wolne miejsce przystępujemy do utworzenia nowej partycji wpisując literę „n". Zostaniemy poproszeni o:

- wybranie typu partycji (domyślnie jest to partycja podstawowa),
- określenie numeru partycji,
- podanie numeru sektora, który ma by pierwszym sektorem partycji (domyślnie jest to pierwszy wolny sektor),
- wpisanie numeru ostatniego sektora partycji (domyślnie jest to ostatni sektor dysku); możemy jednak zamiast numeru sektora podać rozmiar tworzonej partycji poprzez wpisanie znaku „+", wpisanie żądanej liczby oraz jednostki oznaczanej symbolem:
    - K – kilobajty,
    - M – megabajty,
    - G – gigabajty.

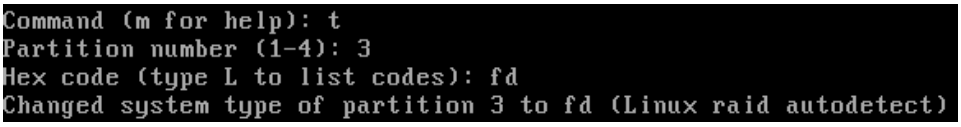
Zrzut ekranu prezentujący przebieg komunikacji z użytkownikiem podczas tworzenia nowej partycji został przedstawiony na rysunku 3.24.



Rys 3.24. Tworzenie partycji za pomocą polecenia `fdisk`

(źródło: opracowanie własne)

Aby utworzona partycja mogła być wykorzystana do utworzenia macierzy RAID konieczne jest uprzednia zmiana jej typu na „Linux raid auto". Zmianę tą również wykonujemy w programie fdisk. Będąc w nim wpisujemy literę „t". Spowoduje to przejście do operacji zmiany typu partycji. Zostaniemy poproszeni o podanie numeru modyfikowanej partycji, a następnie o podanie kodu określającego typ partycji. Listę wszystkich możliwych kodów możemy wyświetlić poprzez wpisanie litery „L". Do typu partycji „Linux raid auto" przypisany jest kod „fd" i taki podajemy. Dialog prezentujący przebieg zmiany typu partycji przedstawiony jest na rysunku 3.25.



Rys 3.25. Zmiana typu partycji za pomocą polecenia `fdisk`

(źródło: opracowanie własne)

Na tym kończy się operacja przygotowywania nowej partycji. Zapisujemy wprowadzone zmiany poprzez wpisanie litery „w" i tym samym zamykamy program `fdisk`. W identyczny sposób tworzymy drugą partycję potrzebną do utworzenia macierzy dysków. Aby macierz chroniła przed awarią dysku należy drugą partycję założyć na innym dysku fizycznym.

W tym momencie możemy przystąpić do tworzenia macierzy. Użyjemy do tego polecenia `mdadm` wraz z niezbędnymi opcjami:

- --create: określenie, ze podejmowaną akcją będzie tworzenie nowej macierzy,
- /dev/md0: podanie nazwy macierzy, która zostanie utworzona; nazwy macierzy rozpoczynają się od liter „md", po których następuje liczba określająca numer kolejnej macierzy,
- --level 1: zdefiniowanie poziomu macierzy dysków, w tym przypadku tworzymy macierz poziomu pierwszego, czyli mirroring,
- --raid-devices=2 /dev/sda3 /dev/sdb4: określenie liczby dysków wchodzących w skład macierzy oraz wyszczególnienie ich.

Polecenie `mdadm` po uruchomieniu wyświetli dodatkowe informacje dla użytkownika oraz poprosi o potwierdzenie utworzenia macierzy. Zrzut ekranu prezentujący tworzenie macierzy przedstawiony został na rysunku 3.26.

```
root@virtual:~# mdadm --create /dev/md0 --level 1 --raid-devices=2 /dev/sda3 /de
v/sdb4
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device.  If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Rys 3.26. Tworzenie macierzy dysków za pomocą polecenia `mdadm`

(źródło: opracowanie własne)

Możemy się upewnić, ze nasza macierz została utworzona. W tym celu możemy wylistować zawartość katalogu `/dev`. Powinniśmy znaleźć tam plik o nazwie `md0`. Jest to plik urządzenia macierzy.
Jego istnienie świadczy o tym, że macierz została utworzona, nie dostarcza jednak żadnych innych informacji. Jeżeli interesuje nas stan macierzy oraz jej konfiguracja możemy wyświetlić zawartość pliku `/proc/mdstat`. Wewnątrz znajduje się informacja o działających w systemie macierzach oraz ich stanie (Michalski, 2012).

Listing zawartości katalogu `/dev` oraz zawartość pliku `/proc/mdstat` zostały zaprezentowane na rysunku 3.27.

```
root@virtual:~# ls /dev/m*
/dev/mcelog  /dev/megadev0  /dev/mixer   /dev/mpt2ctl
/dev/md0     /dev/mem       /dev/mouse@  /dev/mptctl

/dev/mapper:
control
root@virtual:~# cat /proc/mdstat
Personalities : [linear] [raid0] [raid1] [raid10] [raid6] [raid5] [raid4] [multi
path]
md0 : active raid1 sdb4[1] sda3[0]
      995456 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Rys 3.27. Sprawdzanie stanu macierzy
(źródło: opracowanie własne)

Analizując informacje przechowywane w pliku `/proc/mdstat` możemy zauważyć, że w systemie istnieje jedna macierz dysków „md0”. Poszczególne ciągi znaków oznaczają:

- active – macierz jest aktywna, czyli działająca.
- raid1 – rodzaj macierzy to mirroring, czyli RAID poziomu pierwszego,
- sdb4[1] sda3[0] – w skład macierzy wchodzą dwie partycje: sdb4 oraz sda3.

- 995456 blocks – pojemność macierzy to 995456 bloków danych.
- super 1.2 – na macierzy utworzono super blok, czyli strukturę przechowującą dane o macierzy i umożliwiającą jej poprawne podłączenie w wersji 1.2.
- [2/2] [UU] – dwie partycje z dwóch wchodzących w skład macierzy działają poprawnie i są nieuszkodzone.
- W przypadku, gdyby jedna z nich była uszkodzona wpis ten miałby postać [1/2], natomiast w miejscu jednej z liter „U” byłby znak „_”.
- Unused devices none – macierz nie zawiera nieużywanych partycji. W tym miejscu wyświetlane są partycje, które zostały dodane do macierzy jako zapasowe. Zapasowa partycja automatycznie przejmuje rolę uszkodzonej w przypadku wystąpienia awarii. Dane na partycji zapasowej odtwarzane są w tle, a macierz jest odbudowywana w sposób niezauważalny dla użytkownika.

Aby możliwe było zapisywanie danych na macierzy konieczne jest wcześniejsze utworzenie systemu plików. Macierz traktujemy od tej pory tak, jak zwykłą partycję.

Odpowiednią organizację zapisu będzie zapewniał system operacyjny. Tworzenie systemu plików odbywa się poleceniem, które zależy od systemu plików, który chcemy utworzyć.

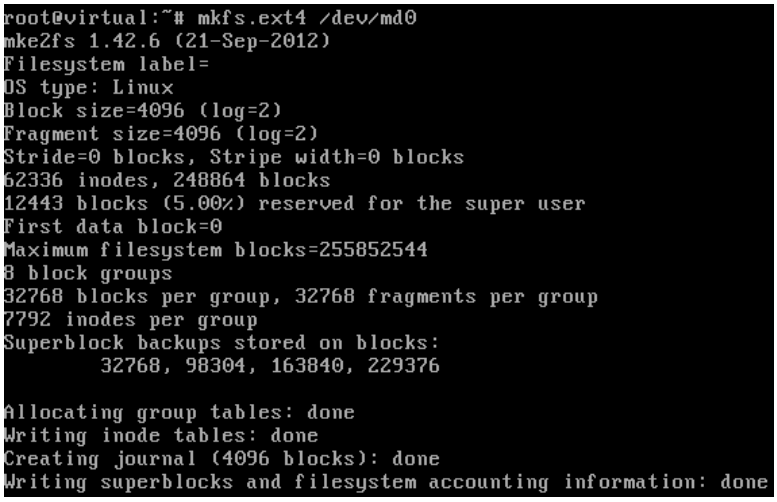
Lista najpopularniejszych systemów plików używanych w Linuxie oraz polecenia je tworzące zostały zestawione w tabeli 3.1. Jako argument do każdego z poleceń podajemy ścieżkę do pliku urządzenia formatowanej partycji.

Tabela 3.1 Polecenia tworzące system plików

System plików	Polecenie tworzące system plików
ext2	mkfs.ext2
ext3	mkfs.ext3
ext4	mkfs.ext4
Reiser FS	mkreiserfs
fat	mkfs.vfat

(źródło: opracowanie własne)

Polecenie tworzące system plików ext4 na utworzonej przez nas macierzy będzie miało postać: mkfs.ext4 /dev/md0. Jego wywołanie powoduje utworzenie systemu plików oraz wyświetlenie informacji podsumowujących proces jego tworzenia, tak jak to zostało zaprezentowane na rysunku 3.28.



Rys 3.28. Tworzenie systemu plików macierzy

(źródło: opracowanie własne)

Macierz została utworzona i może służyć do zapisywania danych. Jednak konieczne jest jeszcze zapisanie konfiguracji macierzy w pliku `/etc/mdadm.conf`. Bez tego konfiguracja macierzy zniknie po pierwszym wyłączeniu komputera. Konfigurację macierzy możemy wygenerować poleceniem: `mdadm –detail –scan`.

Konfiguracja wygenerowana dla przedstawionej w przykładzie macierzy ma postać zaprezentowaną na rysunku 3.29.

Rys 3.29. Wygenerowana konfiguracja macierzy

(źródło: opracowanie własne)

Wygenerowaną konfigurację zapisujemy w pliku `/etc/mdadm.conf`. Uzywamy do tego polecenia: `mdadm –detail –scan > /etc/mdadm.conf`.

Następnie edytujemy utworzony plik dopisując do niego linię zawierającą listę partycji wchodzących w skład macierzy. Kompletny plik powinien mieć taką postać, jak przedstawiono na listingu 3.9.

```
ARRAY /dev/md0 metadata=1.2 name=virtual:0
UUID=a059e7e5:91702208:c473c61c:50459a
devices=/dev/sda3,/dev/sdb3
```

Listing 3.9 Zawartość pliku `/etc/mdadm.conf`

(źródło: opracowanie własne)

Wykonanie wszystkich opisanych powyżej działań pozwala na uzyskanie działającej macierzy dyskowej. Aby rozpocząć korzystanie z niej niezbędne jest jej podmontowanie w systemie operacyjnym. Możemy tego dokonać poleceniem `mount`, którego składnia dla utworzonej w przedstawionym przykładzie macierzy montowanej w katalogu `/mnt/raid` będzie miała postać:

```
mount /dev/md0 –t ext4 /mnt/raid
```

Wadą tego rozwiązania jest konieczność montowania macierzy po każdym ponownym uruchomieniu komputera. O wiele wygodniejsze jest dopisanie odpowiednich instrukcji do pliku `/etc/fstab` powodujących automatyczne montowanie macierzy przez system operacyjny. Przykładowa postać pliku `/etc/fstab` z dopisaną linią pozwalającą na automatyczne podmontowanie macierzy podczas startu systemu została przedstawiona na rysunku 3.30.

```
/dev/sda2       /               ext4        defaults          1   1
#/dev/cdrom     /mnt/cdrom      auto        noauto,owner,ro,commen
t=x-gvfs-show 0   0
/dev/fd0        /mnt/floppy     auto        noauto,owner      0   0
devpts          /dev/pts        devpts      gid=5,mode=620    0   0
proc            /proc           proc        defaults          0   0
tmpfs           /dev/shm        tmpfs       defaults          0   0
/dev/md0        /mnt/raid       ext4        defaults          0   0
```

Rys 3.30. Plik `/etc/fstab`

(źródło: opracowanie własne)

Jak widać na rysunku 3.30 każda linia odpowiada za podmontowanie jednego zasobu. Ostatnia linia dotyczy macierzy dysków.

Dane zawarte w niej umieszczone zostały zgodnie ze strukturą pliku, która obejmuje sześć kolumn, w których odpowiednio umieszcza się:

- Kolumna 1 zawiera ścieżkę do pliku urządzenia blokowego, które będzie podmontowywane,
- Kolumna 2 zawiera ścieżkę do katalogu, w którym zostanie podmontowane urządzenie,
- Kolumna 3 służy do umieszczenia systemu plików, jaki jest używany na podmontowywany urządzeniu,
- Kolumna 4 przeznaczona jest do umieszczenia dodatkowych opcji montowania,
- Kolumna 5 przechowuje parametry przeznaczone dla programu `dump` służącego do archiwizowania partycji,
- Kolumna 6 określa kolejność sprawdzania systemów plików przez program `fsck` w przypadku wykrycia błędów.

4

Steganologia

Cel

Niniejszy rozdział przeznaczony został na przedstawienie zagadnienia steganologii. Jest to jedna z mniej znanych dziedzin ochrony informacji, aczkolwiek nie mniej ważną od powszechnie znanej kryptografii.

Powszechnie znane zastosowanie steganologii to znakowanie wodne, które jest wszechobecne w ochronie utworów oraz praw autorskich. Istnieje jednak wiele innych zastosowań steganografii, o których wielu ludzi nie słyszało. Autor postara się je przybliżyć i przedstawić korzyści z ich stosowania.

Plan

1. Przedstawienie steganologii
2. Zdefiniowanie podstawowych pojęć związanych ze steganologią
3. Omówienie pojęcia „system steganograficzny”
4. Przedstawienie wybranych technik steganograficznych

4.1. PODSTAWOWE POJĘCIA STEGANOLOGII

Steganologią nazywamy naukę, która zajmuje się ukrywaniem cennej informacji w innej nieposiadającej wartości. Steganologia obejmuje dwie przeciwstawne sobie dziedziny: steganografię i stegoanalizę (Garbarczuk & Świć 2005).

Steganografia jest dziedziną nauki zajmującą się metodami ochrony cennej informacji poprzez ukrycie jej w innych danych niemających wartości. Ukrycie ma na celu zabezpieczenie przed wykryciem istnienia cennej informacji przez osoby postronne. Ukrywanie wykonywane jest poprzez wprowadzenie niewielkich, mało znaczących i trudnych do wykrycia zmian do nośnika oryginalnego. Wprowadzone zmiany powinny być niemożliwe do wykrycia przy pomocy zmysłów czy też analizy statystycznej (Petitcolas & Ross & Kuhn 1999).

Stegoanaliza stanowi przeciwieństwo steganografii. Jest to dziedzina nauki zajmująca się opracowywaniem metod łamania zabezpieczeń tworzonych przy pomocy steganografii. W szczególności obejmuje: wykrywanie nośników zawierających ukrytą informację, odczytywanie ukrytej informacji, niszczenie lub modyfikowanie ukrytej informacji, wprowadzanie niejednoznaczności ukrytej informacji.

Stegoanaliza opiera się najczęściej na analizie parametrów nośników. Odbiegające od normy wartości różnych parametrów danych nośnika stanowią wskazanie do dalszej analizy. Każda z metod steganograficznych wprowadza inne modyfikacje danych nośnika, co może stanowić podstawę do zidentyfikowania użytej metody oraz dalszych prac nad odczytaniem ukrytej informacji (Kozieł 2011).

Ochrona danych poprzez ich ukrycie wymaga miejsca, w którym dane zostaną ukryte. W przypadku danych cyfrowych, które są niezależne od nośnika, obiektem, w którym możliwe jest ukrycie mogą być tylko inne dane. Nazywamy je *kontenerem* lub *nośnikiem*. Dane, które chronimy poprzez ukrywanie nazywamy *ukrywanymi danymi, tajną informacją, tajnymi danymi* lub *ukrytym przekazem*. Tajną informację dołączamy do kontenera, tak by była dołączona bezpośrednio do danych. W efekcie powstaje jeden spójny zbiór danych nazywany *stegokontenerem*. Proces ukrywania tajnej informacji w kontenerze nazywamy *dołączaniem* lub *ukrywaniem*. Proces ukrywania przeprowadzany jest za pomocą algorytmu, który umieszcza tajne dane w kontenerze. Aby zwiększyć bezpieczeństwo metod steganograficznych najczęściej w procesie dołączania używany jest *klucz steganograficzny*. Jest to dodatkowa informacja (porcja danych) modyfikująca proces ukrywania. Klucz steganograficzny użyty do ukrywania jest niezbędny do odczytania ukrytych danych. Bez klucza odczytanie jest niemożliwe lub co najmniej bardzo trudne. Proces odczytywania (wydobywania) ukrytej wiadomości nazywamy *ekstrakcją* (Kozieł 2011).

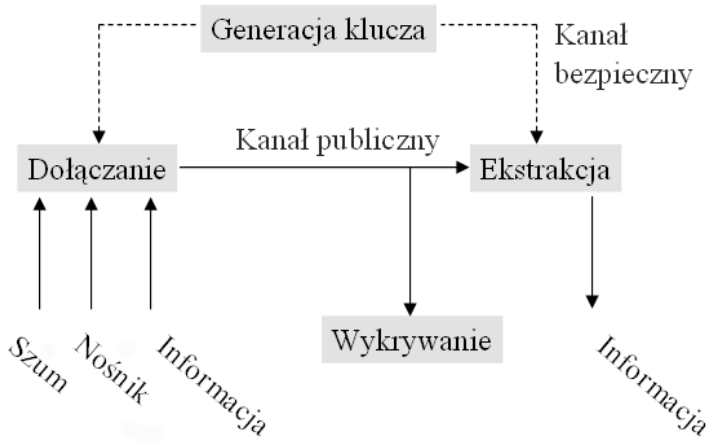
Oczywiście nie tylko upoważnieni użytkownicy będą podejmowali próby odczytania ukrytej wiadomości. Również osoby postronne będą podejmowały działania dążące do wykrycia i odczytania ukrytej informacji. Osobę podejmującą działania mające na celu złamanie zabezpieczeń steganograficznych (atak na stegokontener) nazywamy *stegoanalitykiem*. Poprzez *atak* rozumiemy ogół działań podejmowanych przez stegoanalityka prowadzących do złamania stegosystemu, czyli wykrycia, a następnie odczytania lub uszkodzenia ukrytej informacji.

Ataki możemy podzielić na dwa podstawowe rodzaje:

- Ataki pasywne – podczas których stegoanalityk w żaden sposób nie ingeruje w przesyłaną wiadomość, jedynie podejmuje próbę złamania systemu steganograficznego za pomocą obserwacji danych,
- Ataki aktywne – charakteryzujące się tym, że w celu złamania systemu steganograficznego stegoanalityk ingeruje w strumień danych wprowadzając do niego modyfikacje.

Stegosystemem lub systemem steganograficznym nazywamy ogół środków i działań stosowanych w celu zrealizowania zadania steganograficznego. Model stegosystemu przedstawiony został na rysunku 4.1.

Podstawową operacją steganograficzną jest proces dołączania. Do jego przeprowadzenia wymagane jest posiadanie nośnika oraz ukrywanej informacji. Istotne jest również użycie klucza steganograficznego, bez którego odczytanie informacji będzie możliwe dla wszystkich osób znających zastosowany algorytm. Niekiedy w procesie dołączania używany jest również szum. Dołączany jest on w celu wprowadzenia dodatkowej dezinformacji. Stegoanalitykowi będzie o wiele trudniej odczytać ukrytą informację, jeżeli będzie ona wymieszana z szumem. Utworzony stegokontener używany jest w procesach ekstrakcji oraz wykrywania. Podczas ekstrakcji ukryte dane odczytywane są w sposób autoryzowany za pomocą klucza. Proces wykrywania jest o wiele bardziej skomplikowany. Stegoanalityk musi najpierw poprawnie zidentyfikować nośnik zawierający ukrytą informację, a dopiero następnie może przystąpić do prób odczytania ukrytych danych (Kozieł 2011).



Rys. 4.1 Model systemu steganograficznego
(źródło: Kozieł 2005)

4.2. HISTORIA STEGANOGRAFII

Steganografia jest nauką, której korzenie sięgają czasów starożytności. Sama nazwa wywodzi się od greckich słów *steganos graphei* oznaczających *tajne pismo*.

W starożytnej Grecji steganografia była sztuką ukrywania informacji w taki sposób, aby osoby postronne nie mogły jej dostrzec ani w żaden inny sposób wykryć. Pierwsze wzmianki o steganografii znajdziemy już w mitologii. Zawarta tam została historia Histausa, który więziony przez perskiego króla Dariusza wymyślił prosty, aczkolwiek skuteczny sposób na przesłanie informacji do swojego zięcia, pomimo dokładnego kontrolowania przesyłanych przez niego wiadomości przez służby króla Dariusza. Histiaus najpierw dokładnie ogolił głowę jednego ze swoich niewolników, a następnie kazał wytatuować na niej wiadomość, w której opisał swoje tragiczne położenie. Gdy tylko niewolnikowi odrosły włosy, wysłał go jako posłańca do swojego szwagra z mało ważnym pismem. Kontrola

listu przeprowadzona przez służbę króla Dariusza oczywiście niczego nie wykazała. Posłaniec został więc przepuszczony. Mógł doręczyć list adresatowi. W ten sposób udało mu się przemycić tajną wiadomość. Ponowne ogolenie głowy pozwoliło na bezproblemowe odczytanie ukrytej wiadomości i uwolnienie Histausa (Garbarczuk & Świć 2005).

Innym szeroko znanym przykładem steganografii było pisanie na drewnianych tabliczkach pokrytych woskiem. Działo się to w czasach, gdy drewniane tabliczki były stosowane jako materiał, na którym ryto symbole. Taka tabliczka przenoszona była następnie przez posłańca do adresata. Okazało się jednak, że drewno jest dosyć niewdzięcznym materiałem do precyzyjnego rzeźbienia liter. Zaczęto więc pokrywać tabliczki warstwą wosku, na której z łatwością dawało się wyryć znaki tekstu. Warstwa wosku pokrywająca tabliczkę pozwalała jednakże na ukrycie powierzchni drewna. Wykorzystane zostało to do ukrywania tajnych wiadomości. Były one tradycyjnie rzeźbione w drewnie, które następnie pokrywano warstwą wosku. Tak spreparowana tabliczka wykorzystywana do naniesienia innej wiadomości, niemającej wartości. Ten sposób komunikacji rozpowszechniony był w Chinach (Garbarczuk & Świć 2005).

Sztuka steganografii rozwijana była na całym świecie. W zależności od kultury i dostępnych środków przyjmowała różne formy. W pewnym momencie zaczęto wykorzystywać atrament sympatyczny. Była to substancja niewidoczna dla ludzkiego oka po naniesieniu na nośnik. Odczytanie wiadomości naniesionej na papier za pomocą atramentu sympatycznego możliwe było dopiero po przeprowadzeniu odpowiedniej reakcji, która powodowała wybarwienie się atramentu sympatycznego. Najbardziej popularne było używanie substancji zawierających organiczne związki węgla. Substancje te charakteryzowały się tym, że ciemniały po podgrzaniu. W zależności od dostępności różnych substancji w innych

rejonach świata stosowano odmienne rodzaje atramentu. Często używano do tych celów mleka czy też soku z cytryny lub innych owoców. Napis wykonany za ich pomocą uwidaczniał się dopiero po podgrzaniu zapisanej strony. Popularność tego typu metod prowadziła do stosowania coraz bardziej złożonych substancji, których uwidocznienie było coraz trudniejsze. Najczęściej wymagało przeprowadzenia reakcji chemicznej z inną substancją. Już nie korzystano z podgrzewania, ponieważ metoda ta była znana tak szeroko, że nie zapewniała odpowiedniego poziomu bezpieczeństwa.

Metody steganografii były nieprzerwalnie rozwijane. Ich szczególnie gwałtowny rozwój nastąpił w XX wieku w czasach wojen światowych. Steganografia jest bardzo cenionym narzędziem w działaniach militarnych, szczególnie podczas prowadzenia akcji wywiadowczych.

Na wyróżnienie zasługuje opracowana przez niemieckich naukowców technika mikrokropek. Jest to technika pomniejszania negatywów zdjęć do rozmiaru 0,5 milimetra na 0,5 milimetra. Jednym słowem jest to rozmiar kropki w tekście drukowanym. Stąd też wzięła się nazwa oraz sposób wykorzystania tej techniki. Zdjęcia wykonywane przez pracowników wywiadu, były pomniejszane do rozmiarów kropki, a następnie wklejane do listów w miejsce niektórych z kropek. Tak przygotowane listy były wysyłane pocztą do odbiorcy. Technika ta okazała się bardzo skuteczna, gdyż przez długi czas pozwalała na przesyłanie tajnych danych pomimo kontrolowania poczty przez aliantów. W ten sposób niemieccy szpiedzy przesyłali dokumentację dotyczącą alianckich umocnień, czy raporty na temat ruchu wojsk (Garbarczuk & Świć 2005).

Równolegle rozwijała się steganografia lingwistyczna. Jest to dziedzina steganografii zajmująca się ukrywaniem informacji w tekście poprzez zorganizowanie tekstu tak, by wybrane jego elementy układały się w wiadomość. Wykorzystywane tu były zarówno niuanse gramatyki, jak i preparowano tekst tak, aby żądane elementy pojawiały się w określonych miejscach.

Każdy język pozwala na zapisanie tej samej treści na wiele różnych sposobów. Przykładem może być jedna linia listy zakupów, którą zapiszemy na wiele różnych sposobów:

- `Chleb, mleko i masło`
- `Chleb, mleko, i masło`
- `Chleb i mleko, i masło`
- `Mleko, masło i chleb`
- `Masło, mleko, chleb`

Jak widać wszystkie wymienione powyżej konstrukcje są poprawne. Ich znaczenie również jest takie samo, jednak różnią się one od siebie. Można to wykorzystać do zakodowania ukrytej wiadomości. Wystarczy stworzyć książkę kodową, w której każdej z konstrukcji przypiszemy określone znaczenie. Adresat tej pozornie nic nieznaczącej wiadomości, korzystając z książki kodowej może odczytać wiadomość. Dla osób postronnych będzie ona jednak niemożliwa do odczytania.

Innym przykładem steganografii lingwistycznej jest konstruowanie tekstów tak, aby wybrane litery tekstu tworzyły tajna wiadomość. Często prezentowanym przykładem jest autentyczny tekst przesłany przez jednego z niemieckich szpiegów:

Apparently neutrals protest is thoroughly discourted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

Sekretny przekaz odczytamy łącząc ze sobą drugie litery każdego z wyrazów przedstawionej powyżej wiadomości. Będzie miał on postać: Pershing sails from NY June 1.

Wadą tego rozwiązania jest duża trudność konstruowania tekstu. Narzucone są poszczególne litery, do których należy dopasować tekst. Nawet, jeżeli się to uda to najczęściej tekst brzmi sztucznie i zwraca na siebie uwagę.

Istnieje inne rozwiązanie oparte na wybieraniu liter z istniejącego tekstu lecz pozbawione problemów z tworzeniem treści. Nie należy jednak do steganografii lingwistycznej. Opiera się bowiem na wykorzystaniu dowolnego tekstu drukowanego. Litery ukrytej wiadomości są oznaczane w tekście poprzez nakłuwanie ich szpilką. Niewielkie otworki w kartce papieru są niezauważalne dla ludzkiego oka. Jednak wystarczy popatrzeć na kartkę pod światło by nakłucia stały się widoczne.

Kolejnym przykładem steganografii lingwistycznej jest umieszczanie na określonych pozycjach w tekście całych słów pochodzących z ukrytej wiadomości. Takim przykładem może być tekst, który ukazał się na łamach warszawskiego wydania "Dziennika" w dziale "Świat" w październiku 2007 (tvn24.pl, 2007).

Poniższy tekst na pierwszy rzut oka wygląda zwyczajnie, jednak pierwsze litery kolejnych wersów artykułu "Rice krytykuje Moskwę za szantaż gazowy" układają się w wiadomość, która stanowiła – podobno przypadkowo – swoistą odpowiedź na plotki dotyczące zmiany na stanowisku redaktora naczelnego tej gazety (tvn24.pl, 2007).

Ten rodzaj steganografii również jest trudny do zrealizowania, ze względu na trudności w zredagowaniu naturalnie brzmiącego tekstu zawierającego ukrytą wiadomość.

USA

Rice krytykuje Moskwę za „szantaż gazowy"

Amerykańska sekretarz stanu Condoleezza Rice skrytykowała Rosję za używanie surowców energetycznych jako „broni politycznej".

Stany Zjednoczone szanują interesy Rosji, ale nie może ona używać gazu ziemnego oraz ropy naftowej do tego, by grozić swoim sąsiadom i uzależniać ich od siebie – mówiła w czasie jednego z wykładów. Te ostre słowa padły tydzień po wizycie szefowej dyplomacji USA w Moskwie. Próbowała ona po raz kolejny przekonać prezydenta Władimira Putina, że budowa tarczy antyrakietowej nie zagraża rosyjskim interesom.

W Europie pojawiają się analogiczne zarzuty. Coraz łatwiej usłyszeć, że Rosja angażuje swoje koncerny w typowe akcje polityczne.

Obawy wobec Gazpromu czy Łukoilu nie są już abstrakcyjne. Niedawno szczególnie zapamiętano zagranicę Gazpromu tuż po ostatnich parlamentarnych wyborach na Ukrainie.

Ingerencja tego kremlowskiego superkoncernu i jego żądania, by Ukraina spłaciła mu wobec groźby odcięcia dostaw potężny wielomiliardowy dług, w istotny sposób skomplikowała rozmowy na temat utworzenia nowego ukraińskiego rządu.

Już wcześniej Rosja stosowała podobne naciski wobec sąsiedniej Białorusi. Pod naciskiem Polski UE zdecydowała się niedawno przyjąć zapisy blokujące dalszą ekspansję Gazpromu w Unii.

Polska i kraje bałtyckie sprzeciwiają się także budowie Gazociągu Północnego, który – ich zdaniem – naraża je na groźbę odcięcia dostaw.

pc, reuters

Rys. 4.2 Przykład steganografii lingwistycznej
(źródło: tvn24.pl, 2007)

Wszystkie przedstawione w niniejszym podrozdziale przykłady należą do steganografii analogowej, czyli takiej, która wykorzystuje nośniki analogowe do przechowywania ukrytej informacji.

Nie można przy tej okazji nie wspomnieć o obecnie stosowanych analogowych technikach steganograficznych, jakimi są zabezpieczenia papierów wartościowych. Najlepiej znanymi papierami wartościowymi są banknoty. Na nich również znajdziemy zabezpieczenia steganograficzne. Są nimi chociażby znak wodny oraz elementy odbijające światło ultrafioletowe.

4.3. ZASTOSOWANIA STEGANOGRAFII

Techniki steganograficzne pozwalają na realizowanie różnego rodzaju zadań. W związku z tym ze względu na obszary zastosowań i związane z nimi odmienne wymagania, metody steganograficzne można podzielić na cztery podstawowe grupy (Garbarczuk & Świć 2005)

- Anonimowa komunikacja – stosowana do ochrony komunikacji. Polega na ukryciu przesyłanych danych w dowolnym kontenerze. Pozwala to na ukrycie faktu komunikacji oraz również tożsamości komunikujących się. Nadawca przygotowuje stegokontener zawierający ukrytą informację a następnie anonimowo umieszcza go w publicznym miejscu, gdzie jest dostępny dla wszystkich. Najczęściej wykorzystuje się do tego celu Internet. Odbiorca pobiera zasób i odczytuje z niego dane. On również pozostaje anonimowy, zwłaszcza, że jest tylko jedną z wielu osób pobierających stegokontener. Jednak najważniejszym aspektem zabezpieczenia jest to, że dla osób postronnych rozpoznanie stegokontenera jest bardzo trudne, a niekiedy wręcz niemożliwe. Najistotniejszym wymaganiem Anonimowej komunikacji jest zachowanie przezroczystości, czyli braku zmian mogących wskazywać na to, że zasób umieszczony w miejscu publicznym jest stegokontenerem. Stegokontener musi być jak najbardziej podobny do oryginału,

tak by niemożliwe było wykrycie żadnych nieprawidłowości przy pomocy zmysłów czy też analizy komputerowej.

- Silne znakowanie wodne – stosowane do trwałego znakowania danych cyfrowych, najczęściej utworów muzycznych, grafiki lub filmów w celu umożliwienia późniejszego określenia własności intelektualnej czy praw autorskich. Silne znakowanie wodne polega na trwałym dołączeniu do oryginalnego utworu znacznika identyfikującego.

  Podstawowe wymogi stawiane znakom wodnym to odporność i przezroczystość. Znak wodny nie może ulegać zniszczeniu podczas przeprowadzania popularnych operacji przetwarzania sygnału, zmiany formatu, kompresji, filtracji, etc. Dobry znak wodny musi dać się odczytać dopóki sygnał posiada wartość, czyli do momentu jego znacznego uszkodzenia. Innymi słowy usunięcie znaku wodnego powinno wiązać się ze zniszczeniem znakowanych danych.

  Ponadto dołączenie znaku wodnego nie może zmniejszać wartości utworu, co mogłoby się stać, gdyby znakowanie wodne wprowadzało wykrywalny zmysłami poziom zakłóceń (Katzenbeisser & Petitcolas 2000).
- Słabe znakowanie wodne – odmiana znakowania wodnego mająca na celu potwierdzenie oryginalności nośnika. Znak wodny powinien zostać uszkodzony w przypadku wprowadzenia jakichkolwiek zmian do nośnika. Celem słabego znakowania wodnego jest potwierdzenie autentyczności danych tak, aby odbiorca mógł mieć pewność, że oznakowane dane nie zostały modyfikowane po ich oznakowaniu. Wiąże się to z koniecznością stosowania znaków ulegających zniszczeniu podczas każdej modyfikacji sygnału (Katzenbeisser & Petitcolas 2000).

- Ochrona przed powielaniem – jest dosyć specyficznym zastosowaniem steganografii mającym na celu zabezpieczenie nośników optycznych przed powielaniem. W tym przypadku zazwyczaj stosuje się specjalny zapis danych na nośniku, który nie może zostać poprawnie skopiowany przez dostępne na rynku urządzenia kopiujące.
- Stosowane są takie rozwiązania, jak wprowadzanie celowych uszkodzeń do zapisu, stosowanie niewłaściwej struktury nośnika, umieszczanie znaczników końca dysku wewnątrz danych, przerwy w zapisie czy też umieszczanie części danych w ukrytych ścieżkach. Najczęściej jednak kolejne generacje urządzeń kopiujących są w stanie wykonać kopię dysku zawierającego zabezpieczenia używane przed wprowadzeniem urządzenia na rynek (Kozieł 2006).

4.4. NOŚNIKI UKRYWANEJ INFORMACJI

Aby możliwe było ukrycie informacji niezbędny jest inny obiekt, w którym można ukryć dodatkową informację. W przypadku steganografii analogowej nie było to problemem, gdyż zawsze istniał jakiś nośnik fizyczny, który był do tego celu wykorzystywany. W przypadku steganografii cyfrowej, dane są niezależne od nośnika. Rolę kontenera może pełnić w tym wypadku jedynie inny zbiór danych cyfrowych. Może to być dowolny rodzaj danych. Jednak najlepiej do tego celu nadają się sygnały multimedialne. Spowodowane jest to:

- Łatwością modyfikacji.
- Trudnością przewidzenia wartości kolejnych elementów sygnału – niemożliwe jest określenie wartości kolejnej próbki na podstawie analizy wartości poprzednich próbek.

- Możliwością wprowadzenia zmian bez zniszczenia sygnału – zmiana wartości niektórych bitów wywiera jedynie nieznaczny wpływ na ostateczną postać sygnału multimedialnego.

Pod pojęciem sygnału multimedialnego rozumiemy dowolny rodzaj danych multimedialnych takich, jak: dźwięk, film lub obraz. Nie jest istotne czy jest to plik, czy też ciągły strumień danych taki, jak transmisja radiowa czy telewizyjna.

Każdy rodzaj danych multimedialnych ma swój format oraz zasady reprezentacji jego poszczególnych składników. Aby z nich korzystać należy znać strukturę oraz możliwości wprowadzania modyfikacji bez niszczenia sygnału (Kozieł 2005b).

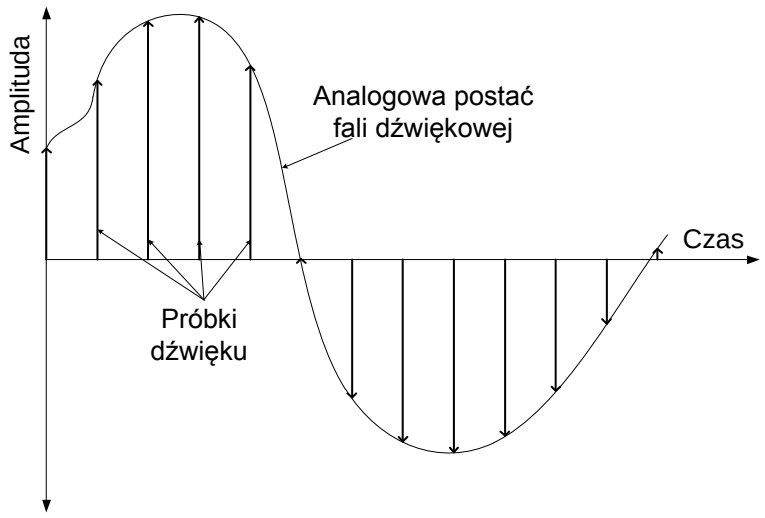
4.4.1. SYGNAŁ DŹWIĘKOWY

Sygnał dźwiękowy w swej analogowej postaci jest falą rozchodzącą się w otoczeniu. Zapis w postaci cyfrowej polega na zapisaniu wartości amplitudy fali dźwiękowej w poszczególnych chwilach czasu. Aby tego dokonać należy sygnał spróbkować. Próbkowanie polega na określaniu wartości amplitudy sygnału w danej chwili czasu. Charakteryzuje się również określoną rozdzielczością. Rozdzielczość jest zależna od liczby bitów przeznaczonych na zapisanie wartości amplitudy – jest to tak zwana rozdzielczość bitowa próbki. Próbką nazywamy jeden zapis wartości amplitudy dźwięku wykonany w określonym momencie.

Próbkowanie odbywa się z ustaloną częstotliwością. Częstotliwość oznacza liczbę próbek dźwięku zapisywanych w ciągu jednej sekundy. Dla przykładu dźwięk zapisywany na płytach audio CD jest próbkowany z rozdzielczością 44,1kHz. Oznacza to, że w ciągu jednej sekundy zostało zapisanych 44100 próbek dźwięku. Typowa rozdzielczość bitowa

próbki dźwięku to 16 bitów. Oznacza to, że każda z próbek może przyjmować jedną z 65536 wartości możliwych do zapisania na 16 bitach.

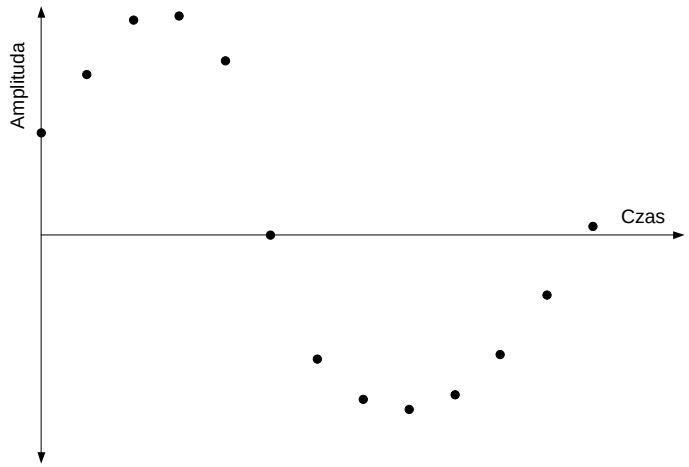
W praktyce do cyfrowego zapisu dźwięku używany jest zakres wartości <-1, 1>. Graficznie proces próbkowania został przedstawiony na rysunku 4.3.



Rys. 4.3. Próbkowanie dźwięku
(źródło: Kozieł 2005b)

W wyniku próbkowania otrzymujemy ciąg wartości reprezentujących sygnał audio w dyskretnych chwilach czasu.
Na podstawie tych wartości aproksymowany jest rzeczywisty przebieg sygnału podczas jego odtwarzania.

Przykład cyfrowego zapisu sygnału przedstawionego na rysunku 4.3 został zaprezentowany w sposób graficzny na rysunku 4.4.



Rys. 4.4. Dźwięk spróbkowany
(źródło: opracowanie własne na podstawie Kozieł 2005b)

Jak łatwo wywnioskować podczas analizy rysunku 4.4, większa liczba próbek przekłada się na dokładniejsze odwzorowanie sygnału – aproksymacja przebiegu analogowego sygnału pozwala uzyskać postać bardziej zbliżoną do rzeczywistej.

Podobnie ma się rzecz z rozdzielczością bitową próbki. Im jest większa, tym dokładniej można zapisać wartość próbki (błąd kwantyzacji jest mniejszy). Jednak niewielki błąd nie wpływa znacząco na jakość dźwięku. Zjawisko to wykorzystane zostało do ukrywania dodatkowych danych. Ukrywane są one poprzez wprowadzania do sygnału zmian, które w niewielkim stopniu wpływają na jego jakość.

4.4.2. OBRAZ

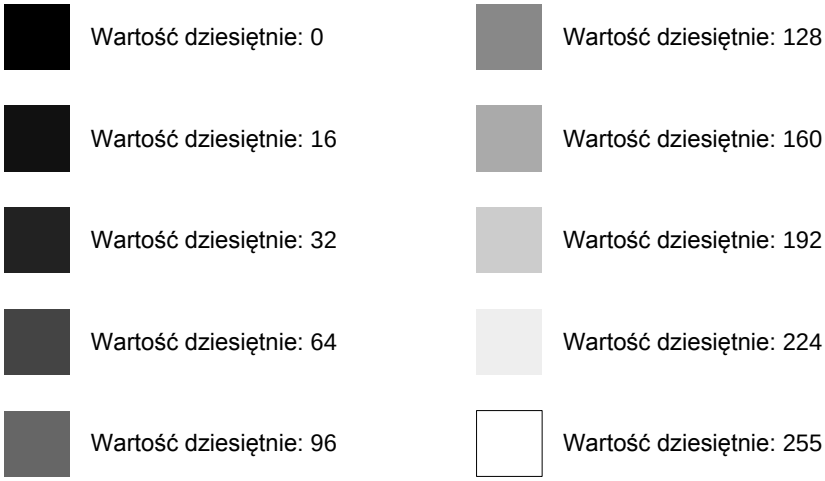
Zapis obrazu w postaci rastrowej doskonale nadaje się do potrzeb steganografii. Ten typ zapisu polega na reprezentowaniu obrazu przez macierz o rozmiarach równych wymiarom obrazu podanym w pikselach (piksel jest najmniejszym, niepodzielnym punktem obrazu).

W każdym elemencie macierzy zapisana jest informacja o jednym pikselu. Jej zawartość zależna jest od zastosowanego formatu pliku. Najczęściej zawiera określenie koloru piksela oraz dodatkowe informacje. Z punktu widzenia steganografii istotny jest zapis koloru.

Kolor zapisywany jest w postaci wartości liczbowej na określonej liczbie bitów. W przypadku obrazów czarno białych jest to tylko jeden bit. Zmiana jego wartości zmienia kolor piksela z czarnego na biały lub na odwrót. Jednak wykorzystując osiem bitów do zapisania barwy możemy uzyskać skalę szarości.

Będzie ona zawierała 256 różnych odcieni, gdyż tyle różnych wartości można zapisać na ośmiu bitach: wartość zero będzie odpowiadała kolorowi czarnemu. Kolor biały będzie reprezentowany przez wartość 255. Pozostałe pośrednie wartości będą opisywały różne odcienie szarości, według zasady: im mniejsza wartość tym ciemniejszy kolor.

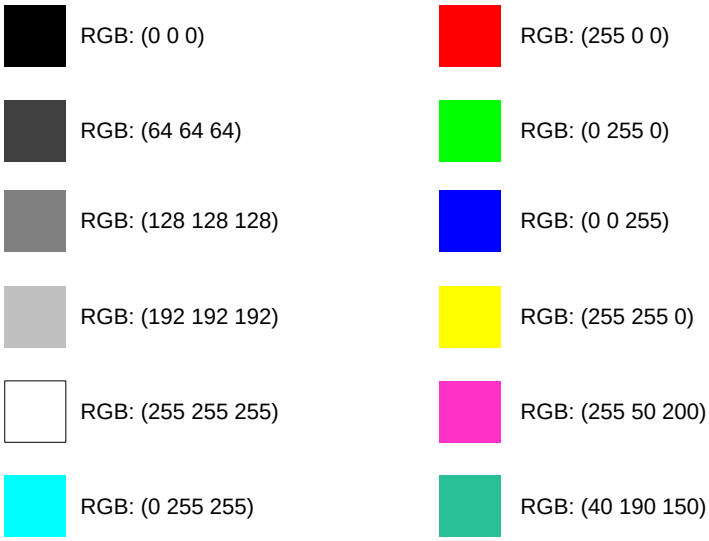
Przykładowe wartości pikseli oraz odpowiadające im barwy zostały zaprezentowane na rysunku 4.5.



Rys. 4.5. Kolory w ośmiobitowej skali szarości
(źródło: opracowanie własne)

Podobnie przedstawia się sytuacja w przypadku zapisywania obrazów kolorowych. Z tym, że kolor piksela jest mieszanina trzech kolorów podstawowych. Każdy z tych kolorów jest reprezentowany oddzielnie. Finalna barwa piksela jest składową wszystkich kolorów. Mieszanie kolorów wykonywane jest analogicznie do mieszania barw światła.

Zmieszanie wszystkich składowych koloru (czerwonej, zielonej i niebieskiej) w jednakowych proporcjach pozwala uzyskać kolor szary. Wszelkie inne proporcje dają kolory z całej palety barw, tak jak to przedstawiono na rysunku 4.6. Każda składowa koloru reprezentowana jest przez wartość liczbową. Im jest ona wyższa, tym większy jest udział danego koloru w ostatecznej barwie piksela. Maksymalna wartość liczbowa określająca udział koloru określona jest liczbą bitów przeznaczonych na zapisanie wartości koloru piksela (nazywana również głębią barw). Najczęściej każdy z kolorów zapisywany jest na ośmiu bitach, co daje łącznie 24 bity na zapisanie barwy całego piksela.



Rys. 4.6. Kolory w zapisie RGB

(źródło: opracowanie własne)

4.4.3. INNE NOŚNIKI

Jakkolwiek obraz i dźwięk są najbardziej popularnymi kontenerami wykorzystywanymi w steganografii, to możliwe jest użycie wielu innych nośników do ukrycia informacji.

Popularnym rozwiązaniem, zwłaszcza w zastosowaniach tajnej komunikacji jest wykorzystywanie pakietów IP.

Każdy pakiet posiada nagłówek zawierający istotne informacje o pakiecie danych przesyłanych poprzez sieć. Jednak nagłówki pakietów zostały zaprojektowane tak, aby były uniwersalne. W związku z tym niektóre pola nagłówków przeważnie nie są wykorzystywane. Stwarza to możliwość umieszczenia tam innych danych, które zostaną dostarczone do odbiorcy wraz z pakietem (Kundur, Ahsan, 2003).

Innym rozwiązaniem możliwym do zastosowania w lokalnych sieciach bezprzewodowych jest wykorzystanie do przenoszenia ukrytej informacji ramek zawierających błędną sumę kontrolną. W ten sposób w sieci bezprzewodowej tworzone jest pasmo dostępne do transmisji ukrytej informacji (Szczypiorski 2006).

W praktyce do przesłania ukrytych danych można wykorzystać dowolny kanał komunikacyjny pozwalający na umieszczenie w nim dodatkowych danych, bez niszczenia przesyłanej informacji. Najczęściej będą to nieużywane fragmenty danych, które zostaną podmienione na inną, sekretną zawartość.

Ciekawą metodą zaprezentowaną w (Mazurczyk, Szczypiorski, 2008) jest metoda celowego opóźniania pakietów w transmisji VoIP. Opóźnione pakiety nie są włączane do odtwarzanego strumienia dźwięku, gdyż docierają zbyt późno, niemniej jednak nie są gubione, więc mogą z powodzeniem zostać użyte do przeniesienia ukrytej informacji.

Dowolne dane mogą zostać zmodyfikowane, tak by dołączyć informację steganograficzną. Nie jest istotny sposób ich modyfikacji. Ważne jest natomiast by możliwe było wprowadzenie modyfikacji różniących się pomiędzy sobą lub też wykrycie miejsc, gdzie modyfikacje mogły być wprowadzone i stwierdzenie, czy zostało to wykonane.

W ten sposób uzyskujemy możliwość ukrycia dowolnej informacji binarnej – różniące się sposoby modyfikacji mogą odpowiadać wartościom zero oraz jeden. Podobnie jest w przypadku, gdy możliwe jest wykrycie miejsc, w których modyfikacje mogły zostać wprowadzone. Wówczas wprowadzenie modyfikacji może być równoznaczne z zakodowaniem binarnej jedynki. Rezygnacja ze zmian odpowiadać będzie wówczas za zakodowanie binarnego zera.

4.5. RODZAJE KONTENERÓW

Poznając nośniki ukrytej informacji należy zdać sobie sprawę z konsekwencji, jakie niesie za sobą zmiana kontenera. Rozróżniamy bowiem dwa rodzaje kontenerów:

- Ciągłe,
- O ograniczonej pojemności.

Najpopularniejsze są kontenery o ograniczonej pojemności. Mianem tym określamy wszystkie nośniki, które mają skończoną objętość. Będą to przede wszystkim pliki, nagrania zapisane na różnego rodzaju nośnikach czy zbiory danych. Ich charakterystyczną cechą jest możliwość precyzyjnego określenia początku, końca oraz rozmiaru. Wynika stąd fakt, że możliwe jest łatwe określenie miejsca, w którym rozpoczynają się dołączone dane (poprzez określenie odległości od początku lub końca kontenera). Jednocześnie istnieje ograniczenie, co do maksymalnego rozmiaru ukrywanych danych.

Nośniki o nieograniczonej pojemności – nazywane też kontenerami ciągłymi – to strumienie transmisji ciągłej. Mianem tym określamy wszystkie transmisje trwające z założenia przez nieokreślony czas. Przykładami mogą być tu transmisje radiowe lub telewizyjne.

Zakładamy, że taka transmisja nie posiada początku ani końca. Oczywiście jest to tylko bardzo ogólne przybliżenie, jednak sprawdzające się z punktu widzenia steganografii. Długość kontenerów ciągłych jest bowiem o wiele większa od długości dołączanej wiadomości. Zakłada się, że kontener ciągły ma nieograniczoną pojemność. Konieczne jest jednak oznaczanie początku i końca ukrytej wiadomości, pozwalające na jej zlokalizowanie. Odbiorca nie zna miejsca dołączenia danych. Musi więc w sposób ciągły monitorować sygnał kontenera.

Przedstawiony podział wymusza stosowanie odmiennych metod dla każdego z wymienionych rodzajów kontenerów. Definiuje również określone różnice, zarówno w podejściu do ukrywania, jak i możliwych do uzyskania efektach. Porównanie przedstawionych rodzajów kontenerów zaprezentowane zostało w tabeli 4.1.

Tabela 4.1 Porównanie kontenerów ciągłych oraz o ograniczonej pojemności

Cecha	Kontener ciągły	Kontener o ograniczonej pojemności
Pojemność	Nieograniczona	Ograniczona, konieczne sprawdzenie, czy informacja zmieści się w kontenerze
Rozpoznawanie miejsca ukrycia danych	Niekonieczne, dane mogą być dołączane od pierwszego bitu kontenera	Niezbędne, początek dołączanych danych znajduje się w losowym miejscu kontenera
Ukrywanie danych	Konieczne w czasie rzeczywistym	Niezależne od czasu
Odczyt danych	Znajdowanie znacznika początku ukrytych danych konieczne w czasie rzeczywistym, odczyt może trwać dłużej	Niezależny od czasu
Konieczność ciągłego monitorowania kontenera	Tak, kontener tylko w pewnych okresach czasu zawiera ukryte dane	Nie, odbiorca posiada informację o tym, czy kontener zawiera ukryte dane

(źródło: opracowanie własne)

4.6. PRZEGLĄD METOD STEGANOGRAFICZNYCH

Steganografia jest prężnie rozwijającą się dziedziną nauki. Zostały opracowane różnorodne metody steganograficzne. Wiele z nich posiada liczne modyfikacje.

Metody steganograficzne ze względu na sposób ukrywania dodatkowej informacji możemy podzielić na kilka grup (Garbarczuk & Kopniak 2005):

- Metody substytucji – ich działanie polega na podmienianiu wartości bitów kontenera wartościami bitów ukrywanych danych. Najczęściej podmieniane są wartości przechowujące nadmiarowe dane lub ich zakresy niewpływające na postrzeganie danych kontenera. Przykładem mogą być wysokie częstotliwości dźwięku czy też najmniej znaczące bity pikseli obrazu czy próbek dźwięku.
- Metody transformacyjne – ukrywające dane w dziedzinie wybranej transformaty. Dane kontenera (obraz lub dźwięk) są najpierw przekształcane przy pomocy wybranej transformaty, w wyniku czego otrzymywany jest zestaw współczynników. Współczynniki te są następnie modyfikowane w celu ukrycia w nich dodatkowych danych. Po ich wprowadzeniu wykonywana jest odwrotna transformata pozwalająca na ponowne uzyskanie reprezentacji sygnału w dziedzinie czasu.
- Metody rozproszonego widma – rozpraszające ukrywaną informację w całym spektrum częstotliwości sygnału kontenera. Rozproszenie danych w różnych zakresach częstotliwości pozwala na uzyskanie wysokiej odporności na uszkodzenia, jak również na skuteczne ukrycie dołączanych danych. Dane dołączone tą metodą są odporne na odfiltrowywanie pojedynczych częstotliwości. Aby zniszczyć dołączone dane często konieczne jest znaczne uszkodzenie kontenera.

- Metody statystyczne – ukrywające dane poprzez wprowadzanie modyfikacji wartości wybranych wartości statystycznych sygnału. Najczęściej modyfikowane są statystyki określonych fragmentów sygnału. Modyfikowane mogą być dowolne cechy statystyczne. Jednymi z najlepiej znanych przykładów są: modyfikacja liczby pikseli o określonym kolorze w zdefiniowanych obszarach obrazu oraz modyfikacja liczby próbek dźwięku o określonej amplitudzie.
- Metody zniekształceniowe – wprowadzające do danych kontenera określone zniekształcenia odpowiadające ukrywanym wartościom binarnym. Poważną wadą metod zniekształceniowych jest konieczność posiadania oryginalnego kontenera podczas odczytu danych. Proces ten odbywa się bowiem poprzez porównanie danych zmodyfikowanych z oryginalnymi.
- Metody generacji nośnika – polegające na generowaniu danych stegokontenera na podstawie ukrywanej informacji. Ukrywający dysponuje jedynie danymi do ukrycia. Na ich podstawie tworzy stegokontener w taki sposób, aby jak najlepiej ukrywa dane oraz jak najwierniej przypominał rzeczywiste dane.

W ramach tego rozdziału przedstawione zostaną popularne metody steganograficzne używane w typowych zastosowaniach takich jak znakowanie wodne czy tajna komunikacja.

4.6.1. METODA NAJMNIEJ ZNACZĄCYCH BITÓW

Jedną z pierwszych metod steganograficznych była metoda najmniej znaczących bitów (ang. *Least Significant Bit*) oznaczana często symbolem LSB. Metoda ta wykorzystuje najmniej znaczące bity próbek sygnału do ukrycia dodatkowych danych (Garbarczuk, Kopniak 2005).

Każda próbka zawiera informację określającą pewną cechę sygnału w danym punkcie. W przypadku dźwięku będzie to amplituda. W obrazie będzie to kolor piksela. Cecha sygnału określona jest za pomocą wartości liczbowej, która zapisana jest na określonej liczbie bitów. Każdy bit reprezentuje inną potęgą liczby dwa. Wartości bitów w liczbie zapisanej na ośmiu bitach prezentuje tabela 4.2.

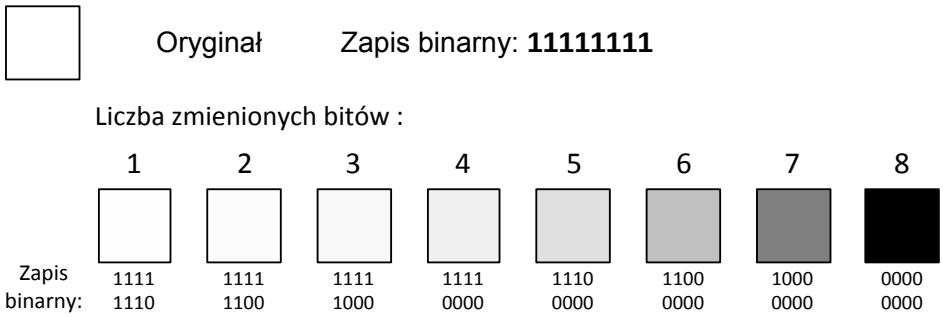
Tabela 4.2 Wartości bitów w zapisie wartości próbki

Numer bitu	8	7	6	5	4	3	2	1
Wartość bitu	128	64	32	16	8	4	2	1

(źródło: opracowanie własne)

Każdy z bitów opisujących próbkę może być zmodyfikowany. Jednak modyfikacja bardziej znaczących ma o wiele większy wpływ na zmianę ostatecznej wartości próbki niż modyfikacja najmniej znaczących bitów. Przykładowo: modyfikacja bitu o numerze 8 spowoduje zmianę wartości próbki o wartość 128, podczas gdy modyfikacja wartości bitu o numerze 1 zmieni wartość próbki jedynie o jeden. Stwarza to możliwość zmodyfikowania wartości niektórych bitów bez wprowadzania znacznych zmian wartości próbki. Najmniej znaczący bit każdej próbki może zostać zmodyfikowany w niemalże każdym sygnale. Nie spowoduje to wprowadzenia postrzegalnych zmian w sygnale. Najmniej znaczący bit każdej próbki przenosi najczęściej szum kwantyzacji.

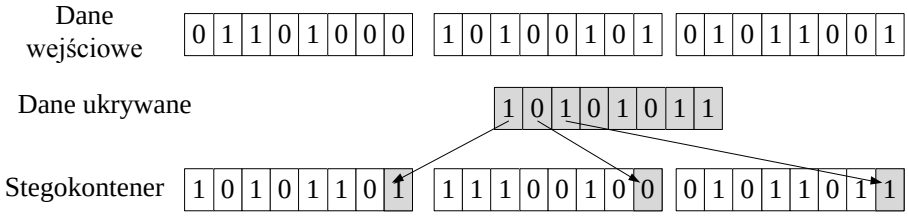
Wpływ liczby bitów, których wartości zostały zmienione na przeciwne został zaprezentowany na rysunku 4.7. Przykład przedstawiony na tym rysunku dotyczy barwy pikseli w obrazie zapisanym w skali szarości, gdzie do zapisania barwy piksela użyto 8 bitów.



Rys. 4.7. Wpływ liczby zmodyfikowanych bitów na barwę piksela
(źródło: opracowanie własne)

Ukrywanie informacji metodą najmniej znaczących bitów polega na zastępowaniu wartości najmniej znaczących bitów poszczególnych próbek wartościami bitów ukrywanej informacji.

W celu zwiększenia pojemności steganograficznej wykorzystywana jest większa liczba najmniej znaczących bitów każdej próbki, czyli podmieniane jest *n* najmniej znaczących bitów. Oczywiście im więcej bitów zostanie podmienionych, tym większa będzie utrata jakości sygnału kontenera. Zasada ukrywania danych metodą najmniej znaczących bitów została zaprezentowana na rysunku 4.8. Na rysunku 4.9 przedstawiono wpływ liczby najmniej znaczących bitów wykorzystanych do ukrycia dodatkowych danych, na jakość wynikowego stegokontenera.



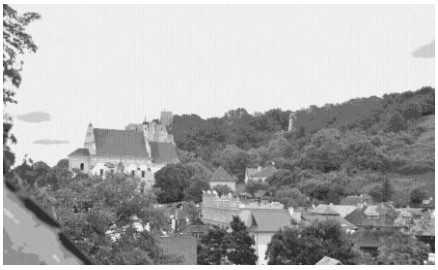
Rys. 4.8. Dołączanie danych metodą najmniej znaczących bitów
(źródło: opracowanie własne)

1

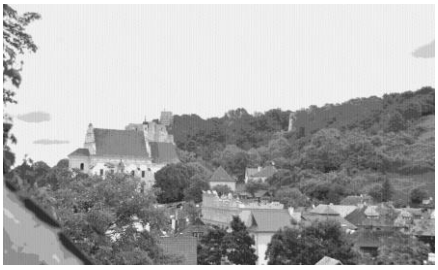
2

3

4



5



6

7

Obraz oryginalny

Rys. 4.9. Porównanie zniekształceń wprowadzanych przez metodę LSB, pod poszczególnymi obrazami umieszczono liczbę zmodyfikowanych bitów w każdym pikselu (źródło: opracowanie własne)

Analiza rysunku 4.9. pozwala zauważyć, że wraz ze wzrostem liczby modyfikowanych bitów w każdym pikselu obrazu następuje utrata jakości obrazu – zniekształcenia stają się coraz bardziej widzialne. Można zaobserwować tu analogię do kompresji stratnej, nie bez powodu zresztą – im więcej najmniej znaczących bitów wykorzystywanych jest do ukrycia dodatkowej informacji, tym więcej informacji o oryginalnym obrazie jest tracone. Identycznie jak w przypadku kompresji – silniejsza kompresja wiąże się z większą utratą informacji o obrazie.

Nie oznacza to jednak, że każdy obraz będzie jednakowo wrażliwy na zniekształcenia wprowadzane przez metodę najmniej znaczących bitów. Wynika to z niedoskonałości ludzkiego oka, które dobrze sobie radzi z odróżnianiem szczegółów w jednorodnych obrazach, ale o wiele gorzej postrzega zmiany w obrazach dynamicznych, zawierających wielobarwne, nieregularne obiekty.

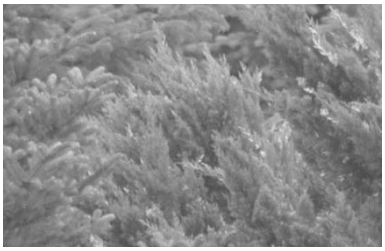
Przykład tego zjawiska został zaprezentowany na rysunku 4.10., gdzie zostały porównane dwa różne obrazy, w których ukryto tę samą informację z wykorzystaniem takiej samej liczby najmniej znaczących bitów. Na rysunku 4.10 w lewej kolumnie przedstawiono wyniki ukrycia informacji w obrazie dynamicznym (fragment fotografii), zaś w prawej kolumnie umieszczone zostały przekształcone wersje rysunku czarno-białego. Jak widać ukrywanie danych w obrazie dynamicznym pozwala zachować przezroczystość nawet przy wykorzystaniu czterech najmniej znaczących bitów do ukrycia informacji. W przypadku obrazu czarno-białego przy wykorzystaniu tej samej liczby najmniej znaczących bitów wyraźnie daje się zauważyć zmiana koloru tła z białego na szary.

Warto zauważyć, że wykorzystując cztery bity każdego piksela modyfikujemy połowę wszystkich bitów obrazu. Oznacza to, iż w obrazie o rozmiarze 2kB możemy ukryć aż 1kB dodatkowych danych.

Zmodyfikowanie pięciu najmniej znaczących bitów wprowadza zauważalne zniekształcenia w obrazie dynamicznym. Daje się zauważyć redukcja szczegółowości obrazu. W obrazie czarno-białym widoczna jest zmiana barw.

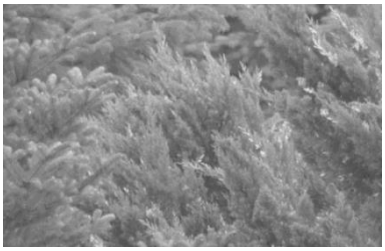
Ciekawe zjawisko można zaobserwować podczas modyfikacji sześciu najmniej znaczących bitów: obraz dynamiczny zupełnie traci jakość, podczas gdy obraz czarno-biały nadal pozostaje wyraźny, pomimo znacznej zmiany barw – powodem tego jest zastosowanie barw o dużym kontraście. W obrazie dynamicznym zaś zastosowano zbliżone do siebie barwy, przez co większość informacji zakodowana została w środkowych bitach każdej próbki. W obrazie czarno-białym różnice koloru poszczególnych pikseli zakodowane zostały na wszystkich bitach, przy czym największą wagę mają najbardziej znaczące bity.

Nie oznacza to bynajmniej, że pomiędzy poszczególnymi pikselami obrazu nie występują różnice barwy: na wydruku nie będą one widoczne, ze względu na znaczące pomniejszenie zdjęć oraz ograniczoną rozdzielczość druku, jednak pod dokonaniu bliższej analizy powiększonego obrazu zauważalne stanie się znaczne zróżnicowanie barw poszczególnych pikseli. Można to zaobserwować na rysunku 4.10, gdzie przedstawiono w dużym powiększeniu fragment tła obrazu czarno-białego, zapisanego jako obraz w skali szarości, po ukryciu w nim dodatkowych danych z wykorzystaniem sześciu najmniej znaczących bitów każdej próbki. Zastosowane powiększenie pozwala na przedstawienie każdego piksela w postaci kwadratu.



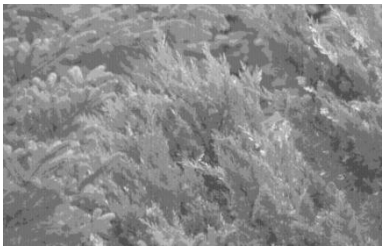
2

2



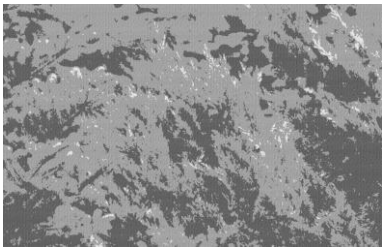
4

4



5

5



6

6

Rys. 4.10. Porównanie widoczności zniekształceń wprowadzanych przez metodę LSB w różnego typu obrazach (pod poszczególnymi obrazami umieszczono liczbę zmodyfikowanych bitów w każdym pikselu)

(źródło: opracowanie własne)

Rys. 4.11. Porównanie zmian wprowadzonych w pikselach barwy białej, po podmienieniu wartości sześciu najmniej znaczących bitów każdego piksela na wartości bitów ukrywanej informacji – każdy piksel reprezentowany jest na rysunku przez kwadrat (źródło: opracowanie własne)

Jak wspomniano wcześniej, kontrast pomiędzy poszczególnymi elementami obrazu ma istotny wpływ na widoczność zmian wprowadzanych przez metodę LSB (jak przestawiono na rysunku 4.12): im większy jest kontrast pomiędzy poszczególnymi barwami, tym więcej bitów należy zmodyfikować, by utracić różnicę pomiędzy tymi barwami. Innymi słowy: im bardziej zbliżone są do siebie barwy, tym więcej najbardziej znaczących bitów opisujących te barwy posiada takie same wartości. Jest to jednoznaczne z tym, że różnice pomiędzy barwami są przechowywane wówczas przez mniej znaczące bity i jeżeli je nadpiszemy, informacja o różnicach zostanie zniszczona, zaś jedyne różnice barw będą wówczas wynikały z różnic pomiędzy wartościami wprowadzonymi przez metodę steganograficzną.

Rys. 4.12. Porównanie wpływu kontrastu w obrazach oryginalnych na utratę szczegółów obrazu podczas ukrywania danych metodą najmniej znaczących bitów (źródło: opracowanie własne)

Jak możemy łatwo zauważyć na rysunku 4.11, przedstawiona analiza pozwala na wyciągnięcie wniosku, iż najmniej widoczne są zmiany wprowadzane do obrazów dynamicznych o dużym kontraście. Takie obrazy najlepiej nadają się do celów steganografii.

Podobnie przedstawia się sytuacja, jeżeli mamy do czynienia z dźwiękiem. Sygnały głośne o dużej dynamice, zawierające dużo różnorodnych dźwięków, pozwalają na uzyskanie większej przezroczystości niż sygnały stonowane, zawierające tylko jeden rodzaj dźwięku. Ludzki zmysł słuchu jest jednak o wiele bardziej czuły niż zmysł wzroku, dlatego też o wiele łatwiej wychwyci wprowadzone zmiany. Skutkuje to koniecznością zmniejszenia liczby modyfikowanych bitów lub zastosowaniem dodatkowych zjawisk pozwalających na ukrycie wprowadzonych zmian, takich jak maskowanie.

4.6.2. METODA DOŁĄCZANIA ECHA

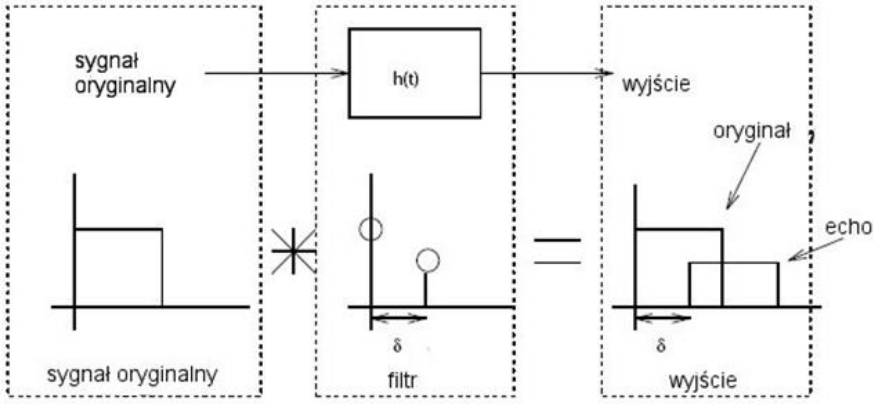
Do ukrycia informacji może służyć dowolne zjawisko, pozwalające na wprowadzenie modyfikacji do kontenera w sposób niewprowadzający postrzegalnych zakłóceń. Echo jest takim zjawiskiem, występującym w sygnałach dźwiękowych. Słuch nie jest w stanie wychwycić dołączonego echa sygnału, jeżeli występuje ono nie później niż 2 milisekundy po sygnale, pod warunkiem, że jego amplituda nie przekracza połowy amplitudy sygnału oryginalnego (Bender et al., 1996).

Możliwe jest więc wykorzystanie tego zjawiska do ukrycia dodatkowej informacji. Do ukrywania danych wykorzystywany jest sterowalny filtr opóźniający, którego praca polega na dodaniu do oryginalnego sygnału jego kopii opóźnionej o zadany czas. Procedura dodawania echa została przedstawiona na rysunku 4.13.

Na rysunku 4.13 symbolem δ oznaczono wartość opóźnienia echa względem sygnału. Wartość ta używana jest do określenia ukrywanego symbolu. Najczęściej ukrywane są wartości zero oraz jeden. Przypisuje się im określone wartości opóźnienia echa, tak jak to zostało zaprezentowane na rysunku 4.14.

Następnie sygnał dzielony jest na fragmenty, do których dodane zostanie echo. Do każdego fragmentu dodawane jest echo o opóźnieniu odpowiadającym wartości, jaka ma zostać ukryta.

Zaznaczyć należy, że w każdym fragmencie ukrywany jest tylko jeden bit dodatkowej informacji.



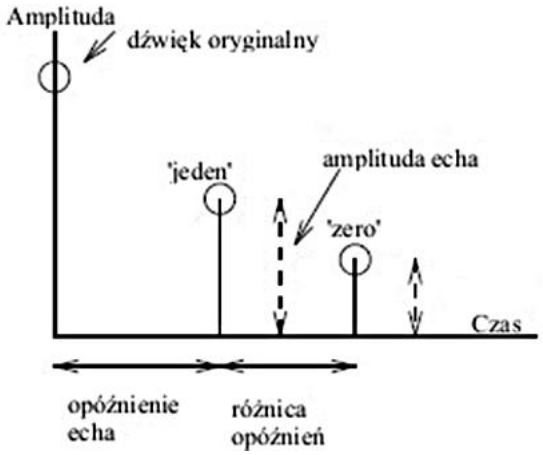
Rys. 4.13. Dołączanie echa do sygnału
(źródło: Dymarski 2006)

Dołączanie echa w praktyce realizowane jest na całym sygnale dźwiękowym jednocześnie: tworzone są bowiem dwa niezależne sygnały zawierające echo dołączone z różnym opóźnieniem. Opóźnienia zastosowane w tych sygnałach odpowiadają wartościom jeden oraz zero.

Następnie z tak przygotowanych fragmentów pobierane są fragmenty, które są umieszczane w sygnale stegokontenera.

Użycie fragmentów sygnału zawierających różne wartości opóźnienia echa wprowadza do sygnału słyszalne zakłócenia. Uniknięcie ich możliwe jest albo poprzez zastosowanie płynnej zmiany opóźnienia podczas przejścia z jednego fragmentu do drugiego lub też poprzez zmianę opóźnienia o wartość nie większą od 0.05 milisekundy, przy czym częstość zmian nie może przekroczyć 10 milisekund.

Warunki percepcji dołączonego sygnału echa zostały określone: największe opóźnienie pozwalające zachować niesłyszalność sygnału echa określone zostało na poziomie 1 - 2 milisekund przy amplitudzie nieprzekraczającej 50% amplitudy dźwięku oryginalnego (Dymarski et al., 2003; Pobłocki, 2003).



Rys. 4.14. Opóźnienia echa sygnału odpowiadające ukrytym wartościom zero oraz jeden (źródło: Dymarski 2006)

Odczyt ukrytej informacji odbywa się poprzez podzielenie sygnału na fragmenty, a następnie określenie wartości opóźnień zastosowanych w poszczególnych fragmentach. Wartość opóźnienia dołączenia echa określana jest poprzez obliczenie autokorelacji fragmentu sygnału.

Metoda dołączania echa posiada wiele modyfikacji, takich jak:

- Zastosowanie pre-echa, czyli sygnału echa wyprzedzającego dźwięk. Taki rodzaj echa wykorzystywany jest do kodowania wartości zero, podczas gdy echo następujące po sygnale używane jest od kodowania wartości jeden.
- Użycie echa o dodatniej i ujemnej polaryzacji – w tym przypadku każda z polaryzacji odpowiada wówczas za zakodowanie innej wartości (Harbarchuk, Kozieł, 2008).

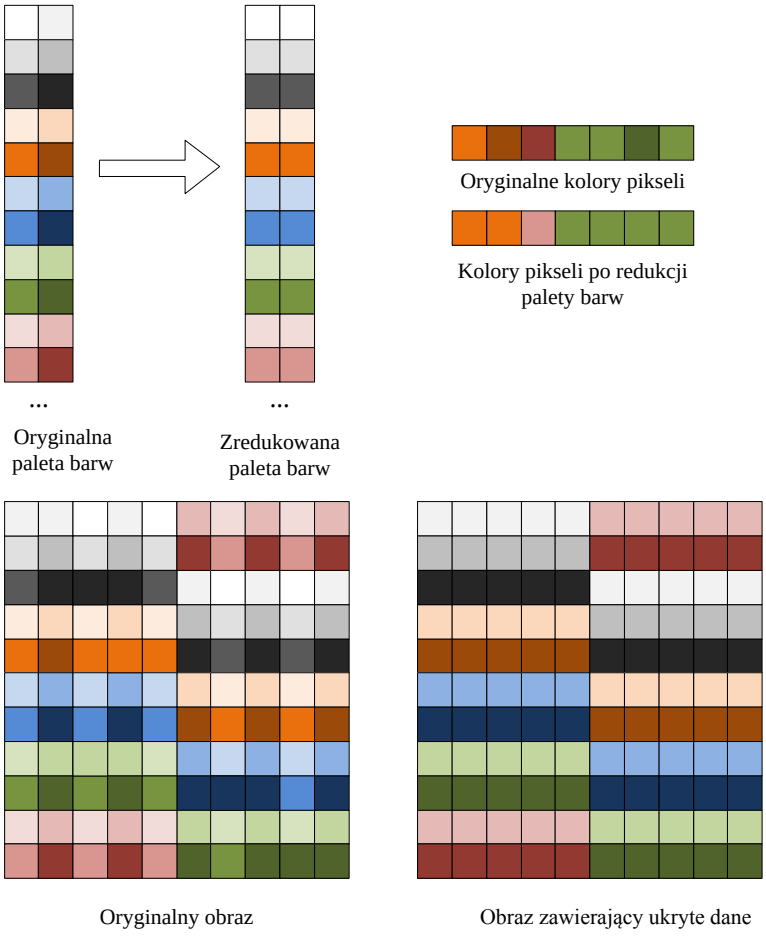
Metoda dołączania echa charakteryzuje się niewielką pojemnością steganograficzną i wysoką odpornością na uszkodzenia ukrytej informacji.

4.6.3. METODA MODYFIKACJI KOLORÓW INDEKSOWANYCH

Metoda ta stosowana jest do ukrywania informacji w obrazach posiadających paletę barw. Paleta barw stosowana jest w niektórych formatach zapisu obrazu do określania kolorów poszczególnych pikseli.
Zamiast zapisywania wartości koloru w każdym pikselu z osobna, tworzona jest paleta barw zawierająca wszystkie użyte w obrazie kolory. W pikselu umieszcza się natomiast informację, który element palety przechowuje jego barwę – wszystkie piksele o identycznej barwie będą wskazywać na jeden element palety barw, co pozwala na zredukowanie liczby bitów potrzebnych do zapisania kolorów wszystkich pikseli.

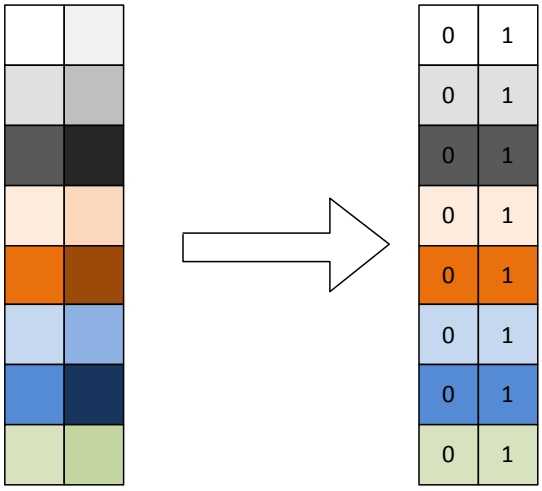
Metody steganograficzne ukrywające dane w tego typu obrazach wykorzystują operację redukcji palety barw. Polega ona na zmniejszeniu liczby kolorów w palecie. Najczęściej liczba ta zmniejszana jest o połowę. Rozmiar palety nie ulega jednak zmianie. Kolory w palecie umieszcza się wielokrotnie (w przypadku redukcji liczby kolorów o połowę – dwukrotnie). Otrzymujemy więc paletę o tym samym rozmiarze zawierającą po-

wtórzone kolory. Następnie przypisujemy pikselom kolory z nowej palety barw w taki sposób, aby każdy piksel miał przypisaną barwę jak najbardziej zbliżoną od pierwotnej. Jednak ze względu na powtórzenie kolorów w palecie, pikselowi możemy przypisać jeden z wielu elementów palety przechowujących ten sam kolor. Poprzez przypisanie określonego elementu palety kodowane są ukrywane wartości (Garbarczuk, Kopniak 2005). Przykład redukcji palety barw, przedstawiono na rysunku 4.15.



Rys. 4.15. Redukcja palety barw w metodzie kolorów indeksowanych (źródło: opracowanie własne)

Proces ukrywania danych w metodzie redukcji palety barw polega na zredukowaniu palety barw tak, jak to zostało przedstawione na rysunku 4.15. Po zredukowaniu palety każdy kolor, który w niej pozostał występuje więcej niż raz. W przedstawionym przykładzie palety każdy kolor występuje dwukrotnie, można więc każdemu z elementów palety posiadającemu tę samą barwę przypisać inną wartość binarną tak, jak zostało to przedstawione na rysunku 4.16.

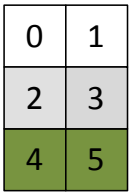


Rys. 4.16. Redukcja palety barw w metodzie kolorów indeksowanych wraz z zaznaczonymi wartościami binarnymi przypisanymi poszczególnym elementom palety barw
(źródło: opracowanie własne)

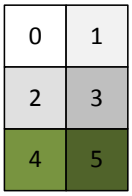
Jeżeli każdy z elementów palety barw opiszemy kolejną liczbą całkowitą oraz przypiszemy elementom palety o tej samej barwie odpowiadające im wartości binarne, to proces ukrywania informacji będzie polegał na przypisaniu przetwarzanemu pikselowi barwy wskazywanej przez

element palety barw posiadający przypisaną taką samą wartość binarną, jak wartość ukrywana.

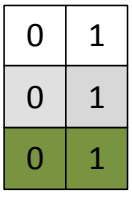
Przykład ukrywania informacji wykorzystującego fragment zredukowanej palety barw został przedstawiony na rysunku 4.17.



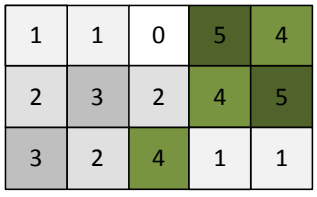
Liczby odpowiadające poszczególnym elementom zredukowanej palety barw



Liczby odpowiadające poszczególnym elementom oryginalnej palety barw



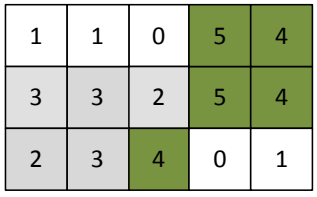
Wartości binarne przypisane do poszczególnych elementów zredukowanej palety barw



Oryginalny obraz wraz z przypisanymi do niego elementami oryginalnej palety barw

11010 11010 01001

Ciąg binarny do ukrycia



Zmodyfikowany obraz wraz z przypisanymi do niego elementami zredukowanej palety barw

Rys. 4.17. Proces ukrywania informacji metodą redukcji palety barw

(źródło: opracowanie własne)

4.6.4. METODA PROCENTOWA

Metoda procentowa używana jest do dołączania danych do obrazów czarno-białych. Jako, że zbiór możliwych barw jest ograniczony do dwóch wartości, to możliwe jest opisanie koloru pojedynczego piksela przy pomocy jednego bitu. Wartość „0" odpowiada za kolor czarny, zaś „1" za kolor biały (Garbarczuk & Kopniak 2005).

W takich obrazach niemożliwe jest zastosowanie metody najmniej znaczących bitów. Aby umożliwić dołączanie ukrytej informacji zaprojektowano metodę statystyczną, polegającą na dzieleniu obrazu na fragmenty o określonej wielkości.

Przewaga pikseli w kolorze białym w wydzielonym fragmencie obrazu jest jednoznaczna z zakodowaniem wartości binarnej „1". Przewaga punktów czarnych interpretowana jest jako ukryta wartość „0" (Garbarczuk & Kopniak 2005).

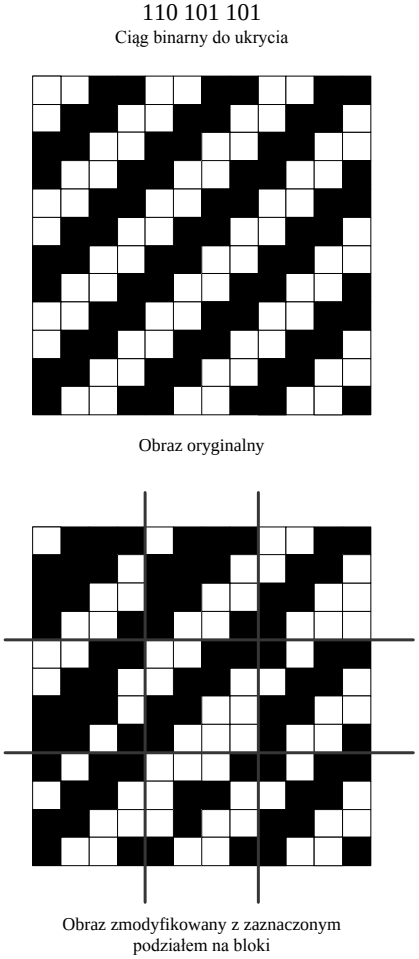
Obraz będący kontenerem metody procentowej dzielony jest na fragmenty (obszary) o określonym rozmiarze. Obszary są prostokątami, których wymiary zależne są od klucza steganograficznego. Po wydzieleniu są one sortowane pseudolosowo, a następnie do każdego z nich dołączana jest ukrywana wartość binarna poprzez zmodyfikowanie liczby pikseli czarnych i białych. Dokonuje się tego poprzez zmianę barwy pikseli położonych na granicy obszarów czarnych i białych. Jeżeli ukrycie informacji w danym fragmencie nie jest możliwe (na przykład wówczas, gdy liczba pikseli, które musiałyby zostać zmodyfikowane jest zbyt duża) fragment jest oznaczany jako nieużywany (Garbarczuk & Kopniak 2005).

Ze względu na możliwość przekłamań podczas transmisji czy zmiany formatu obrazu stosowany jest margines bezpieczeństwa λ. Określa on maksymalną procentową liczbę punktów, jakie mogą ulec przekłama-

niu. Ukrycie informacji przy uwzględnieniu parametru λ polega na modyfikacji liczby pikseli (P) w kolorze, który ma mieć więcej punktów w danym obszarze, tak by spełniała nierówność:

$$P>X+\lambda \qquad (4.1).$$

Przez X oznaczono całkowitą liczbę pikseli w analizowanym fragmencie obrazu.



Rys. 4.18. Proces ukrywania informacji metodą procentową
(źródło: opracowanie własne)

Jeżeli przyjmiemy, że binarnej „1” będzie odpowiadał obszar zawierający więcej czarnych pikseli, a binarnemu „0” obszar, w którym przeważają piksele białe, to dla metody procentowej używającej bloków o rozmiarze 4x4 piksele proces ukrywania będzie przebiegał tak, jak zostało to przedstawione na rysunku 4.18.

W przedstawionym przykładzie łatwo można zaobserwować wprowadzane zniekształcenia, ze względu na regularność wzoru występującego na obrazie. W przypadku rzeczywistych obrazów zniekształcenia będą znacznie mniej widoczne.

4.6.5. METODY BAZUJĄCE NA TRANSFORMACIE

W przedstawionych wcześniej metodach operacja ukrywania wykonywana była w dziedzinie czasu. Jest to naturalna dziedzina reprezentacji sygnału. Jednak użycie innej dziedziny reprezentacji niesie za sobą dodatkowe możliwości oraz pozwala uzyskać inne właściwości metody. Aby przejść do innej dziedziny reprezentacji sygnału, konieczne jest wykonanie określonego przekształcenia. Najczęściej jest to jedno z przekształceń (Garbarczuk & Kopniak 2005):

- transformata falkowa,
- transformata Fouriera,
- transformata cosinusowa,
- transformata Z.

Przy pomocy wybranego przekształcenia sygnał transformowany jest do dziedziny użytej transformaty. W wyniku uzyskuje się zbiór współczynników stanowiących zapis sygnału w określonej dziedzinie. Ukrycie informacji możliwe jest poprzez modyfikację wartości otrzymanych współczynników. Wykonanie odwrotnej transformaty na zmodyfikowanym

zestawie współczynników powoduje otrzymanie reprezentacji sygnału w dziedzinie czasu. Oczywiście jest ona nieco zmieniona względem oryginału, gdyż zawiera dołączoną informację. Metody transformacyjne pozwalają na ogół uzyskać lepszą odporność na uszkodzenia ukrytej informacji. Spowodowane jest to faktem, że informacja nie jest ukryta w jednym konkretnym miejscu, lecz rozproszona w całym kontenerze, który był przekształcony jako całość. Istotne jest tu zaznaczenie, że sygnał może być dzielony na fragmenty (bloki), które są przekształcane niezależnie. W takim wypadku ukrywana informacja rozpraszana jest tylko w obrębie przetwarzanego bloku, nie zaś w całym kontenerze.

Podział na bloki pozwala często na przyspieszenie działania metody, gdyż przetwarzanie mniejszych fragmentów sygnału wymaga użycia mniejszej ilości zasobów. Ponadto podział sygnału na bloki pozwala na stosowanie metod transformacyjnych w kontenerach ciągłych.
Dodatkową zaletą jest niezależne ukrywanie fragmentów informacji w poszczególnych blokach, więc jeżeli jeden z nich ulegnie uszkodzeniu nie wpłynie, to na poprawność odczytu ukrytej informacji z pozostałych.

Literatura

Bender W., Gruhl D., Morimoto N., Lu N. (1996) Techniques for data hiding, IBM system journal no. 5

Camou M., Goerzen J.,Van Couwenberghe A. (2000) Debian Linux Księga eksperta, Gliwice, Helion

DRBD (2012) What is DRBD. Pobrano 12.11.2012 z lokalizacji http://www.drbd.org/home/what-is-drbd..

Dymarski P., Pobłocki A., Baras C., Moreau N. (2003) Algorytmy znakowania wodnego sygnałów dźwiękowych, Krajowe Sympozjum Telekomunikacji, Bydgoszcz

Dymarski P. (2006) Filtracja sygnałów dźwiękowych jako metoda znakowania wodnego i steganografii, Bydgoszcz

Garbarczuk W., Świć A. (2005) Podstawy ochrony informacji, Lublin, Politechnika Lubelska

Garbarczuk W., Kopniak P. (2005) Steganologia: współczesne metody ochrony informacji (przegląd), Pomiary Automatyka Kontrola,. 3/2005

Harbarchuk V., Kozieł G. (2008) Review of Sound-based Steganographic techniques, Iskusstvennyj Intellekt, no. 4, s. 4-14

Katzenbeisser S., Petitcolas F. A. P. (2000) Information hiding techniques for steganography and digital watermarking, Artech House

Kozieł G. (2005), Cyfrowe znaki wodne w steganologii, w: Grzegórski S., Miłosz M., Muryjas P. (red.) *Varia informatica : obliczenia i technologie*, s. 229-235, Lublin, Polskie Towarzystwo Informatyczne

Kozieł G. (2005a), Cyfrowy zapis dźwięku w zadaniach steganograficznych, w: Miłosz M., *Bezpieczeństwo informacji : od teorii do praktyki*, s. 157-166, Warszawa, MIKOM, 2005

Kozieł G. (2006), Steganografia w zabezpieczeniach nośników multimedialnych, w: Miłosz M., Muryjas P. (red.) Varia Informatica – technologie i bezpieczeństwo, s. 193-200, Lublin, PTI

Kozieł G. (2011) Zmodyfikowane metody cyfrowego przetwarzania sygnałów dźwiękowych w steganografii komputerowej (praca doktorska), Lublin, Politechnika Lubelska

Kozieł G. (2011 a) Information security policy creating, w: Actual Problems of Economics No. 12(126) 2011, s.376-380

Kozieł G., Harbarchuk V. (2005), Digital Signature – tasks and problems, w: Актуальні Проблемы Економіки no.10(52)/2005

Krutul K. (2003) Nowy klaster TASK. Pobrano 12.11.2012, z lokalizacji http://www.frazpc.pl/artykuly/56951/Nowy_klaster_TASK

Kundur D., Ahsan K. (2003), Practical Internet Steganography: Data Hiding in IP, Proceedings of Texas Workshop on Security of Information Systems

Kwiecień A. et al. (2012), *Bezpieczeństwo Teleinformatyczne*. Pobrano 12.11.2012 z lokalizacji: http://pl.wikipedia.org/wiki/Bezpieczeństwo_teleinformatyczne

Linbit (2012). Pobrano 12.11.2012 z lokalizacji http://www.linbit.com/en/

Mazurczyk W., Szczypiorski K. (2008) Steganography of VoIP streams, OTM 2008, Part II – Lecture Notes in Computer Science (LNCS) 5332, Springer-Verlag Berlin Heidelberg

Michalski M. (2012) RAID programowy w systemie Linux (Debian). Pobrano 12.11.2012 z lokalizacji http://www.spy86.cba.pl/download/-artykuly/RAID programowy w systemie LINUX.pdf

Microsoft (2012), Typy kopii zapasowych. Pobrano 12.11.2012 z lokalizacji http://technet.microsoft.com/pl-pl/library/cc784306%28v=ws.10%29.aspx>

Pekao S.A.(2012). Przestrzegaj zasad bezpieczeństwa. Pobrano 12.11.2012 z lokalizacji http://www.pekao.com.pl/indywidualni/bankowosc_elektroniczna/Bezpieczenstwo/

Petitcolas F. A. P., Ross J., Kuhn G. (1999), Information Hiding—A Survey, Proceedings of the IEEE, special issue on protection of multimedia content, 87(7), pp. 1062–1078

Pobłocki A. (2003) Cyfrowe znakowanie wodne sygnałów dźwiękowych z wykorzystaniem echa, praca dypl., Politechnika Warszawska

Szczypiorski K. (2006) Steganografia w bezprzewodowych sieciach lokalnych (praca doktorska). Pobrano 12.11.2012, z lokalizacji http://krzysiek.tele.pw.edu.pl/ pdf/doktorat.pdf

Szychowiak M. (2006), Bezpieczeństwo systemów komputerowych, Pobrano 13.11.2012 z lokalizacji http://wazniak.mimuw.edu.pl/index.php?title=Bezpiecze%C5%84stwo_system%C3%B3w_komputerowych

tvn24.pl (2007) Wała tOmaszowi LiSowi. Pobrano 2.11.2012 z lokalizacji http://www.tvn24.pl/wiadomosci-z-kraju,3/wala-tomaszowi-lisowi,38170.html

Valkor (2012) Linux Partition HOWTO. Pobrano 13.11.2012 z lokalizacji http://www.tldp.org/HOWTO/Partition/fdisk_partitioning.html

What is Clonezilla (2012). Pobrano 12.11.2012 z lokalizacji http://clonezilla.org/

Wikipedia (2012), Computer cluster. Pobrano 13.11.2012 z lokalizacji http://en.wikipedia.org/wiki/Computer_cluster

Wikipedia (2012a), Beowulf. Pobrano 13.11.2012 z lokalizacji http://pl.wikipedia.org/wiki/Beowulf_%28informatyka%29

Zespół PLD (2012), PLD Linux Distribution, Podręcznik użytkownika, administratora i twórcy. Pobrano 12.11.2012 z lokalizacji http://pl.docs.pld-linux.org/pld_dok.html