Uniwersytet w Białymstoku

Wydział Matematyczno-Fizyczny

Instytut Matematyki

Krzysztof Rapczyński

Systemy kontroli dzienników systemowych serwera pracującego pod kontrolą Solaris

Praca dyplomowa napisana
pod kierunkiem
dr hab. Krzysztofa Prażmowskiego

Białystok 2003

Składam serdeczne podziękowania
mgr. Mariuszowi Żynelowi
za poświęcony czas i pomoc
w pisaniu tej pracy.

Krzysztof Rapczyński

Spis treści

Wstęp

Wraz z upowszechnieniem się Internetu pojawił się problem zabezpieczania systemów komputerowych oraz całych sieci komputerowych. W ostatnich latach powstało wiele publikacji na temat bezpieczeństwa [4, 5] i ochrony danych. Nie jest to podyktowane jakąś modą lecz praktyczną potrzebą.

Na początku lat 90-tych, kiedy powstawał Internet, kwestia zabezpieczania komputerów dotyczyła prawie wyłącznie komputerów wojskowych. Wiele usług, które powstały w tym czasie, jak na przykład Telnet czy FTP, nie maj wbudowanych żadnych mechanizmów szyfrowania, a identyfikacja użytkowników odbywa się wyłącznie na podstawie hasła.

Z zagadnieniem bezpieczeństwa systemów komputerowych ściśle związana jest analiza dzienników systemowych. Po samym zachowaniu systemu i zachowaniu oprogramowania, trudno jest określić, co tak naprawdę dzieje się na komputerze, czy nieupoważnione osoby oglądają poufne dane itp. Nieprawidłowości w działaniu systemu mogą świadczyć o destrukcyjnej działalności intruza, a w tej sytuacji jest na ogół za późno na podjęcie kroków zapobiegawczych. Analiza logów, wykonywana regularnie, wydatnie pomaga ustrzec się przed rezultatami ataku, wykryć zawczasu intruza oraz podjąć działania prewencyjne.

W niniejszej pracy ograniczyliśmy się tylko do tych dzienników, które powstają na serwerze `math`. Nie jest to poważne ograniczenie, wręcz przeciwnie. Na serwerze tym uruchomionych jest wiele usług i dopilnowanie wszystkich logów jest wyzwaniem. Pracę rozpoczynamy od ogólnego przeglądu wszystkich dzienników, które powstają na `math`, następnie przechodzimy do szczegółowej analizy informacji tam zgromadzonych. Trzy kryteria brane są pod uwagę podczas analizy:

- bezpieczeństwo,
- stabilność i wydajność,
- jakość usług.

Kwestia bezpieczeństwa i ostrzegania przed zagrożeniami ściśle związana jest z systemami IDS opisywanymi w rozdziale 2. Stabilność i wydajność systemu wiąże się z monitorowaniem różnych parametrów systemu. Rozdział 3 poświęcony jest oprogramowaniu MRTG, które temu właśnie służy. W rozdziale

4 przedstawiamy natomiast statystyki dotyczące jednych z najważniejszych usług na serwerze `math`, a mianowicie HTTP i FTP. Na podstawie statystyk można oceniać funkcjonalność usług i dostosowywać je do potrzeb użytkowników.

Doświadczenia zdobyte podczas analizy logów pozwoliły przygotować dokumentację na temat zabezpieczeń systemów komputerowych. Pełny tekst w fomacie PDF znajduje się pod adresem:

`http://theta.uwb.edu.pl/projects/loganalysis/bezpsk.pdf`

Rozdział 1

Dzienniki systemowe

Dzienniki systemowe inaczej nazywane logami są jednym z najważniejszych, z punktu widzenia bezpieczeństwa, elementów systemu. Tworzą one historię działania systemu. W nich spisywane są działania użytkowników takie jak logowanie i wylogowywanie się, zmiana hasła, oraz aktywność samego systemu operacyjnego i ważniejszych programów. To dzięki nim można dowiedzieć się o nieprawidłowościach w działaniu systemu, próbach penetracji i włamań.

Czytanie logów systemowych jest bardzo trudną i czasochłonną czynnością, więc wygodnie jest napisać skrypty wyszukujące interesujące informacje z konkretnych plików lub korzystać z gotowych programów, które zajmują się analizą logów. Dzięki informacjom zawartym w logach można wykryć luki w zabezpieczeniach systemu oraz wyśledzić ewentualnych intruzów. Aby to było możliwe należy zadbać o samo bezpieczeństwo logów. Powinny one być zabezpieczone przed modyfikacjami przez nieupoważnione osoby, czy programy. Dla zmaksymalizowania wiarygodności logów warto, aby dane z logów zapisywane były automatycznie do innego pliku lub urządzenia. Zapisywanie logów w niestandardowy sposób utrudnia intruzom zatarcie śladów.

Na ogół logi są to zwykłe pliki tekstowe ASCII. Każde zdarzenie notowane w logu zapisywane jest jako pojedynczy wiersz takiego pliku. Zwykle wiersz loga rozpoczyna się od dokładnej daty zdarzenia. Podana jest również nazwa programu i numer procesu, który dokonuje wpisu.

Ze wszystkimi dziennikami wiąże się konieczność pilnowania rozmiaru odpowiednich plików, jako że programy notujące zdarzenia nigdy nic z dzienników nie usuwają, a tylko dopisują nowe informacje. Także poza przeglądaniem zawartości dochodzi nowe zadanie dla administratora, a mianowicie *rotacja* logów.

Generalnie są trzy sposoby rotacji logów:

1. usuwanie,
2. zmiana nazwy, być może połączona z kompresją,
3. zmiana nazwy i usuwanie starych logów.

Najprostszym sposobem ustrzeżenia się przed rozrastającymi się logami jest ich regularne czyszczenie (usuwanie). Tracimy jednak w ten sposób możliwość sprawdzenia wydarzeń, które miały miejsce przed wyczyszczeniem loga.

Lepszym rozwiązaniem jest zmiana nazwy loga, na przykład przez dopisanie aktualnej daty i poddanie go kompresji. Można też spakowane logi przechowywać na innych nośnikach niż dysk twardy komputera (np. taśmy magnetyczne, CD-R itp.). Posiadanie w archiwum starych logów pozwala odtworzyć zdarzenia z przeszłości.

Dobrym rozwiązaniem w przypadku wielu logów (np. Syslog, `messages`, `popauth.log`), jest regularne przenumerowywanie loga i jego archiwów, tak że archiwum po przekroczeniu ustalonej wartości liczbowej jest usuwane. W przypadku Syslog'a, na systemie Solaris, co niedziela o godzinie 3:10 plik `syslog.7` jest usuwany, `syslog.6` staje się `syslog.7`, itd., natomiast `syslog` jest zmieniany na `syslog.0` i tworzony jest pusty plik `syslog`.

Administrator musi ocenić wagę dzienników i zdecydować który sposób rotacji do nich zastosować.

1.1 Logi systemu operacyjnego

Syslog

Syslog jest jednym z najważniejszych dzienników systemowych. Spośród innych logów wyróżnia go możliwość konfiguracji oraz sposób zapisu do niego. Otóż, Syslog zawiera informacje o zdarzeniach pochodzących od różnych programów. Ponieważ nie jest możliwy zapis do jednego pliku przez kilka procesów jednocześnie, powstaje problem kolejkowania wpisów. Zajmuje się tym program, którego nazwa jest taka sama jak nazwa dziennika to znaczy `syslog`. Tak więc wpisy do Syslog'a wykonuje program `syslog` na żądanie innych programów.

Dziennik Syslog można skonfigurować tak, aby zdarzenia z danego hosta były spisywane na nim bezpośrednio lub na innym wyznaczonym hoście, zwanym *loghost*. Pierwsza metoda jest wygodniejsza w przypadku pojedynczego serwera, gdy jednak zarządzamy kilkoma serwerami lepszym rozwiązaniem jest jeden centralny serwer logów, na którym gromadzone są logi z pozostałych hostów.

Na serwerze `math` w Syslog'u znajdują się głównie informacje z Sendmail, POP i IMAP, czyli programów bezpośrednio związanych z pocztą. Przykładowy wpis w Syslog'u dotyczący pojedynczej wiadomości może wyglądać następująco:

```
Jun 18 22:43:25 math sendmail[5357]: [ID 801593 mail.info] h5IKhPK05357:
from=<owner-majordomo@mizar.uwb.edu.pl>, size=2269, class=-60, nrcpts=12,
msgid=<Pine.GSO.4.30.0305270954410.28031-100000@math>, proto=ESMTP,
daemon=MTA, relay=mailgate [212.33.71.77]
```

```
Jun 18 22:43:26 math sendmail[5358]: [ID 801593 mail.info] h5IKhPK05357:
to=<trybulec@math.uwb.edu.pl>,\bancerek,<arturk@math.uwb.edu.pl>,\mariusz,\adam,
<milewski@math.uwb.edu.pl>,<jarek@math.uwb.edu.pl>,<bartek@math.uwb.edu.pl>,
\krzypraz, <bylinski@math.uwb.edu.pl>,<adamn@math.uwb.edu.pl>,\lapinski,
delay=00:00:01, xdelay=00:00:00, mailer=local, pri=618565, dsn=2.0.0, stat=Sent

Jun 18 22:43:26 math sendmail[5358]: [ID 801593 mail.info] h5IKhPK05357:
to="|/opt/sbin/mail2sms bancerek", ctladdr=<bancerek@math.uwb.edu.pl> (1004/10),
```

Z informacji tu zawartych można odczytać, że 18 czerwca o godzinie 22:43 została odebrana wiadomość od *owner-majordomo@mizar.uwb.edu.pl* za pośrednictwem komputera mailgate. Sekundę później wiadomość została przeskanowana i wyliczeni zostali odbiorcy przesyłki, do których ją wysłano.

Ostatni wpis świadczący o obecności *aliasu* lub pliku *forward* w przypadku odbiorcy *bancerek*. Sendmail informuje tutaj o niezbędnym, dodatkowym przetwarzaniu.

Wpis pochodzący od programu IMAP może wyglądać jak poniżej:

```
Jun 16 11:56:26 math imapd[20973]: [ID 666661 mail.info] Authenticated
user=krzypraz host=[212.33.73.244]
```

Jak widać 16 czerwca, o godzinie 11:56 miała miejsce autoryzacja użytkownika *krzypraz*. Podane jest z jakiego komputera (adresu IP) użytkownik się logował. Informacje te mogą być niezwykle pomocne w przypadku, gdy występuje podejrzenie, iż nieznany intruz próbuje czytać pocztę z konta użytkownika, który o tym nic nie wie.

`messages`

Kolejny plik mogący dostarczyć ważnych informacji to `/var/adm/messages`. Do pliku tego zapisywane są przeważnie komunikaty z jądra z czasu startu systemu oraz błędy pochodzące z różnych programów. Każda linia zawiera datę i godzinę uruchomienia programu, nazwę programu i komunikat. Szczególnie interesujące mogą być komunikaty pochodzące z programów `login` i `su`.

```
Jun  9 14:46:45 math su: [ID 810491 auth.crit] 'su root' failed for mariusz
on /dev/pts/17
```

Powyższa linia, pochodząca z pliku `messages` z serwera `math` informuje o tym, że 9 czerwca użytkownik *mariusz* próbował przełączyć się na konto *root*. Ponieważ podane hasło było nieprawidłowe zalogowanie zostało uniemożliwione. Wpis ten informuje, że albo użytkownik pomylił się wpisując hasło, albo próbował zalogować się jako *root* bez posiadania odpowiednich uprawnień, co może świadczyć o próbie włamania, zdobycia uprawnień root'a a tym samym pozyskania całkowitej kontroli nad serwerem.

Próby przełączenia na konto *root* to jedna z rzeczy o której administrator na pewno chciałby wiedzieć. Oczywiście w `messages` są odnotowywane jeszcze inne interesujące administratora zdarzenia. Do analizy tego loga służy miedzy innymi program Swatch. Wraz z administratorem zainstalowaliśmy go na serwerze `math`. Dokładne informacje o programie Swatch znajdują się w Rozdziale 2.1.

Podobne jak poprzednio choć już nie tak groźne są wpisy przedstawione poniżej. Dotyczą one: wylogowania użytkownika z powodu przekroczenia określonego czasu pracy z określoną usługą , na przykład przy FTP (`proftpd`), czy SSH (`sshd`), błędnego wpisu hasła przy logowaniu się przez użytkowników, rezultat zmiany hasła przez użytkowników (`popassd`).

```
Jun  3 20:34:52 math proftpd[15026]: [ID 567783 daemon.notice] math
(dhcp101.tkb.net.pl[212.33.84.101]) - FTP login timed out, disconnected.

May 29 00:20:52 math sshd[15901]: [ID 800047 auth.crit] fatal:
Read from socket failed: Connection timed out

Jun  2 11:20:16 math poppassd[20752]: [ID 818269 local2.error]
password changed for mskutnik
```

Do loga `messages` kierowane są też informacje z takich serwisów i programów jak: Samba (`smbd`), IP Filter (`ipf`), Snort (`Snort`). Program Snort opisany jest dokładnie w Rozdziale 2.2.

`lastlog, utmpx, wtmpx`

Dzienniki `lastlog, utmpx, wtmpx` różnią się od pozostałych, między innymi tym, że są zapisane w postaci binarnej, kompletnie nieczytelnej człowiekowi. W dzienniku `lastlog` znajdują się informacje o ostatnich wejściach użytkowników do systemu, w logu `utmpx` są zawarte informacje o aktualnie zalogowanych użytkownikach systemu, natomiast `wtmpx` kronikuje informacje o wejściach i wyjściach z systemu. Wszystkie te dzienniki zamieściłem w jednym podrozdziale, gdyż dotyczą one tej samej czynności, a mianowicie odnotowywania informacji na temat logowania użytkowników. Jak już wyżej wspomniałem logi te są nieczytelne i do oglądania ich służy program `last`. Program ten pobiera informacje z trzech wyżej wymienionych logów i wypisuje odpowiednie informacje w zależności od parametrów z jakimi został zawołany, np. polecenie `last -a -5` wypisze ostatnich pięć logowań do systemu, podając pełne nazwy zdalnych komputerów:

```
math# last -a -5
margitt   pts/3    Tue Jun 24 12:36 - 12:36  (00:00)   merkury.impan.gov.pl
anna.gom  pts/3    Tue Jun 24 12:10 - 12:12  (00:01)   212.33.73.74
mariusz   console  Tue Jun 24 12:05   still logged in  :0
bylinski  pts/11   Tue Jun 24 12:00   still logged in  212.33.71.231
aminov    pts/15   Tue Jun 24 11:17 - 11:31  (00:14)   212.33.73.210
```

Program `last` wypisuje:

- nazwę użytkownika (*margitt, mariusz, bylinski*),
- z jakigo terminala nastąpiło połączenie (pts/3, console),
- kiedy nastąpiło zalogowanie i wylogowanie oraz długość czasu w którym użytkownik był zalogowany (Tue Jun 24 11:17 - 11:31 (00:14)), ewentualnie którzy użytkownicy są nadal zalogowani (Tue Jun 24 12:00 still logged in), oraz
- z jakiego adresu IP nastąpiło połączenie (212.33.73.74).

Przy pomocy polecenia `last` łatwo można sprawdzić, kto aktualnie jest zalogowany w systemie, ostatnie logowanie danego użytkownika (np. `last -1 username`) oraz datę ostatniego restartu systemu (np. `last -1 reboot`).

sulog

Plik `/var/adm/sulog` jest to dziennik systemowy, w którym spisywane są wszystkie akcje programu `su`. Program `su` służy do przełączania się użytkownika z jednego na drugiego (z ang. switch user), tak aby uzyskać inne prawa dostępu i przywileje. Na przykład użytkownik *mariusz* chcąc uzyskać prawa *root*'a (administratora) po wpisaniu polecenia `su root` (konto *root* jest domyślne więc w tym wypadku wystarczy samo `su`) zostanie poproszony o podanie hasła *root*'a i jeśli je zna, to dalej będzie pracował jako *root*. Przełączenie takie pokazane jest w poniższym przykładowym wpisie z `sulog`. Każde wywołanie programu `su` wiąże się z dokonaniem wpisu do dziennika `sulog`, bez względu na to, czy próba przełączenia powiodła się, czy nie.

Oto dwa przykładowe wpisy z dziennika systemowego `sulog`:

```
SU 06/09 14:46 - pts/17 mariusz-root
SU 06/19 15:56 + pts/14 mariusz-root
```

Widzimy tam następujące informacje:

- dokładna data wywołania przez użytkownika programu `su`,
- czy system pozwolił dokonać zmiany użytkownika, jeśli tak to widnieje +, jeśli system nie pozwolił na zmianę, np. z powodu wpisania błędnego hasła, wtedy będzie −,
- terminal z którego został zawołany program `su`,
- oraz z jakiego na jakiego użytkownika nastąpiła, lub miała nastąpić zmiana.

W pewnym sensie informacje z `sulog` są duplikowane w logu `messages`, opisanym dokładnie w rozdziale 1.1. Do `messages` trafiają między innymi informacje o nieudanych próbach użycia programu `su`.

Zauważmy, że informacja zawarta w przykładzie w rozdziale 1.1 dotyczy tego samego nieudanego przełączeni się na *root*'a, co wyżej cytowany fragment dziennika `sulog`. Świadczy o tym: czas (Jun 9 14:46:45), nazwa użytkownika z którego miało nastąpić przełączenie (*mariusz*), oraz terminal (pts/17). Tak więc jeden program może kierować informacje o swojej pracy do kilku dzienników.

1.2 Dziennik `popauth`

Prawidłowo skonfigurowany serwer pocztowy, zezwala na wysyłanie poczty jedynie lokalnym użytkownikom. W przypadku `math` serwerem SMTP jest Sendmail. W konfiguracji Sendmail'a administrator może określić adresy IP komputerów, z których można wysyłać pocztę. Jeśli komputer ma przypisany adres IP na stałe nie ma problemu. Kłopot pojawia się gdy użytkownik łączy się z serwerem za pomocą modemu lub z sieci, w której zastosowano translację NAT adresów IP.

Jednym z rozwiązań jest zastosowanie programu `popauth`. Pobiera on z dziennika `syslog` informacje o autoryzacji użytkowników dokonane poprzez POP lub IMAP, a następnie dopisuje znalezione adresy IP do bazy adresów IP, z których zezwala się na wysyłanie poczty. Ponadto fakt ten trafia do dziennika `popauth.log`:

```
Jun 16 09:38:20 randrusz authenticating relaying for 212.33.73.194
Jun 16 11:45:25 milewski authenticating relaying for 212.33.66.153
Jun 16 11:56:26 krzypraz authenticating relaying for 212.33.73.244
Jun 16 12:18:06 trybulec authenticating relaying for 212.33.73.78
```

Z dziennika tego szybko można odczytać: kiedy użytkownik się zalogował (Jun 16 09:38:20), jaka jest jego nazwa (randrusz), czy program `popauth` dokonał autoryzacji (authenticating relaying), oraz z jakiego adresu IP użytkownik się łączył.

1.3 Dziennik `sudo.log`

W dzienniku `sudo.log` spisywane są informacje o działaniu programu `sudo`[10]. Program `sudo` umożliwia użytkownikowi wykonywanie pojedynczych poleceń i używanie programów, które wymagają uprawnień *root*'a. Aby użytkownikowi nie dawać pełnych praw dostępu do konta administratora, odpowiednio konfiguruje się program `sudo`, tak aby pozwalał określonemu użytkownikowi, lub grupie użytkowników pracować z ustalonymi programami jako administrator. Jest to bardzo pomocne przy dużych serwerachm, gdzie administrator może polecić osobie zaufanej wykonywanie niektórych prac, jak na przykład robienie archiwizacji. Może być to również pomocne w pracy samego administratora, który nie będzie musiał za każdym razem przełączać się na konto *root*'a, gdyż logowanie do systemu jako *root* powinno być zabronione. Każdorazowo przy uruchomieniu programu `sudo` do `sudo.log` jest dopisywana informacja o zdarzeniu. Oto przykładowy fragment:

```
May 23 20:56:39 : mariusz : TTY=pts/5 ; PWD=/export/home/mariusz ;
USER=root ; COMMAND=/usr/bin/kill -9 named

Jun 18 11:23:07 : mariusz : TTY=pts/9 ; PWD=/var/log ; USER=root ;
COMMAND=/usr/bin/crontab -l
```

Pierwszy wpis informuje o:

- czasie użycia programu `sudo` (May 23 20:56:39),
- nazwie użytkownika który z tego programu korzystał (*mariusz*),
- z jakiego terminala nastąpiło połączenie (pts/5),
- w jakim katalogu użytkownik się znajdował, w momencie uruchomienia programu (`/export/home/mariusz`),
- z jakich uprawnień korzystał (*root*),
- jakie polecenie wykonał (kill -9 named).

Informacje te pozwalają administratorowi kontrolować, czy zaufane osoby nie nadużywają przyznanych przywilejów. Dzięki informacjom zawartym w tym dzienniku można też odtworzyć, jakie zmiany zostały wykonane przez osoby zaufane.

1.4 Dziennik `rsync`

Program `rsync` [8] łączy w sobie cechy takich programów jak `cp`, `scp` oraz `rcp`. Posiada jednak o wiele więcej funkcji. Generalnie `rsync` służy do synchronizacji całych drzew katalogów, nie tylko w obrębie jednego komputera,

lecz przede wszystkim na odległość, skąd `r` (od ang. remote) w nazwie programu. Program `rsync` wykorzystuje algorytm tworzący sumy kontrolne plików, przez co nadpisuje on jedynie różnice między plikami. Jest to idealne narzędzie do aktualizacji *mirror*'ów.

Program `rsync` uruchomiony z opcją `--daemon`, działa jako demon i wtedy pełni funkcję podobną do serwera FTP. W Instytucie Matematyki na `math` poprzez serwer `rsync` dostępne jest to samo drzewo katalogów, które udostępnia serwer FTP. `rsync` nie daje możliwości "surf'owania" po zasobach jak FTP, ale jest znacznie wydajniejszy przy kopiowaniu całych gałęzi udostępnionego drzewa.

Serwer `rsync` odnotowuje wszystkie transfery w swoim logu. Poniżej prezentujemy jego mały fragment:

```
2003/05/06 18:07:00 [16589] rsync on multiboot from
  dyn-69-253.tor.dsl.tht.net (134.22.69.253)

2003/05/06 18:07:04 [16589] wrote 349159 bytes  read 11531 bytes
  total size 1690603
```

Możemy tu odczytać, że 6-tego maja 2003, o godzinie 18:07 z komputera o adresie IP 134.22.69.253 było pobierane drzewo o nazwie `multiboot`, serwer `rsync` przeczytał 11531 bajtów, wysłał 349159 bajty, a całkowity rozmiar synchronizowanego drzewa wynosi 1690603 bajtów.

1.5 Dziennik `idled`

Program `idled` działa jako demon i nadzoruje sesje użytkowników. Sprawdza:

- Czy zalogowani użytkownicy są aktywni, czy też są przez dłuższy czas bezczynni (stąd nazwa programu od ang. idled = bezczynny). Jeśli użytkownik jest zalogowany i nie podejmuje przez dłuższy czas żadnego działania, może to wskazywać na to, iż zapomniał się wylogować. Program powiadamia o tym fakcie wypisując na terminalu odpowiedni komunikat, odczekuje pewną chwilę i w końcu wyloguje użytkownika.

- Czy użytkownik nie jest zbyt długo zalogowany na serwerze. Czas pracy na serwerze przy jednorazowym zalogowaniu jest ustalony z góry przez administratora. W przypadku przekroczenia tego limitu, użytkownik zostanie poproszony o zakończenie sesji i po odczekaniu pewnego czasu wylogowuje użytkownika.

- Czy nie jest otwartych zbyt wiele sesji przez jednego użytkownika oraz wszystkich użytkowników łącznie. Program `idled` prosi o wylogowanie tych użytkowników, którzy mają otwartych najwięcej sesji lub są najdłużej bezczynni. Po upływie ustalonego czasu wylogowuje część użytkowników.

Poniżej znajdują się przykładowe wpisy z dziennika `idled.log`:

```
Mon May 14 09:49:44 : bancerek on /dev/pts/2 because idle
Tue May 15 07:02:27 : bartek on /dev/pts/4 because idle
```

Wiemy stąd, że 14-tego maja 2003, o godzinie 9:49 użytkownik *bancerek* zalogowany na terminalu `/dev/pts/2` został wylogowany z powodu dłuższej bezczynności.

1.6 Dziennik `poppassd`

Program `poppassd` umożliwia zmianę hasła przez sieć, bez konieczności logowania się na serwer. Działa w trybie klient-serwer i używa bardzo prostego protokołu, zawierającego cztery słowa kluczowe: *user, pass, newpass, quit.* Dzięki zastosowaniu `poppassd` zmianę hasła można dokonać poprzez formularz na stronie WWW przy pomocy przeglądarki. Wyeliminowana jest możliwość manipulacji hasłami, gdyż nie ma bezpośredniego dostępu do systemu i do plików z hasłami. Jak widać działanie programu ma istotny wpływ na zachowanie bezpieczeństwa na serwerze, dlatego też informacje o wszystkich operacjach przeprowadzanych przez ten program są zapisywane w logu `poppassd-log`. Poniżej znajduje się fragment takiego pliku:

```
Jun 10 13:38:11 math poppassd[27683]: [ID 818269 local2.error] password
changed for elam

Jun 19 13:57:21 math poppassd[22023]: [ID 818269 local2.error] password
changed for randrusz
```

Podana jest dokładna data kiedy program został użyty, skąd nastąpiło jego wywołanie, kto próbował zmienić hasło, czy próba się udała, czy też nie. Z punktu widzenia administratora dziennik `poppassd-log` może dostarczyć cennych informacji, na przykład czy nie ma prób włamania przez złamanie hasła. O takim działaniu świadczyłyby wpisy dotyczące jednego konta, powtarzające się w krótkich odstępach czasu. W logu tym niestety brakuje adresu IP, z którego następuje połączenie z `poppassd`, co utrudnia wytropienie ewentualnego intruza.

1.7 Apache

Efektywne i prawidłowe działanie serwisu Apache HTTP jest możliwe tylko wtedy jeśli serwis jest prawidłowo zarządzany. To natomiast pociąga za sobą ciągłe monitorowanie jego pracy, kontrolę jego aktywności, oraz jak najszybsze usuwanie zaistniałych problemów. Niezastąpionym źródłem informacji na ten temat są dzienniki tworzone przez serwer Apache. Zazwyczaj są to dwa

pliki: `access.log` i `error.log`. Każdy wirtualny serwer powinien mieć swoje odrębne logi. Przy dużym ruchu HTTP szybko rośnie objętość tych logów, szczególnie `access.log`. Aby zapobiec niekontrolowanemu rozrostowi logów, co spowalnia działanie serwera Apache, najlepiej stosować automatyczną, regularną rotację. Na serwerze `math` stosowany jest drugi sposób rotacji opisany w rozdziale 1.

`error.log`

Dziennik `error.log` jest logiem, w którego serwer Apache spisuje informacje o przebiegu pracy usługi, informacje diagnostyczne, oraz ewentualnie zaistniałe błędy. Jest to podstawowe miejsce, gdzie administrator powinien zajrzeć w przypadku nieprawidłowej pracy serwera Apache, w celu zdiagnozowania problemu. Poniżej znajdują się przykładowe wpisy z tego loga.

```
[Sun Jun 15 03:58:55 2003] [error] [client 203.109.249.136] File does
not exist: /export/home1/httpd/math.uwb.edu.pl/htdocs/favicon.ico

[Sun Jun 15 06:05:46 2003] [error] [client 212.33.60.71] File does not
exist: /export/home1/httpd/math.uwb.edu.pl/htdocs/scripts/root.exe
```

Kolejno podany jest czas wystąpienia błędu, klasa zdarzenia, adres IP klienta, który spowodował wystąpienie błędu oraz treść komunikatu. W obu przypadkach użytkownicy wywoływali pliki które nie istnieją.

`access.log`

Do dziennika `access.log` trafiają informacje o żądaniach odebranych przez serwer Apache. Format dziennika jest z zgodny ze standardem CFL (Common Logfile Format), co pozwala na wygenerowanie statystyk różnymi programami. Na serwerze `math` statystyki są tworzone programem Webalizer [13] i dokładny ich opis znajduje się w rozdziale 4.1. Poniżej znajdują się przykładowe wpisy z dziennika `access.log`:

```
66.196.65.12 - - [15/Jun/2003:03:40:54 +0200] "GET /robots.txt HTTP/1.0"
200 100 "-" "Mozilla/5.0 (Slurp/si; slurp@inktomi.com;
http://www.inktomi.com/slurp.html)"

64.68.82.67 - - [15/Jun/2003:03:41:15 +0200] "GET
/pl/html/staff/vitae.mhtml?cID=158 HTTP/1.0" 200 - "-" "Googlebot/2.1
(+http://www.googlebot.com/bot.html)"
```

Każdy pojedynczy wpis zawiera kolejno:

- adres IP użytkownika który korzystał z usługi Apache (66.196.65.12 w pierwszym przypadku),

- nazwę użytkownika, jeśli dostęp do żądanego dokumentu chroniony jest hasłem; dwa myślniki oznaczają, że identyfikacja użytkownika nie była wykonywana,
- dokładny czas korzystania z usługi (15/Jun/2003:03:41:15),
- rodzaj metody, zazwyczaj jest to metoda GET, ścieżka do żądanego dokumentu (/robots.txt) oraz wersja protokołu HTTP (HTTP/1.0),
- rodzaj kodu odpowiedzi serwera, jeśli kod ten zaczyna się od 2, tak jak w wyżej podanych przykładach, oznacza to, że nawiązano kontakt, jeśli zaczyna się od 3, wynikł błąd ze strony użytkownika, a jeśli 4 to nastąpił błąd serwera,
- wielkość pliku przesłanego do użytkownika w bajtach, jeśli nic nie zostało przesłane pojawi się −,
- nazwa przeglądarki użytkownika (Mozilla/5.0),
- z jakiej strony użytkownik został przekierowany (tzw. referrer).

1.8 Samba

Oprogramowanie Samba to realizacja protokołów SMB (Server Meassage Block) oraz CIFS (Common Internet File System) głównie używanych w systemach Windows [1]. Pozwala na wymianę plików oraz dzielenie drukarek między komputerami Unix i Windows w ramach jednej sieci. W skład oprogramowania Samba wchodzą dwa demony: `smbd` oraz `nmbd`. Pierwszy odpowiada za współdzielenie plików i drukarek, drugi wspomaga przeglądanie zasobów sieciowych. Każdy z nich ma swój własny dziennik. Oba dzienniki można skonfigurować tak, aby ich rozmiar nie przekraczał pewnego ustalonego limitu. Dzięki temu nie jest konieczna rotacja tych logów.

Na `math` Samba pozwala użytkownikowi montować swój katalog domowy z serwera na komputerze Windows jako dysk logiczny.

log.smb

Plik `log.smb` jest logiem demona `smbd`. Oto jego fragment:

```
[2003/04/04 10:39:03, 1] smbd/process.c:process_smb(611)
Connection denied from 213.25.239.86

[2003/06/05 15:25:51, 1] smbd/service.c:make_connection(550)
analiza-2 (212.33.73.242) connect to service kotowicz as user
kotowicz (uid=1012, gid=10) (pid 4728)
```

Z informacji tu zawartych administrator może odtworzyć, co działo się w związku z usługą SMB. W tym wypadku dokładnie 4-tego kwietnia 2003 o godzinie 10:39 z komputera o adresie 213.25.239.86 podejmowana była próba podłączenia się do zasobów serwera `math`. Ponieważ tylko komputery z lokalnej sieci IM mają prawo korzystać z usługi Samba, `smbd` odmówił nawiązania tego połączenia.

Drugi wpis można odczytać następująco: 5-tego czerwca 2003 o godzinie 15:25 użytkownik *kotowicz* zalogował sie na serwer poprzez usługę SMB z komputera *analiza-2* o adresie 212.33.73.242. Uzyskał prawa użytkownika o numerze 1012 i grupy 10 na `math`. Proces `smbd` obsługujący to połączenie miał numer 4728.

log.nmb

Plik `log.nmb` jest logiem demona `nmbd` i niżej jest załączony jego fragment:

```
[2003/02/22 01:25:31, 1]
nmbd/nmbd_incomingdgrams.c:process_reset_browser(730)
process_reset_browser: received diagnostic browser reset request from
THETA<00> IP 212.33.73.195 state=0x2

[2003/05/03 14:36:16, 0]
nmbd/nmbd_become_lmb.c:become_local_master_stage2(405)
```

Na podstawie danych zawartych w `log.nmb` można ustalić, jakie komputery uczestniczą w dzieleniu zasobów i jaką w tym procesie pełnią rolę. Log ten jest szczególnie ważny wtedy, gdy Samba realizuje dodatkowo usługę WINS (Windows Internet Name Service), podobną do DNS (Domain Name Service), specyficzną dla Windows. W Instytucie Matematyki WINS nie jest używany.

Z powyższego przykładu wynika, że 22 lutego 2003 o godzinie 1:25 komputer o nazwie `theta` zażądał od serwera `math` odświeżenia listy komputerów dzielących zasoby w sieci SMB. Następny wpis z 3 marca 2003 mówi, że `math` stał się lokalnym nadzorcą listy komputerów współdzielących zasoby.

1.9 ProFTPD

Usługa FTP jest starszą usługą niż HTTP i powstała jako uzupełnienie do usługi Telnet. O ile Telnet pozwala logować się na zdalnym serwerze i uruchamiać na nim programy, to FTP umożliwia transfer plików pomiędzy komputerami. Niestety w standardzie FTP nie przewidziano możliwości przekazywania haseł w postaci zaszyfrowanej, co w zasadzie ogranicza FTP do publicznych serwisów anonimowych, gdzie hasło nie jest potrzebne do wejścia na serwer FTP. W ten właśnie sposób FTP jest zorganizowane na `math`. Realizacja usługi FTP odbywa się przy wykorzystaniu oprogramowania ProFTPD [12].

Poprzez konfigurację serwera ProFTPD można decydować o tym jakie dzienniki będą tworzone i jakie w nich znajdą się informacje. Domyślnie używane są trzy dzienniki: `auth.log`, `acces.log` oraz `xferlog`.

auth.log

W dzienniku `auth.log` ProFTPD spisuje logowanie się użytkowników do FTP.

```
Anonymous FTP - University of Bialystok, Institute of Mathematics [3264]
  service.dik.cvut.cz [18/May/2003:00:00:09 +0200] "USER anonymous" 331

Anonymous FTP - University of Bialystok, Institute of Mathematics [3264]
  service.dik.cvut.cz [18/May/2003:00:00:09 +0200] "PASS -wget@" 230
```

Na podstawie powyższych wpisów wiemy, że dnia 18 maja tego roku, o północy, z komputera *service.dik.cvut.cz* nastąpiło logowanie. Jako nazwę użytkownika podano `anonymous` i akcja zakończyła się ze statusem 331. Drugi wpis mówi, że jako hasło podano `-wget@`.

access.log

Rola dziennika `acces.log` jest podobna jak dziennika o tej samej nazwie w serwerze Apache (por. 1.7) i tak jak tam jest on zgodny ze standardem CLF. Zamiast metody GET będziemy tutaj mieli słowo kluczowe RETR (od ang. retrieve = pobierz) ptotokołu FTP.

```
service.dik.cvut.cz UNKNOWN ftp [18/May/2003:02:16:48 +0200]
  "RETR mrtg-2.9.29-Solaris8-i686.pkg.tgz" 226 492386

service.dik.cvut.cz UNKNOWN ftp [18/May/2003:02:16:55 +0200]
  "RETR mtools-3.9.1-Solaris8-x86.pkg.tgz" 226 209736
```

Użytkownik, którego logowanie opisaliśmy wyżej, wysłał do serwera FTP polecenie RETR, aby pobrać plik `mrtg-2.9.29-Solaris8-i686.pkg.tgz` oraz `mtools-3.9.1-Solaris8-x86.pkg.tgz`.

xferlog

Ponieważ na `math` jest wyłącznie anonimowe FTP, dzienniki tej usługi będą miały bardziej charakter statystyczny, niż diagnostyczny. Najciekawszym w tym względzie jest tu opisywany `xferlog`.

```
Sun May 18 02:16:48 2003 16 service.dik.cvut.cz 492386
  /solaris/x86/5.8/mrtg-2.9.29-Solaris8-i686.pkg.tgz
    b _ o a -wget@ ftp 0 * c

Sun May 18 02:16:55 2003 6 service.dik.cvut.cz 209736
  /solaris/x86/5.8/mtools-3.9.1-Solaris8-x86.pkg.tgz
    b _ o a -wget@ ftp 0 * c
```

Wyżej przedstawiony fragment tego loga ujawnia więcej szczegółów dotyczących transferu z 18 maja w nocy. Poza dokładną data i adresem klienta podany tutaj mamy rozmiar pobieranego pliku, pełną ścieżkę do niego i kod wyniku.

Na podstawie `xferlog` generowane są statystyki opisane w rozdziale 4.2. Do rotacji logów serwera FTP napisaliśmy dwa skrypty Shell'owe opisane w 5.2.

1.10 Horde/IMP

Na serwerze `math`, dla ułatwienia korzystania z poczty, zainstalowany jest system Hored/IMP [11]. Pozwala on przeglądać i wysyłać pocztę przy pomocy zwykłej przeglądarki internetowej, z dowolnego miejsca na świecie gdzie jest dostępny Internet. Usługa jest szczególnie przydatna dla pracowników Instytutu przebywających w podróży, na konferencji, tam gdzie jest przeglądarka, ale nie ma wygodnego programu do zarządzania pocztą.

```
Jun 19 12:46:51 Web e-mail system [notice] [imp] Logout for
justyna@math.uwb.edu.pl [193.147.222.244] from {math.uwb.edu.pl:993}
[on line 34 of
"/export/home1/httpd/mathmail.uwb.edu.pl/htdocs/imp/login.php"]

cze 19 13:44:34 Web e-mail system [notice] [imp] Login success for
randrusz@math.uwb.edu.pl [217.96.137.135] to {math.uwb.edu.pl:993}
[on line 64 of
"/export/home1/httpd/mathmail.uwb.edu.pl/htdocs/imp/redirect.php"]
```

Z dziennika tworzonego przez system Horde/IMP można dowiedzieć się kto, kiedy i z jakiego miejsca korzystał z systemu. I tak 19-tego czerwca o godzinie 12:46 użytkownik *justyna*, łączący się z komputera o adresie IP 193.147.222.244, wylogował się z systemu IMP. Natomiast nieco później o 13:44 użytkownik *randrusz* połączył się z komputera o adresie 217.96.137.135 i poprawnie zalogował się.

Rozdział 2

Systemy IDS

System IDS (Instrusion Detection System) służy do wykrywania zdarzeń stanowiących potencjalne zagrożenie dla bezpieczeństwa komputera.

Decyzje o tym czy dane zdarzenia stanowią zagrożenie, czy nie system podejmuje na podstawie bazy danych reguł i charakterystycznych wzorów, podobnie jak to jest w przypadku oprogramowania antywirusowego. Zatem sprawność systemu IDS zależy od aktualności takiej bazy wzorców. W odróżnieniu od programów antywirusowych, kóre potrafią "wyleczyć" pliki, systemy IDS nie podejmują żadnych akcji, poza infomowaniem administratora o potencjalnych zagrożeniach. Po uzyskaniu od IDS ostrzeżenia administrator powinien ocenić stopień zagrożenia i podjąć stosowne działania prewencyjne. Może to być na przykład zablokowanie podejrzanego adresu IP na firewall'u. W tej pracy nie będziemy jednak zajmować się technikami zabezpieczania systemów komputerowych, skupiamy się wyłącznie na analizie logów.

Na serwerze `math` zostały zainstalowane dwa programy typu IDS: Swatch i Snort.

2.1 Swatch

Swatch jest to pogram mający na celu ułatwić prace administratora oraz zwiększyć bezpieczeństwo serwera. Bardzo trudnym i żmudny obowiązkiem jest sprawdzanie dzienników systemowych w poszukiwaniu podejrzanie wyglądających zdarzeń i sygnałów o nieprawidłowościach lub świadczących o próbie, włamania czy chęci modyfikacji plików przez intruza. Dane spisane w dziennikach systemowych czyta się bardzo trudno; ciężko z pośród dużej ilości informacji w nich zawartych wyłapać te istotne, czy mówiące o jakimś zagrożeniu. Program Swatch śledzi zawartość dzienników systemowych, sprawdza każdą dodaną do nich informację i wysyła komunikaty do administratora z informacjami o potencjalnych problemach, jakie zostały wykryte. Mając podany wzór (regularne wyrażenie) podejrzanych zdarzeń typu: *faild, Permission denied*, itp., wyłapuje takie informacje z dzienników systemowych. Jeśli po-

dany wzorzec pasuje do komunikatu w logu program podejmuje określone w konfiguracji działanie.

Może on:

- wysyłać na pulpit ostrzeżenie,
- włączyć alarm w postaci sygnału dźwiękowego
- informować administratora pocztą elektroniczną.

Swatch może śledzić informacje dopisywane sukcesywnie do dziennika, lub może przeczytać dziennik w całości. W drugim przypadku zaznacza w logu najważniejsze informacje w postaci podświetlenia ich różnymi kolorami, co ułatwia przeglądanie. Jeśli w gąszczu informacji z dziennika systemowego jest gdzieś informacja o próbie logowania się na serwer jako root, Swatch po przeanalizowaniu loga podświetli tą informację na czerwono, jako że jest ona bardzo istotna dla administratora.

Zazwyczaj Swatch pracuje jako demon i analizuje informacje dopisywane do dziennika systemowego linijka po linijce. Jeśli analizowanym plikiem jest Syslog lub `messages`, to po rotacji tych logów konieczny jest restart Swatch'a, tak aby zaczął on monitorować nowo powstały plik w wyniku rotacji. Na systemie Solaris oba wymienione logi są restartowane co niedziela o godzinie 3:10 rano. Tak więc następujący wpis do tablicy *crontab*

```
# Swatch needs to be restarted after messages are rotated
40 3 * * 0 /etc/init.d/swatch restart
```

spowoduje, że Swatch zostanie zrestartowany o godzinie 3:40 w każdą niedzielę, czyli tuż po rotacji Sysloga i `messages`. W ten sposób zainstalowaliśmy program Swatch na serwerze `math`. Jego pełny plik konfiguracyjny wygląda jak poniżej:

```
# Swatch configuration file for constant monitoring

# Bad login attempts
watchfor   /failed/
        mail

# System crashes and halts
watchfor   /(panic|halt)/
        mail

# System reboots
watchfor   /SunOS Release/
        mail
```

Plik ten jest dość sugestywny i tak Swatch wyłapuje z analizowanego loga `messages` komunikaty zawierające słowa: failed, panic, halt oraz napis "SunOS Release". Po znalezieniu jednego z nich wysyłana jest kopia komunikatu do administratora. Ostatni napis pojawia się w momencie startu systemu, a więc administrator zostaje poinformowany o każdym restarcie serwera, np. w wyniku zaniku napięcia.

2.2 Snort

Snort jest narzędziem określanym jako IDS (Instrusion Detection System - system wykrywania intruzów). Dokonuje on inspekcji pojedynczych pakietów oraz całych ich strumieni (sesji) pod kątem symptomów ataku. Mogą nimi być określone treści, np. charakterystyczna sekwencja bajtów zorientowana na przepełnienie bufora aplikacji i uruchomienie powłoki użytkownika z prawami administratora, odwołanie do skryptu CGI (którego luki są powszechnie znane), kod wirusa czy inna, zdefiniowana sekwencja znaków. Przeszukiwanie ma charakter masowy. Przeprowadzane jest na podstawie bazy sygnatur, które precyzyjnie określają jego warunki. Systemy IDS mogą więc wykryć różne rodzaje ataków, od typowych włamań do systemów komputerowych, poprzez ataki polegające na odmowie świadczenia usług, aż do przesyłania złośliwych programów czy spamu. Należy się jednak liczyć z możliwością ukrycia przed nim niebezpiecznych transmisji, czy oszukania fałszywymi atakami w celu zamaskowania prawdziwego zagrożenia.

Jeśli Snort ma wykrywać jak najwięcej, musi mieć możliwość wglądu w jak największą część ruchu. Jeśli zlokalizujemy go za firewallem, pozbawimy się możliwości wglądu w ataki zatrzymane przez ścianę ogniową. Można ulokować Snorta przed bramą do właściwej sieci lokalnej, jednak nie może on wówczas śledzić ruchu przychodzącego łączami wirtualnymi sieci prywatnych. Dodatkowe zamieszanie może wprowadzać translacja adresów IP (NAT). Snort ustawiony przed routerem dokonującym translacji będzie widział jedynie adres publiczny i zazwyczaj fałszywe porty usługowe.

Ograniczenia te można obejść, uruchamiając np. kilka kopii Snorta na różnie zlokalizowanych topologicznie maszynach lub tak sterując ruchem, by trafiał on w całości do Snorta. Za pierwszym rozwiązaniem przemawia duża sprawność Snorta, a także kompleksowości rozwiązania. Drugie wymaga bardziej skomplikowanych czynności administracyjnych gdyż nie zawsze da się pokierować bezpiecznie całym ruchem, tak by był do wglądu dla Snorta.

Efekt działania Snort'a

W celu zwiększenia bezpieczeństwa oraz otrzymywania informacji na temat ewentualnie zaistniałych niebezpieczeństw wraz z administratorem zainstalowaliśmy w Instytucie Matematyki na serwerze `math` program Snort [9]. Pro-

gram został pobrany z oficjalnej strony internetowej Snort'a w postaci kodu źródłowego. W celu ułatwienia instalacji i konfiguracji przygotowaliśmy pakiet instalacyjny na platformę Solaris x86 zgodny z formatem `pkgadd`[1]. Informacje o wykrytych zagrożeniach Snort gromadzi w bazie danych. Może to być dowolna baza SQL. W przypadku `math` jest to baza MySQL. Uzupełnieniem programu jest interfejs ACID (Analysis Console for Intrusion Databases) [2]. Po zainstalowaniu, skonfigurowaniu i uruchomieniu wszystkich elementów programu, efekty działania Snort'a można oglądać pod adresem:

`http://math.uwb.edu.pl/snort/`

Jako że informacje tam zawarte są potrzebne wyłącznie administratorowi wgląd na stronę chroniony jest hasłem.

Dane zawarte na głównej stronie pozwalają określić: kiedy po raz ostatni administrator odwiedził stronę, co od tego momentu się wydarzyło, ile wystąpiło alarmów (alerts), w ilu kategoriach tematycznych pogrupowane są alarmy, jaki jest rozkład procentowy użycia poszczególnych protokołów (TCP, UDP, ICMP). Na stronie głównej są też odnośniki do stron z:

- pięcioma lub piętnastoma najczęściej pojawiającymi się alarmami,
- opisem ostrzeżeń, w zależności od czasu w którym nastąpiły (od ostatniej wizyty, w przeciągu 24, 72 godzin, czy też najnowsze),
- graficznym przedstawieniem działania programu, itp.

Poniżej przedstawiony jest widok ze strony przedstawiającej pięć najczęściej powtarzających się alarmów.

	< Signature >	< Classification >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
☐	[nessus] [snort] WEB-MISC robots.txt access	web-application-activity	1426 (23%)	1	221	1	2003-06-09 15:03:04	2003-06-30 00:38:24
☐	[snort] [snort_decoder] Truncated Tcp Options	unclassified	583 (9%)	1	1	295	2003-06-09 14:47:11	2003-06-20 22:31:09
☐	[snort_(snort_decoder): IP Options Detected	unclassified	544 (9%)	1	20	2	2003-06-10 19:10:25	2003-06-27 10:09:51
☐	[snort] WEB-IIS cmd.exe access	web application attack	492 (8%)	1	63	1	2003-06-09 20:48:30	2003-06-29 23:49:52
☐	us [snort] SCAN SOCKS Proxy attempt	attempted-recon	477 (8%)	1	99	33	2003-06-09 10:28:26	2003-06-20 22:54:57

Do zrozumienia informacji zgromadzonych w powyższej tabeli nie jest potrzebne studiowanie dokumentacji Snort'a, ani też duża wiedza z zakresu zabezpieczeń systemów komputerowych. W pierwszej kolumnie gdzie wypisane są informacje o rodzaju ostrzeżenia w nawiasie kwadratowym umieszczono odnośniki do strony z dokładnym wyjaśnieniem zarejestrowanego zdarzenia i stopnia zagrożenia. Z tabeli można odczytać ile zdarzeń tego samego typu wystąpiło, kiedy pierwsze i ostatnie zdarzenie tego typu miało miejsce. Tabela posortowana jest ze względu na częstość występowania ostrzeżenia. Pierwszy

[1]Pakiet dostępny jest na `ftp://math.uwb.edu.pl/solaris/x86/5.8`

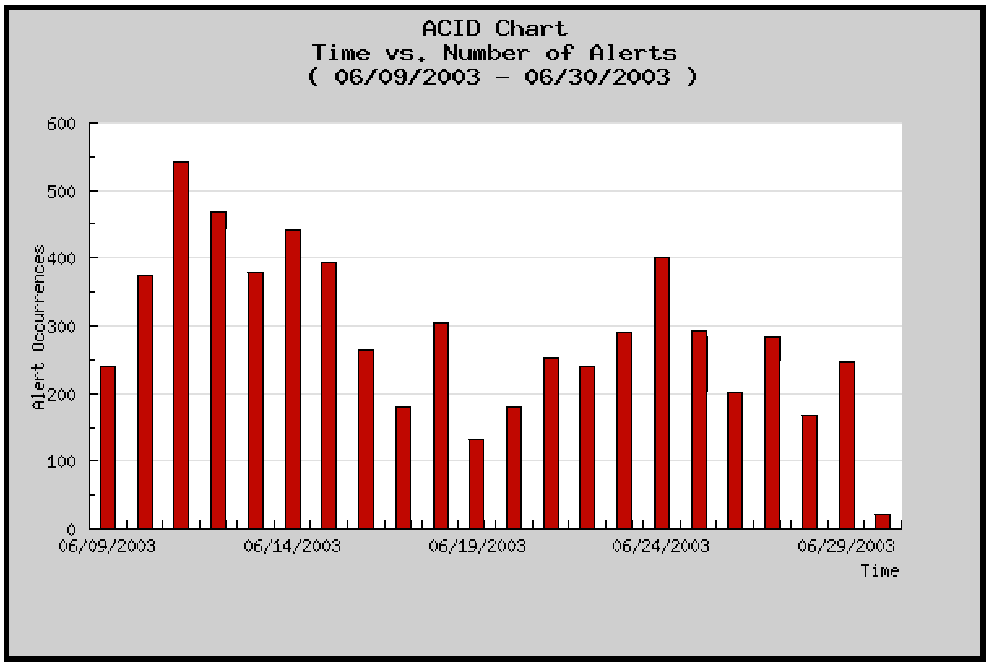
wiersz z tabeli mówi że najczęściej powtarzającymi się ostrzeżeniami, stanowiącymi aż 23% ogółu, są informacje o pobieraniu przez różne wyszukiwarki internetowe pliku `robots.txt`, w którym znajdują się dyspozycje dla tych wyszukiwarek. Tego typu informacje pomagają administratorowi zabezpieczyć serwer przed włamaniem, czy niepowołanym działaniem osób z zewnątrz.

W trzeciej kolumnie podana jest ilość ostrzeżeń tego samego typu. Liczby te są jednocześnie odnośnikami do stron przedstawiających szczegółowy rozkład tych ostrzeżeń. Przykład takiej strony został podany poniżej.

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-2)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:03:04	202.108.250.199:55814	212.33.73.194:80	TCP
#1-(1-3)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:04:59	66.196.65.14:10092	212.33.73.194:80	TCP
#2-(1-9)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:36:44	12.21.32.30:7592	212.33.73.194:80	TCP
#3-(1-11)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:38:28	209.73.164.50:42877	212.33.73.194:80	TCP
#4-(1-13)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:47:50	209.237.238.173:53848	212.33.73.194:80	TCP
#5-(1-14)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:49:43	66.196.65.13:59350	212.33.73.194:80	TCP
#6-(1-15)	ressus[snort]WEB-MISC robots.txt access	2003-06-09 15:50:58	209.237.238.176:43164	212.33.73.194:80	TCP

Z tabeli można odczytać szczegóły dotyczące klasy ostrzeżenia (WEB-MISC robots.txt access - wcześniej opisywane pobieranie pliku `robots.txt`), kiedy dokładnie ono nastąpiło, jaki był źródłowy adres IP użytkownika oraz na którym porcie transmisja się odbywała, jaki był docelowy adres IP z uwzględnieniem portu (adres serwera `math` 212.33.73.194 :80 port usługi HTTP), oraz z jakiego protokołu użytkownik korzystał (TCP). Powyższa tabela może pomóc administratorowi w ustaleniu, kto dokładnie dopuścił się ewentualnego wykroczenia. Praktycznie pod każdym napisem, czy też liczbą znajduje się odnośnik do kolejnej strony pokazującej bardziej szczegółowe dane.

Administrator oprócz gotowych stale tworzonych tabel, w których zawarte są najważniejsze informacje o alarmach, może też samodzielnie tworzyć wykresy. Parametry wykresu ustala się w formulażu. Przykładowy wykres znajduje się poniżej.



Administrator może stworzyć wykres zawierający dokładnie te dane które go interesują, bez konieczności szukania w gąszczu wszystkich wpisów. Dobierając odpowiednio: rodzaj usługi która ma być przedstawiona (w tym przypadku ilość alarmów), okres czasu w którym nastąpiły naruszenia (od 06/05/2003 do 30/06/2003), rodzaj wykresu (słupkowy), rozmiar wykresu (jego wielkość w pikselach), otrzymuje wykres na temat konkretnych danych które go interesowały.

Jak widać, wyłapanie podejrzanych i niewłaściwych działań użytkowników, oraz danych o ich samych, nie sprawia najmniejszego problemu. Dzięki programowi Snort w zasadzie nie jest możliwe wykonanie na serwerze jakichkolwiek operacji bez wiedzy administratora. W programie Snort istnieje możliwość konfiguracji wagi zagrożeń. Przy wystąpieniu określonych zdarzeń program momentalnie powiadomi administratora o niebezpieczeństwie.

Rozdział 3

MRTG

MRTG (Multi Router Traffic Grapher) jest narzędziem do monitorowania i wizualizacji niemal dowolnych wielkości związanych z działaniem systemu komputerowego, pracującego pod kontrolą systemu Unix lub Windows NT. Możliwe jest także monitorowanie innych wielkości, o ile dostarczymy odpowiednie dane. Wyniki działania programu MRTG zapisywane są w formie HTML, co umożliwia oglądanie tych wyników poprzez internet, przy pomocy zwykłej przeglądarki.

Podstawowym zastosowaniem MRTG jest monitorowanie ruchu sieciowego, wykorzystując protokół SNMP (Simple Network Management Protocol). Śledzić można również inne zmienne SNMP, można także użyć zewnętrznego programu dostarczającego dane, co umożliwia monitorowanie np. obciążenia procesora, sesji użytkowników itd. Dane te gromadzone są w plikach `log` i przedstawiane są w postaci wykresów. MRTG obrazuje zmiany na wykresie dziennym, tygodniowym, miesięcznym oraz rocznym.

Oprócz wizualizacji monitorowanych wielkości MRTG udostępnia dodatkową funkcjonalność:

- możliwość reagowania na zmiany wartości mierzonych wielkości,
- można ustalić wartości progowe, minimalne i maksymalne dla każdej z monitorowanych wielkości i określić jakie akcje będą podejmowane po przekroczeniu tych progów.

W niniejszej pracy poszczególne monitorowane obiekty nazywane są *celami*, aby uniknąć konieczności używania za każdym razem powyższego długiego sformułowania. Określenie to jest odpowiednikiem angielskiego *target*, które jest powszechnie stosowane w anglojęzycznej dokumentacji pakietu MRTG. Celami mogą zatem być wspomniane wcześniej: ruch sieciowy, obciążenie procesora, użycie pamięci itd.

3.1 Sposób działania MRTG

MRTG pobiera dane o wartości monitorowanej w równych odstępach czasu. W domyślnym sposobie pracy wartości otrzymane z kolejnych odczytów są od siebie odejmowane. Jest to użyteczne w przypadku odczytywania ilości przesłanych bajtów przez interfejs sieciowy za pomocą SNMP. Otrzymujemy wtedy ilość przesłanych bajtów pomiędzy poszczególnymi odczytami.

Następnie dla przedziału czasu pomiędzy przedostatnim a ostatnim odczytem obliczana jest wartość średnia transferu mówiąca ile średnio przesyłano bajtów na sekundę. Uzyskujemy ją dzieląc ilość przesłanych bajtów przez (stałą) wartość czasu, w którym zostały one przesłane. Wartości prezentowane przez MRTG są wartościami średnimi, MRTG nie obrazuje zmian mierzonej wielkości w każdym momencie.

Po obliczeniu wartości średnich w przedziałach, pomiędzy którymi dokonano odczytów następuje normalizacja. Celem normalizacji jest zmiana przedziałów, w których pierwotnie dokonano pomiarów na znormalizowane przedziały - o tej samej długości, lecz przesunięte tak, aby rozpoczynały się np. o pełnej godzinie.

Znormalizowane wartości zapisywane są w pliku `log`. Aby zapobiec nieskończonemu rozrastaniu się takiego pliku dokonywana jest kompresja otrzymanych wartości. Na podstawie znormalizowanych wartości odpowiadających przedziałom 5 minut tworzony jest wykres dzienny.

Po pewnym czasie (nieco więcej niż jeden dzień) wartości z 6 kolejnych przedziałów łączone są w jedną wartość odpowiadającą 30 minutom. Wartości takie z kolei tworzą wykres tygodniowy, a po nieco więcej niż 8 dniach są scalane w wartości odpowiadające 2 godzinom tworząc w rezultacie wykres miesięczny. Ostatnim etapem scalania jest tworzenie przedziałów odpowiadających dniu, które to przedstawiane są na wykresie rocznym.

Oprócz opisanego sposobu działania możliwe jest także dokonywanie pomiarów bez odejmowania wartości mierzonej w danej chwili od wartości poprzedniej. W dalszym ciągu jednak wartość ta dzielona jest przez przedział czasu. Innymi słowy odpowiada to domyślnemu sposobowi obliczania wartości średniej, z tym że za wartość poprzednią przyjmowane jest zawsze zero.

Możliwe jest także zrezygnowanie z dokonywania jakiegokolwiek obliczania wartości średniej przez MRTG. Wartości otrzymane z pomiarów nie są odejmowane od siebie i nie są też dzielone przez przedział czasu. Jest to użyteczne, gdy mierzymy np. zajętość dysków, obciążenie procesora, czy też temperaturę.

3.2 Instalacja i konfiguracja

W celu uproszczenia instalacji i konfiguracji programu MRTG, przygotowaliśmy pakiet instalacyjny na platformę Solaris x86 zgodny z formatem `pkgadd`, który zapewnia niezawodną instalację i ewentualnie deinstalację oprogramo-

wania.

W skład pakietu wchodzą:

- program MRTG,
- skrypty pomocnicze, do wyliczania i odczytu różnych parametrów systemu,
- pliki konfiguracyjne,
- szablon `index.html`.

Wszystkie składowe pakietu zamieszczono w systemie plików, w standardowych katalogach, tak aby nie powodować konfliktów z innym oprogramowaniem. Programy wykonywalne instalują się w `/opt/mrtg/bin`, pliki konfiguracyjne w `/opt/mrtg/etc`, natomiast logi oraz pliki wynikowe ulokowano w `/var/opt/mrtg`, zgodnie z ogólnymi regułami co do rozmieszczenia plików w systemie Unix.

Do monitorowania wybrano następujące cele: SNMP

- analiza ruchu (domyślnie dla dwóch interfejsów).

Firewall

- wszystkie pakiety wychodzące i przychodzące,
- pakiety przychodzące do HTTP,
- pakiety przychodzące do DNS,
- pakiety przychodzące do SMTP.

IP Filter

- tablice stanów NAT i IP,
- zablokowane pakiety.

System

- średnie obciążenie,
- użycie procesora,
- użycie pamięci.

Dyskt

- użycie dysku `/var`,
- użycie dysku `/export/home`.

Sendmail

- przetworzona poczta - ilość wiadomości na godzinę,
- przetworzona poczta - wysłana ilość bajtów.

MySQL

- kwerendy i wątki.

Apache

- ilość żądań na minutę,
- obciążenie procesora.

Dzięki możliwości wstawiania plików konfiguracyjnych jednego w drugi za pomocą dyrektywy `include`, wydzielono część wspólną konfiguracji w pliku `mrtg.conf`, oraz pogrupowano cele tematycznie tak jak na powyższej liście.

3.3 Statystyki MRTG

Wykresy w dalszej części tego rozdziału są przykładami statystyk MRTG z serwera `math`. Statystyki można ogladać na stronie

```
http://math.uwb.edu.pl/mrtg/
```

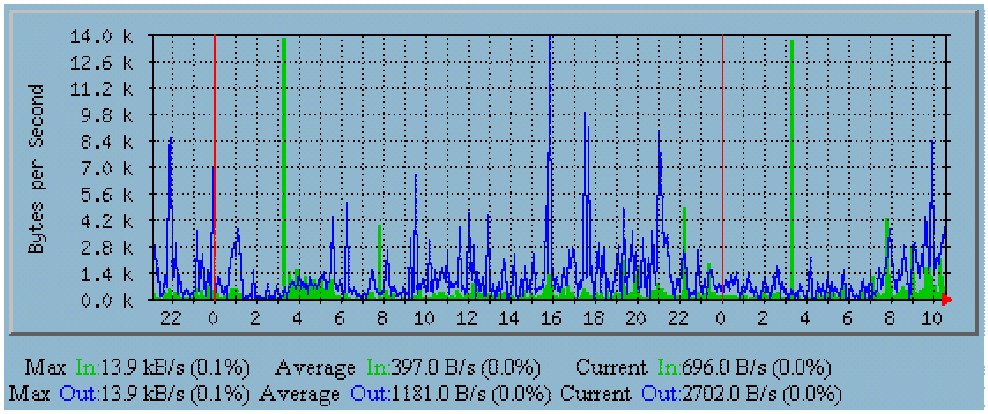
Uznaliśmy, że tego rodzaju statystyki nie powinny być ujawniane publicznie, dlatego też, dostęp do powyższej strony chroniony jest hasłem

Zebrane tutaj wykresy są graficzną ilustracją działania programu MRTG. Na `math` monitorowane są niemal wszystkie parametry systemu, do których odczytu MRTG posiada gotowe skrypty[1]. W praktyce trzeba raczej wybrać najważniejsze parametry biorąc pod uwagę rolę serwera. W ramach tej pracy chcieliśmy z jednej strony sprawdzić możliwości MRTG, z drugiej zobaczyć możliwie najwięcej statystyk różnych parametrów. Tak więc są tu wykresy niosące niezwykle cenne informacje dla administratora o: obciążeniu serwera, użyciu pamięci, zajętości dysków i użyciu tablic stanów IP. Część z prezentowanych wykresów ma jednak charakter ciekawostki.

Wszystkie przedstawione wykresy są dziennymi statystykami poszczególnych celów. Ostatni odczyt znajduje się z prawej strony wykresu. Każdy z wykresów jest tworzony dynamicznie; skala na osi Y zależy od wartości parametrów systemu lub pobranych z dzienników systemowych.

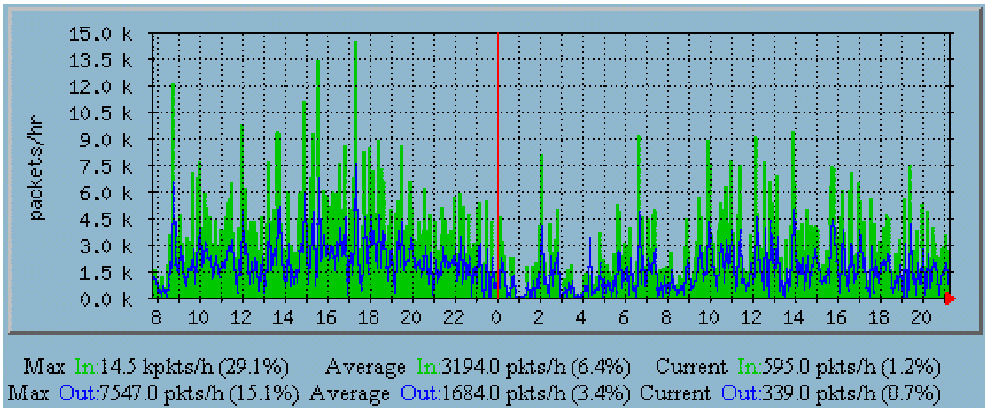
[1] Część z tych skryptów musiała być dostosowana do specyfiki systemu Solaris.

SNMP: Analiza ruchu



Wykres jest graficznym przedstawieniem dziennego ruchu na pierwszym interfejsie (karcie sieciowej). Dane do dziennego wykresu są pobierane co 5 min. Na zielono zaznaczono ilość danych w KB odebranych przez interfejs, a na niebiesko ilość danych wysłanych przez interfejs. Wykres ten daje obraz tego ile w sumie danych przepływa przez serwer, niezależnie od rodzaju tych danych. W przypadku serwera internetowego jest to istotna wskazówka, mówiąca o wykorzystaniu sprzętu i jego obciążeniu. Dzienny wykres pozwala określić godziny najwyższego obciążenia i spokoju, natomiast tygodniowy określa dni na które przypada duży, względnie mały ruch. Z powyższego wykresu wynika, że więcej danych jest wysyłanych niż odbieranych, co jest typowe dla serwera sieciowego. Generalnie ruch wychodzący utrzymuje się na średnim poziomie 2,7 KB/s a wchodzący na poziomie 700 B/s poza pewnymi lokalnymi fluktuacjami. Co ciekawe ruch rozkłada się równomiernie w ciągu całej doby, co oznacza, że komunikacja, z sieciami w naszej strefie czasowej jest na tym samym poziomie, co z innymi strefami czasowymi.

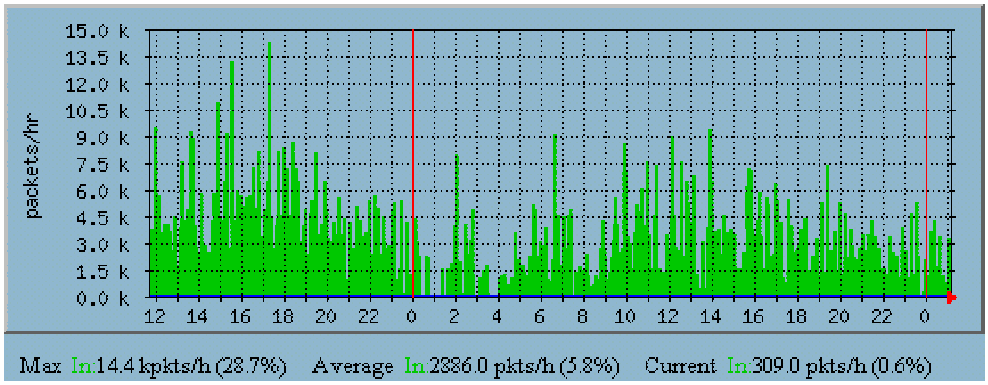
Firewall: Pakiety wychodzące i przychodzące



Wykres obrazuje ilość pakietów które przeszły przez firewall. Dane do dziennego wykresu są pobierane co 5 min. Na zielono zaznaczona jest ilość pakietów

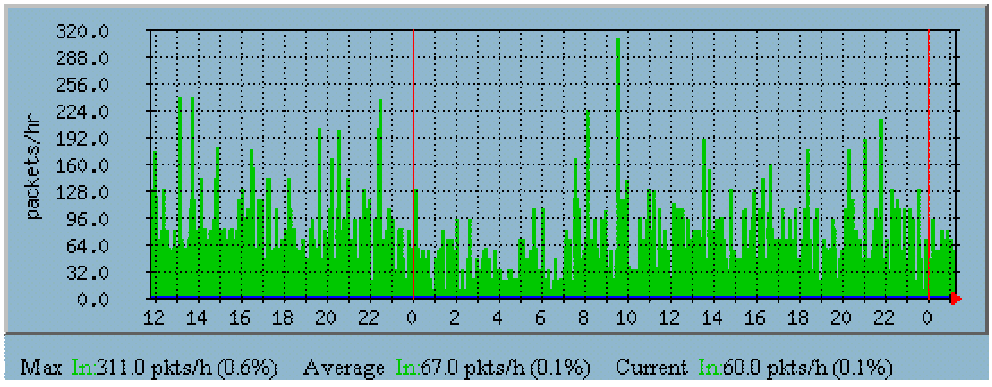
przychodzących, a na niebiesko wychodzących. Z wykresu można łatwo odczytać że średnio ilość pakietów wychodzących z serwera jest większa, niż ilość pakietów do niego przychodzących.

Firewall: Pakiety przychodzące do HTTP



Wykres przedstawia częstość i intensywność łączenia się na port 80 czyli do serwera HTTP. Dane do wykresu są pobierane co 5min. Zielony wykres przedstawia ilość pakietów wysłanych przez użytkowników łączących się ze stronami WWW na danym serwerze. Ilość pakietów wychodzących przez port 80 jest równa 0, gdyż serwer HTTP używa go wyłącznie do odbierania komunikatów, natomiast odpowiedzi udziela zawsze na innym porcie o wysokim numerze. Dlatego też ilość pakietów wychodzących została pominięta. Dzięki temu wykresowi możemy kontrolować czy usługa HTTP na danym serwerze nie jest przeciążona. Jednocześnie jest to źródło informacji dla administratora kiedy może wprowadzać ewentualne poprawki, czyli kiedy usługa HTTP jest najmniej wykorzystywana w ciągu dnia, aby nie przeszkadzać użytkownikom którzy korzystają z HTTP.

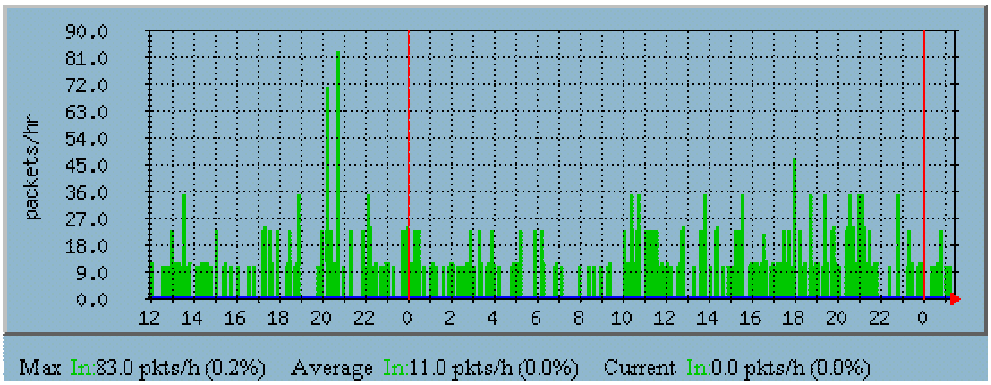
Firewall: Pakiety przychodzące do DNS



Na każdym serwerze jedne usługi są ważniejsze, inne mniej ważne. Monitorujemy te, które uznamy za istotne. Poza HTTP ważnymi usługami są DNS i

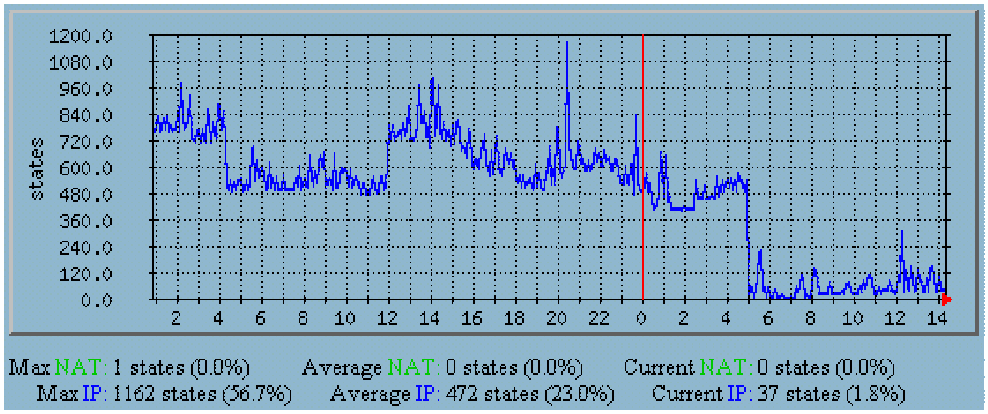
SMTP. Ilość pakietów wychodzących pominięto z tych samych powodów co w przypadku HTTP. Ten wykres obrazuje ilość pakietów przychodzących na port 53, czyli do serwera DNS.

Firewall: Pakiety przychodzące do SMTP



Na wykresie podana jest ilość pakietów odbieranych przez port 25 tzn. przez serwer SMTP.

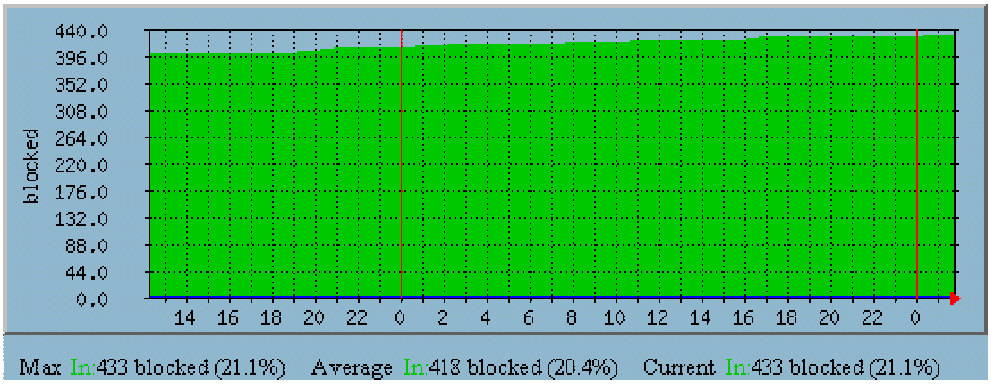
Tablice stanów Nat i IP



W przypadku, gdy firewall jest skonfigurowany tak aby pamiętać stan połączeń z klientami, co podnosi bezpieczeństwo, istotne jest monitorowanie ilości pamięci zajmowanej przez tzw. tablicę stanów IP. Gdy tablica stanów zostanie zapełniona, serwer przestanie odpowiadać na nowe połączenia. Zajętość tablicy stanów IP jest najważniejszym wskaźnikiem firewall'a.

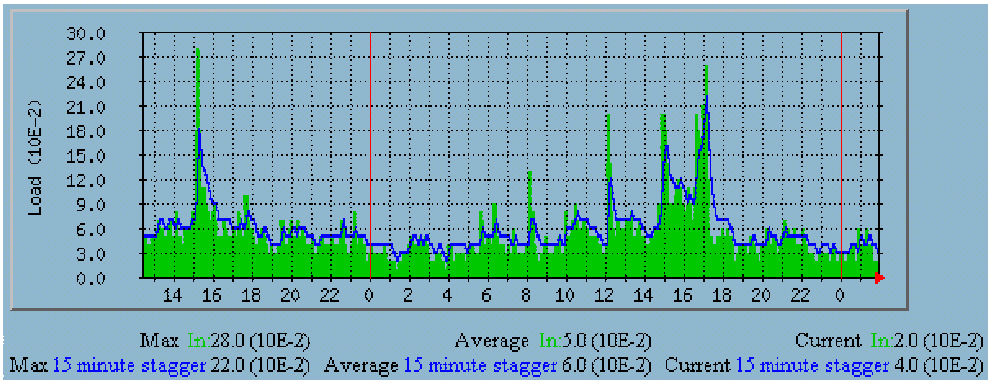
Drugim ważnym wskaźnikiem firewall'a jest zajętość tablic stanów NAT, gdy serwer pełni funkcję translatora adresów (NAT). W Instytucie Matematyki ta funkcja firewall'a nie jest używana, więc odpowiadającą jej część wykresu należy zignorować.

Zablokowane pakiety



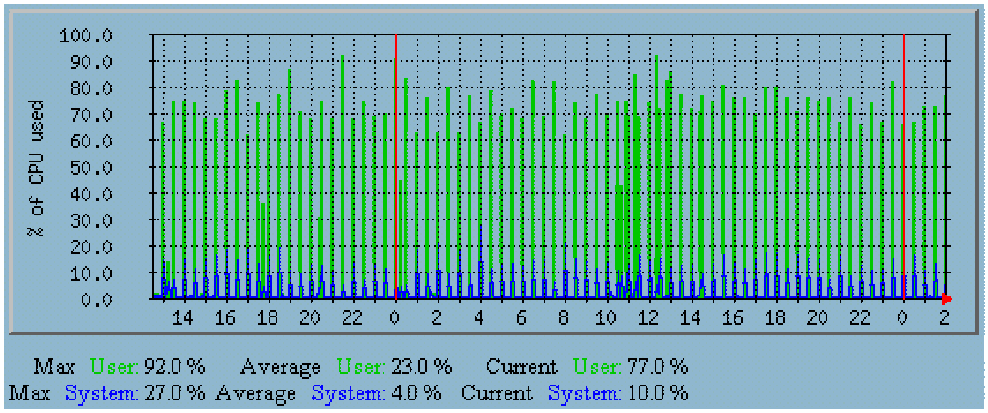
Podstawową funkcją firewall'a jest filtrowanie pakietów, tzn. przepuszczanie tylko "bezpiecznych" i blokowanie pozostałych. Ten wykres obrazuje poziom na jakim utrzymuje się ilość pakietów zatrzymanych przez firewal'a.

System: Średnie obciążenie



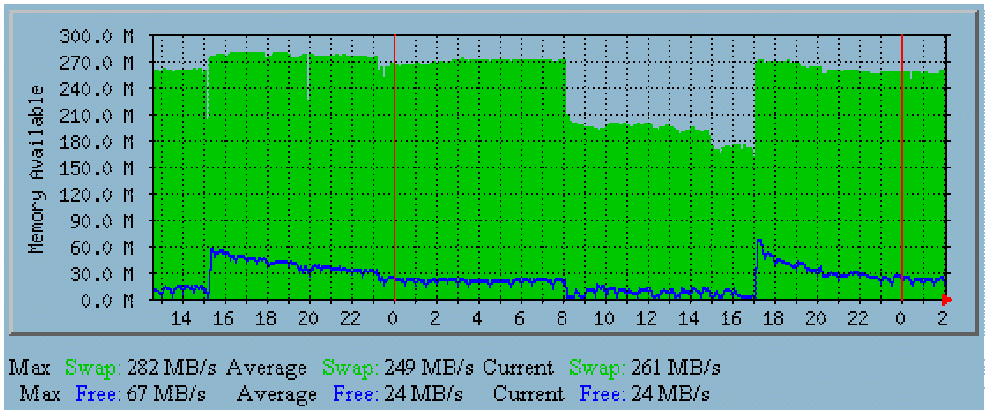
Wykres obrazuje średnie obciążenie całego systemu pracującego na danym serwerze. Jako że zmiany są dość dynamiczne dane do wykresu są pobierane co 5 min. Dzięki wykresowi widać jak system jest obciążony. Jednocześnie wykres ten dostarcza informacji administratorowi o tym, kiedy system jest najmniej, a kiedy najbardziej obciążony. Może okazać się to bezcenne w przypadku instalacji programów które muszą być uruchamiane cyklicznie, na przykład raz dziennie. Wykres ten ułatwi dobranie pory, w której program ma być uruchamiany, tak aby nie powodował przeciążenia systemu. Wykres pomaga również planowanie wyłączeń serwera, w czasie jego najmniejszego użycia.

System: Użycie procesora



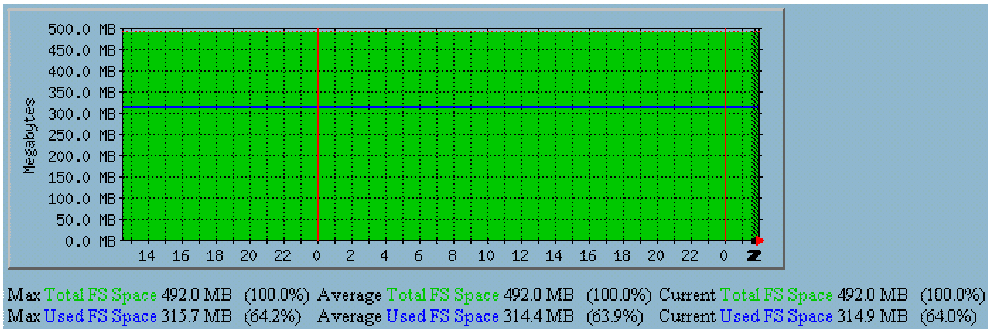
O ile obciążenie mówi o ilości procesów to ten wykres mówi o wykorzystaniu mocy procesorów zainstalowanych na serwerze. 100% oznacza wykorzystanie wszystkich procesorów do maksimum. Moc procesorów serwera `math` jest, jak widać wykorzystywana średnio w 77% równomiernie w ciągu doby. Dane z tego wykresu mogą być przydatne przy zakupie nowego procesora lub przy jego wymianie. Możemy dostosować parametry procesora do zapotrzebowania na danym sprzęcie.

System: Użycie pamięci



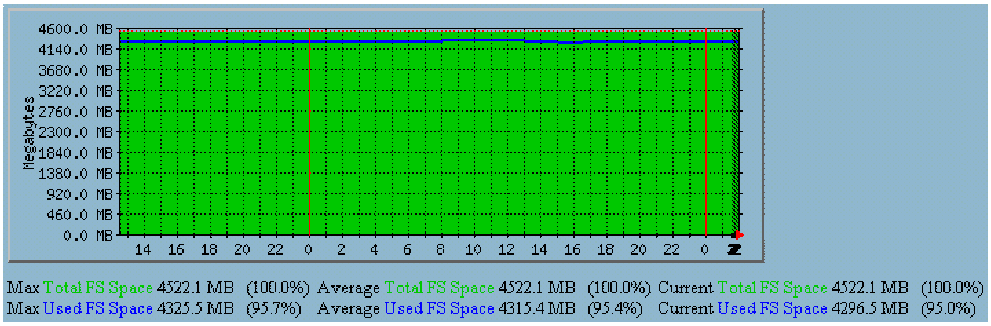
Niebieska linia pokazuje ilość wolnej, fizycznej pamięci RAM. Jak widać z wykresu serwer `math` ma za mało pamięci. Optymalnie średnia wartość wolnej pamięci nie powinna spadać poniżej 128MB, a nie tak jak w tym przypadku gdy wolna przestrzeń oscyluje średnio w granicach 24MB. Zielony wykres pokazuje ilość wolnej pamięci wirtualnej (`swap`). Jak widać partycja `swap` została założona z odpowiednim zapasem i średnio jest około 260MB wolnej pamięci wirtualnej.

Pomiar zajętości dysków: /var



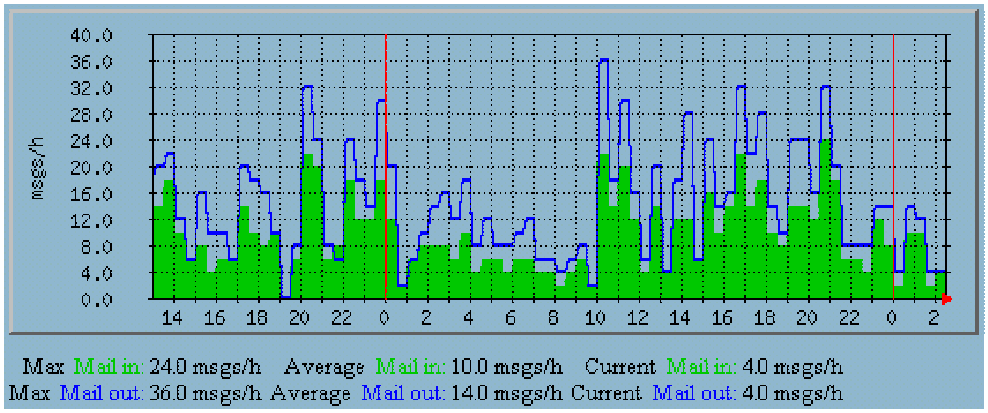
Zielony wykres przedstawia całkowitą pojemność systemu plików `/var`. Widać z niego że na `/var` jest przeznaczone 500MB. Niebieska linia przedstawia ilość wolnego miejsca na `/var`. W `/var` znajdują się pliki które często się zmieniają: dzienniki systemowe, poczta, kolejka poczty i wydruki, stąd nazwa od *variable* – zmienny. Najwięcej miejsca na tym systemie plików zajmuje poczta użytkowników dlatego też kontrolowanie tej przestrzeni jest bardzo ważne, gdyż w razie zapełnienia `/var` poczta przestanie funkcjonować. W tym też celu napisliśmy program `chkmailquota`, dokładny opis tego programu znajduje się w rozdziale 5.1.

Pomiar zajętości dysków: /export/home



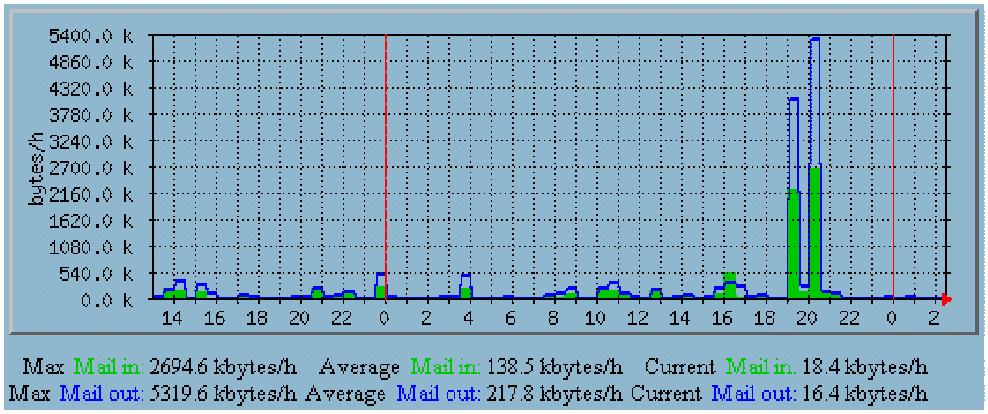
Podobnie jak wyżej zielony wykres przedstawia pojemność całego systemu plików `/export/home` i jest to 4,5GB. Są tam konta użytkowników, ich przeczytana poczta oraz strony internetowe itp. Niebieska linia pokazuje ilość wolnego miejsca na `/export/home`. Łatwo zauważyć że pozostało już niewiele ponad 5% wolnej przestrzeni. By usprawnić pracę administratora napisaliśmy program `chkhomequota` sprawdzający zajętość systemu plików `/export/home` (por. rozdział 5.1).

Sendmail: Ilość wysłanych wiadomości



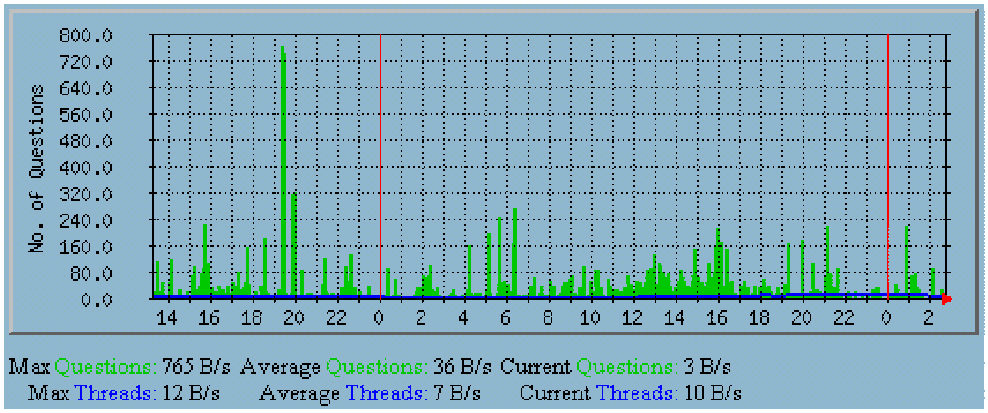
Wykres obrazuje ilość wysłanych wiadomości w ciągu godziny przez serwer pocztowy Sendmail. Wykres ten jest aktualizowany i tworzony co 30 min, po pierwsze by nie obciążać systemu, po drugie ponieważ zmiany jakie zachodzą nie są zbyt dynamiczne i tym samym nie muszą być monitorowane co 5min. Dzięki wykresowi administrator wie, kiedy użytkownicy najczęściej wysyłają pocztę, o jakich porach dnia. Administrator porównując wykresy może zadecydować, kiedy należy przeprowadzać prace związane z Sendmailem, aby jak najmniej utrudnić korzystanie z poczty użytkownikom. Zielony wykres pokazuje ile wiadomości przyszło do serwera, natomiast niebieska linia obrazuje ilość wiadomości, która została wysłana przez użytkowników.

Sendmail: Ilość wysłanych bajtów



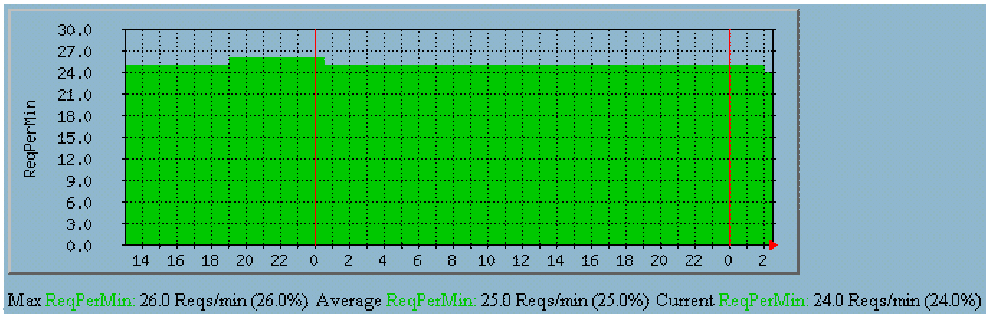
Z tego wykresu można odczytać ile poczty wysyła serwer w kilobajtach na godzinę (kbytes/h). Obraz ten jest tworzony co 30 min z takiego samego powodu jak w przypadku poprzednim. Niebieska linia przedstawia ilość kilobajtów wysłanych z serwera, natomiast zielony wykres obrazuje ilość kilobajtów które doszły do serwera w postaci poczty.

MySQL: Kwerendy i wątki



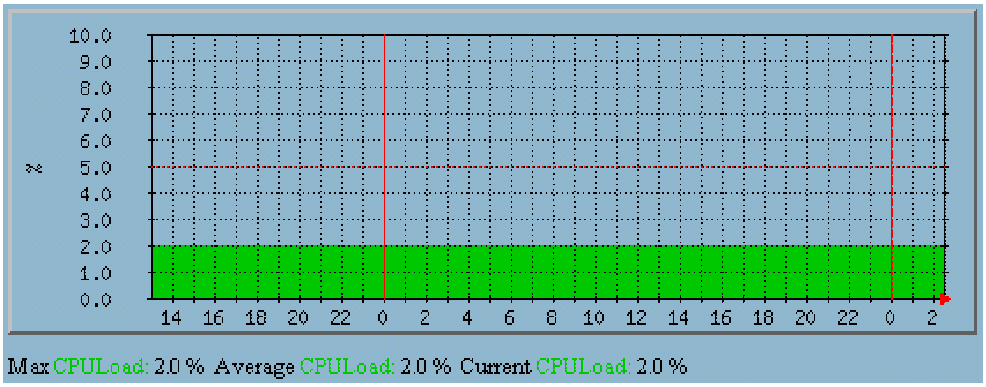
Na serwerze `math` umieszczone są strony WWW, do których dane pobierane są z bazy danych MySQL. Wykres pokazuje ile średnio kwerend dostaje serwer MySQL w ciągu sekundy. Jednocześnie spływające do MySQL kwerendy obsługiwane są przez serwer jako osobne wątki. Z wykresu można odczytać ile takich wątków powstaje średnio w ciągu sekundy.

Apache: Żądań na minutę



Jedną z głównych usług na serwerze `math` jest WWW. W tej sytuacji warto mieć na bieżąco podgląd na to, jak pracuje serwer HTTP. Na tym wykresie można odczytać ile zapytań średnio w ciągu minuty dostaje serwer Apache.

Apache: Obciążenie procesora



Wykres przedstawia tylko jedną zmienną zaznaczoną na zielono, która pokazuje w jakim stopniu Apache obciąża procesor serwera. Wartość jest opisywana w procentach. Z wykresu można odczytać, że Apache wykorzystuje średnio procesor w 2%. Tak niewielkie wykorzystanie procesora przez Apache'a jest spowodowane tym, że większość stron na serwerze `math` jest dynamiczna i właściwie cały ciężar obsługi żądań spada na Meta-HTML.

Rozdział 4

Webalizer

Webalizer [13] jest to program służący do tworzenia statystyk w oparciu o informacje zapisane w logach. Statystyki te tworzone są okresowo. Na serwerze `math`, przy pomocy programu Webalizer, tworzone są statystyki serwerów HTTP i FTP. Statystyki te umożliwiają administratorowi sprawdzenie w jakim stopniu dana usługa jest wykorzystywana, kto z danych usług najczęściej korzysta i w jakim czasie (godziny, dni) określone usługi są najbardziej eksploatowane. W statystykach występują następujące typy informacji:

Hits — ilość wszystkich wywołań jakichkolwiek elementów z serwera w określonym czasie,

Files — ilość wszystkich wysłanych do użytkowników plików (tekstowych i binarnych) z serwera,

Sites — ilość unikalnych adresów IP, z których nastąpiło połączenie do serwera,

Visits — orientacyjna ilość odwiedzin strony; jako jedna wizyta liczą się wszystkie odwołania do danego serwera HTTP jakie nastąpiły w ciagu 30 minut z jednego adresu w sieci, po upływie tego czasu, kolejne odwołanie taktowane jest jako nowa wizyta,

Pages — ilość otwartych stron `.htm` i `.html` z danego serwera.

4.1 Statystyki HTTP

Statystyki odwiedzin strony internetowej Instytutu Matematuki UwB znajdują się pod adresem

```
http://math.uwb.edu.pl/usage
```

Są one generowane programem Webalizer, co niedziela rano o godzinie 3:30. Statystyki te są dobrym źródłem informacji dla opiekuna strony jakie informacje są poszukiwane, skąd następuje wejście na strony Instytutu, kto z serwisu WWW korzysta i w jakich porach dnia i tygodnia.

Poniżej przedstawione są przykłady wykresów i tabel ze wspomnianej strony, za miesiąc maj bieżącego roku.

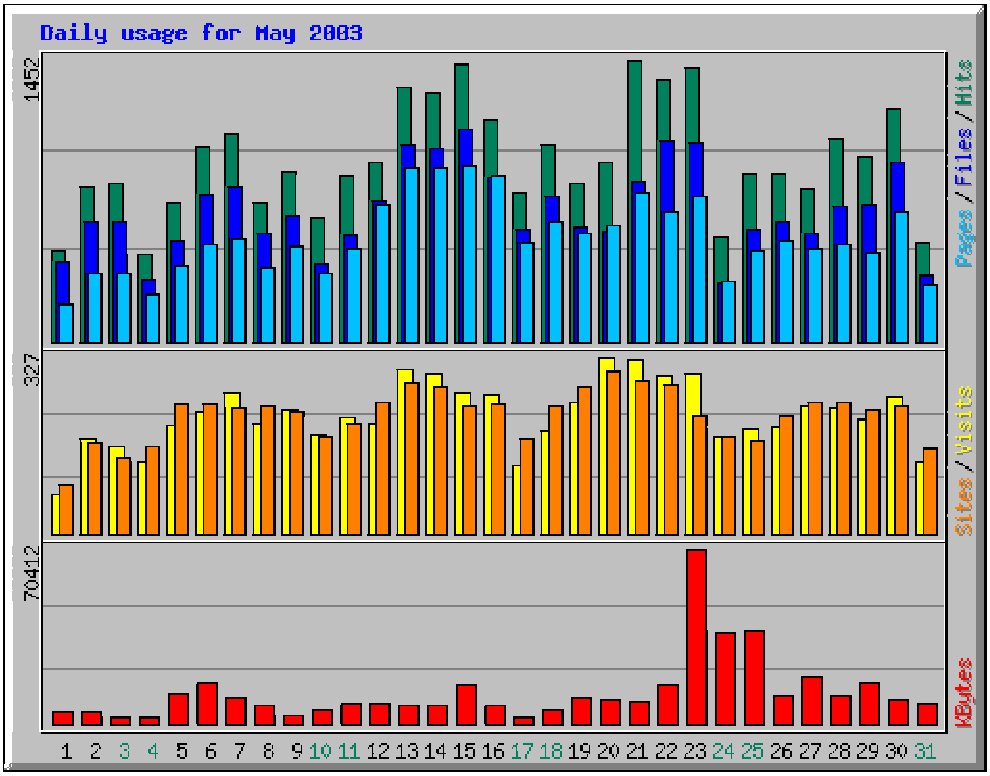
Miesięczne statystyki za Maj 2003

Monthly Statistics for May 2003	
Total Hits	**28965**
Total Files	**20933**
Total Pages	**16653**
Total Visits	**6818**
Total KBytes	**399598**

Tabela obrazuje informacje zebrane z całego miesiąca, kolejno opisując jest to suma:

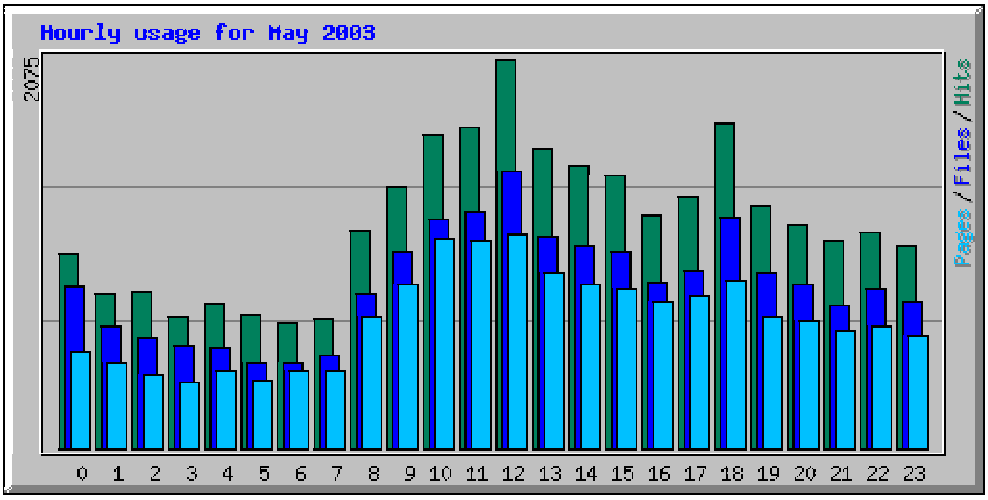
- Wywołań wszystkich elementów z serwera - 28965
- Pobranych plików - 20933
- Otwartych stron - 16653
- Orientacyjnej ilości osób oglądających strony na `math` - 6818
- Wszystkich pobranych plików 399598 KB czyli około 390MB

Dzienne statystyki za Maj 2003



Jest to wykres przedstawiający dzienny rozkład parametrów określających ruch w usłudze HTTP. Można określić jakie jest obciążenie usługi w poszczególnych dniach. Wartości pobierane do wykresu są średnimi wartościami przypadającymi na dane dni z całego miesiąca. Z pierwszego wykresu słupkowego można odczytać w jakich dniach najczęściej są otwierane strony (Pages), pobierane pliki (Files), i wywoływane jakiekolwiek żądań z serwisu (Hits). Na środkowym wykresie jest pokazana ilość unikalnych adresów IP z których nastąpiło połączenie (Sites) oraz orientacyjna ilość osób odwiedzających stronę(Visits). Na ostatnim wykresie można zaobserwować w których dniach jest pobierana największa ilość KB (KByte).

Godzinne statystyki za Maj 2003



Jest to godzinny wykres obrazujący ilość: otwartych stron (Pages), pobranych plików (Files), oraz jakichkolwiek wywołań z serwera (Hits). Wykres pokazuje jakie wartości przyjmują te parametry w poszczególnych godzinach, średnio w danym miesiącu. Szczególnie pomocny administratorowi może być ten wykres przy próbie naprawy lub pracy nad usługą HTTP, gdzie występuje potrzeba jej wyłączenia, dzięki któremu może dostosować swa prace tak aby jak najmniej przeszkodzić użytkownikom.

Najpopularniejsze strony

Top 30 of 437 Total URLs					
#	Hits		KBytes		URL
1	2198	7.59%	183	0.05%	/pl/html/
2	1740	6.01%	859	0.21%	/server-status
3	1729	5.97%	18	0.00%	/pl/html/staff/vitae.mhtml
4	840	2.90%	82	0.02%	/robots.txt
5	661	2.28%	291	0.07%	/~mariusz/multiboot/
6	555	1.92%	11	0.00%	/pl/html/structure/department.mhtml
7	363	1.25%	0	0.00%	/pl/html/research-desc.mhtml
8	323	1.12%	895	0.22%	/~lapinski/
9	320	1.10%	87	0.02%	/~knmism/
10	308	1.06%	18	0.00%	/pl/html/staff/vitae-publ.mhtml
11	304	1.05%	13597	3.40%	/spnm/program.html
12	273	0.94%	29	0.01%	/pl/html/staff/
13	220	0.76%	40502	10.14%	/~kotowicz/W0203AM1.pdf
14	212	0.73%	662	0.17%	/~kotowicz/wyklady.html

Jest to wycinek tabeli pokazujący najczęściej oglądany strony na serwerze `math`. URL jest końcówką adresu, czyli na przykład najczęściej oglądaną stroną jest `math.uwb.edu.pl/pl/html`. Informacje są pomocne przy tworzeniu stron, dzięki wykresowi widać które strony są najbardziej popularne, co pozwala webmaster'owi rozwijać serwis HTTP tak aby był jak najczęściej oglądany i aby użytkownicy mogli znaleźć jak najwięcej potrzebnych informacji.

Najpopularniejsze strony ze względu na KB

Top 10 of 437 Total URLs By KBytes

#	Hits		KBytes		URL
1	220	0.76%	40502	10.14%	/~kotowicz/W0203AM1.pdf
2	2	0.01%	28141	7.04%	/~artutk/tajnauc.arj
3	189	0.65%	17713	4.43%	/~kotowicz/W0203SM.pdf
4	304	1.05%	13597	3.40%	/sprnm/program.html
5	96	0.33%	12218	3.06%	/~kotowicz/W0203TMC.pdf
6	97	0.33%	11386	2.85%	/~kotowicz/W0203RP.pdf
7	23	0.08%	5809	1.45%	/~kotowicz/W0203AM1.dvi
8	5	0.02%	4387	1.10%	/sprznm/prezentacjappt.zip
9	9	0.03%	4180	1.05%	/sprznm/html/schole/9/schole9.pdf
10	12	0.04%	3034	0.76%	/pl/html/education/informator.ps

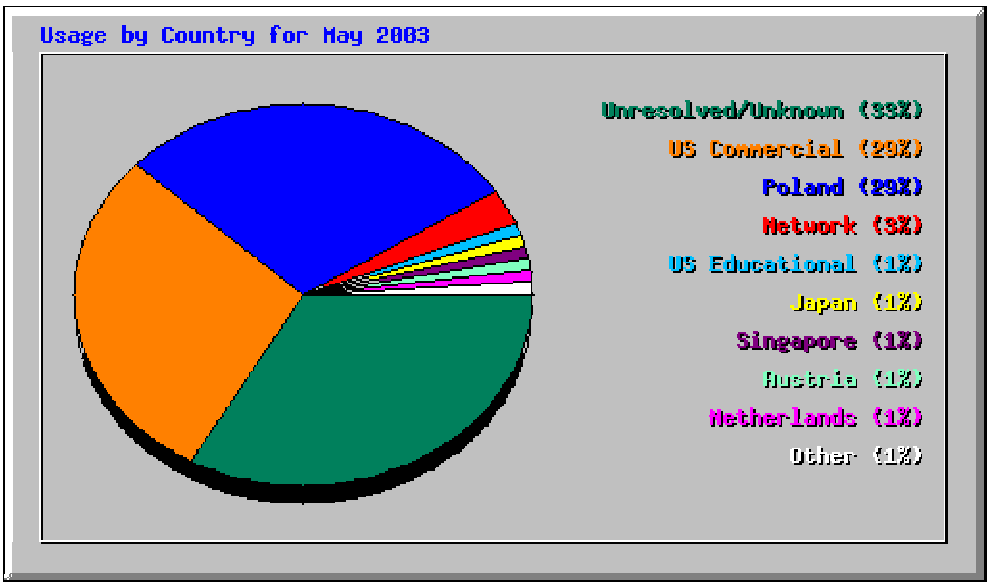
Tabela ta pokazuje strony, z których jest pobierana największa ilość danych w KB. Widać że na pierwszym miejscu jest strona dr Jarosława Kotowicza. Jako że maj jest to ostatni miesiąc przed sesją widać że studenci pobierają ze strony doktora pliki z wykładami, przykładowymi egzaminami itp. Są to pliki tekstowe najczęściej z rozszerzeniem `.pdf`. Jako że są to pliki o dość dużej pojemności to jest powodem tego że właśnie te strony są na pierwszych miejscach.

Strony początkowe

Top 10 of 136 Total Entry Pages					
#	Hits		Visits		URL
1	2198	7.59%	525	13.41%	/pl/html/
2	661	2.28%	523	13.36%	/~mariusz/multiboot/
3	304	1.05%	278	7.10%	/spnm/program.html
4	323	1.12%	270	6.90%	/~lapinski/
5	202	0.70%	183	4.67%	/groups2001/hotelg.html
6	320	1.10%	133	3.40%	/~knmism/
7	138	0.48%	115	2.94%	/~mariusz/dualboot/
8	212	0.73%	111	2.84%	/~kotowicz/wyklady.html
9	165	0.57%	110	2.81%	/~trybulec/
10	155	0.54%	106	2.71%	/sprzrm/html/schole/9/

Z tej tabeli można odczytać jakie najczęściej adresy są wpisywane w pole URL przeglądarki internetowej. Czyli z jakimi adresami najczęściej użytkownicy łączą się z serwisem HTTP na serwerze `math`. Może być to ważna informacja, gdy chcemy ustalić która strona powinna być wyświetlana jako pierwsza po wpisaniu w pole ULR `math.uwb.edu.pl`. Ale widzimy że najczęściej otwierana jest strona główna. Może to być też wskazówka dla webmaster'a aby umieścić odnośniki do tych stron na stronie głównej, lub w łatwo dostępnym miejscu.

Kraje - wykres



Ten wykres kołowy przedstawia państwa z których najczęściej zostaje nawiązane połączenie z usługą HTTP. Widać że najwięcej połączeń następuje z sieci

których nie można ustalić Państwa (Unresolved/Unknown), Następne w kolejności są adresy komercyjne czyli z końcówką `.com`. 29% osób które oglądają stronę matematyki to Polacy. Pozostała część osób należy do innych nacji, takich jak Japonia, Singapur, Australia itd.

Kraje - tabela

Top 30 of 73 Total Countries							
#	Hits		Files		KBytes		Country
1	9423	32.53%	7175	34.28%	66974	16.76%	Unresolved/Unknown
2	8499	29.34%	6007	28.70%	18366	4.60%	US Commercial
3	8300	28.66%	6320	30.19%	304444	76.19%	Poland
4	868	3.00%	555	2.65%	2879	0.72%	Network
5	281	0.97%	168	0.80%	1018	0.25%	US Educational
6	199	0.69%	167	0.80%	220	0.05%	Japan
7	192	0.66%	15	0.07%	47	0.01%	Singapore
8	178	0.61%	9	0.04%	23	0.01%	Austria
9	163	0.56%	80	0.38%	128	0.03%	Netherlands
10	99	0.34%	20	0.10%	13	0.00%	France
11	77	0.27%	54	0.26%	579	0.14%	Germany
12	71	0.25%	63	0.30%	598	0.15%	Canada
13	46	0.16%	1	0.00%	13	0.00%	Mexico
14	43	0.15%	14	0.07%	68	0.02%	Australia

Tabela jest bardziej szczegółowym przedstawieniem powyższego wykresu kołowego. Podane są tutaj: ilość żądań (Hits), ilości pobranych plików (Files), oraz ilości pobranych danych w KB (KBytes). W kolumnie Country znajduje się nazwa państwa, z którego pochodzą żądania do serwera. Te informacje pozwalają ocenić opiekunowi serwisu WWW, jakie wersje językowe strony warto przygotować.

Oprócz wyżej pokazanych przykładów w statystykach można znaleźć informacje o tym:

- które strony są przez użytkowników jako ostatnie oglądane czyli na których stronach kończą przeglądanie stron z serwisu,
- z których stron nastąpiło przekierowanie na stronę IM i tu wiedzie prym wyszukiwarka internetowa google,
- z jakich adresów (z pod jakiego IP) najczęściej następuje połączenie z usługą HTTP,
- na jakich stronach znajdują się odnośniki do strony `math.uwb.edu.pl`,

- jakie słowa kluczowe są wpisywane w wyszukiwarki, dzięki którym te przekierowują użytkowników na stronę IM, są to na przykład: Pitagoras, Tales, multiboot, solaris, uwb itd.
- jakimi przeglądarkami są oglądane strony na serwerze `math` (Mozilla, Internet Explorer).

Szczególnie istotne są dwa ostatnie punkty, ponieważ różnymi technikami robi się strony pod różne przeglądarki, a w źródło strony wpisuje się słowa kluczowe, co ułatwia wyszukiwarkom oddanie użytkownikowi jak najbardziej trafnej odpowiedzi. Czy – krótko mówiąc – administrator i webmaster (w Instytucie Matematyki funkcję tę pełni ta sama osoba) oglądając te statystyki może odpowiednio dostosować stronę do wymagań i zapotrzebowania użytkowników.

4.2 Statystyki FTP

Przy pomocy programu Webalizer można wygenerować nie tylko statystyki serwera HTTP ale również serwera FTP, o ile tylko oprogramowanie serwera tworzy dzienniki o standardowej budowie. Są dwa standardy dzienników używanych przez serwery FTP: xferlog zdefiniowany przez twórców WUftpd oraz CLF (Common Logfile Format) zgodny z formatem logów Apache.

Na serwerze `math` jako serwer FTP używane jest oprogramowanie ProFTPD [12]. Dokładny opis jego dzienników podany jest w 1.9. Statystyki generowane są przy pomocy programu Webalizer na podstawie dziennika `xferlog` i można je oglądać pod adresem:

`http://math.uwb.edu.pl/ftpusage`

Poniżej przedstawiono kilka tabel i wykresów spośród statystyk dla serwera `math` za miesiąc maj bieżącego roku. Ponieważ użyto tego samego oprogramowania Webalizer, co w przypadku serwera Apache, sens statystyk jest bardzo podobny do tego z 4.1.

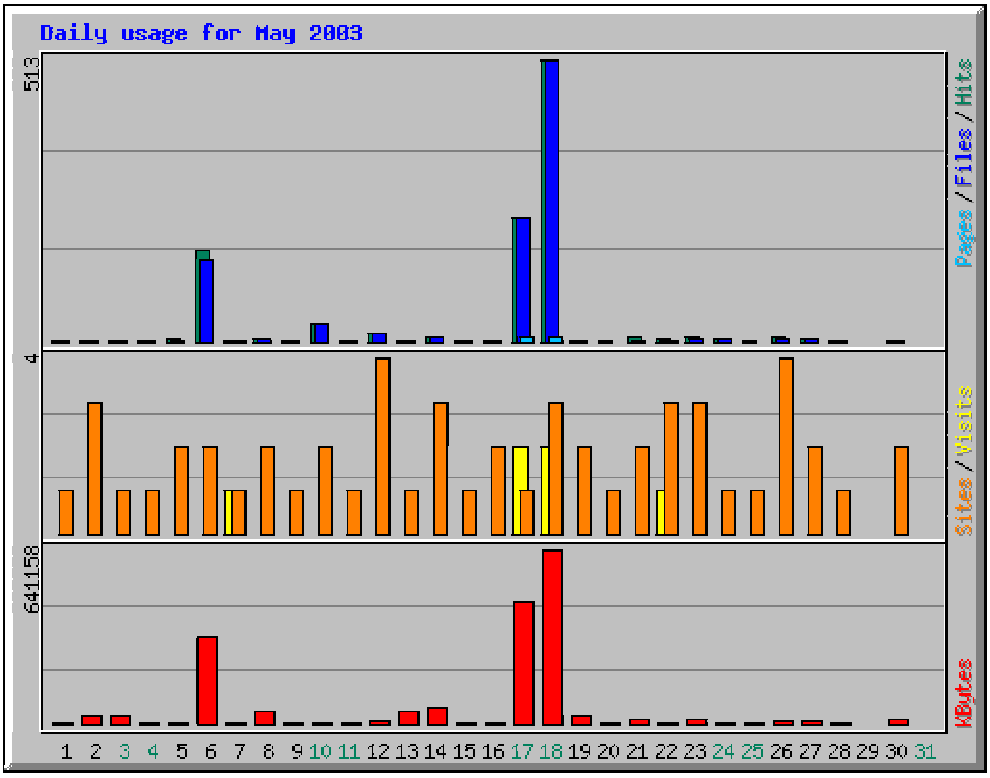
Miesięczne statystyki za Maj 2003

Monthly Statistics for May 2003	
Total Hits	1040
Total Files	999
Total Pages	19
Total Visits	6
Total KBytes	1763893

W tabeli tej znajduje się podsumowanie całego miesiąca. Kolejno mamy:

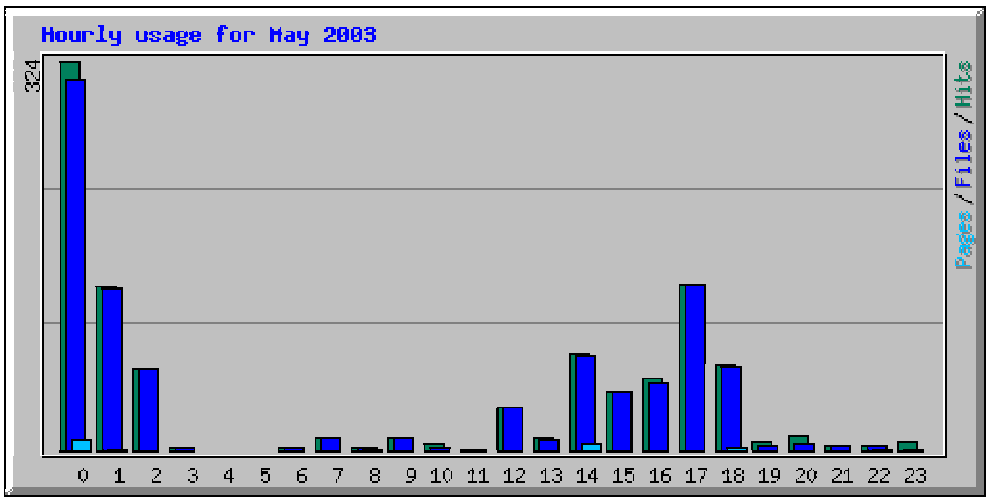
- sumę odwołań do plików na serwerze: 1040,
- liczbę pobranych plików: 999,
- pobranych plików o charakterze tekstowym (`.txt`, `.html`): 19,
- orientacyjną ilość osób oglądających zawartość serwera FTP na `math`: 9,
- łączny rozmiar wszystkich pobranych plików: 1763893KB, czyli około 1,7 GB.

Dzienne statystyki za Maj 2003



Na tym wykresie pokazano średni, dzienny rozkład wartości opisujących ruch na serwerze FTP. Warto zwrócić tutaj uwagę na to, że o ile liczba odwiedzających jest stała w ciągu całego miesiąca, to 17-tego i 18-tego została pobrana wyraźnie większa ilość danych przez FTP niż w pozostałe dni.

Godzinne statystyki za Maj 2003



Wykres przedstawia jak zmieniają się ilość pobieranych plików przez FTP średnio w ciągu dnia. Ruch na FTP wzmaga się w godzinach 14-18. Ten wykres i poprzedni pomagają administratorowi planować różne czynności konserwacyjne związane z FTP.

Biorąc pod uwagę poprzedni wykres, duża ilość pobranych plików w godzinach 0-3 świadczy o tym, że 17-tego - 18-tego maja, w godzinach 0-3 pobrano ogromną ilość plików z FTP.

Najczęściej pobierane pliki

Top 30 of 554 Total URLs

#	Hits		KBytes		URL
1	7	0.67%	7	0.00%	/solaris/common/5.8/plkbd-1.1-Solaris8.pkg.tgz
2	7	0.67%	41146	2.33%	/solaris/x86/5.8/gpc-20011202-Solaris8-x86.pkg.tgz
3	6	0.58%	6	0.00%	/solaris/x86/5.8/README
4	5	0.48%	2397	0.14%	/solaris/x86/5.8/mrtg-2.9.29-Solaris8-i686.pkg.tgz
5	5	0.48%	230704	13.08%	/solaris/x86/5.8/xfree86-4.2.0-Solaris8-x86.pkg.tgz
6	4	0.38%	7308	0.41%	/solaris/x86/5.8/apache-1.3.26+ssl-Solaris8-i486.pkg.tgz
7	4	0.38%	2126	0.12%	/solaris/x86/5.8/ftype2-2.0.1-Solaris8-x86.pkg.tgz
8	4	0.38%	1488	0.08%	/solaris/x86/5.8/gd-2.0.1-Solaris8-x86.pkg.tgz
9	4	0.38%	15030	0.85%	/solaris/x86/5.8/gpc-2.1-Solaris8-x86.pkg.tgz
10	4	0.38%	1706	0.10%	/solaris/x86/5.8/libjpeg-6b-Solaris8-x86.pkg.tgz

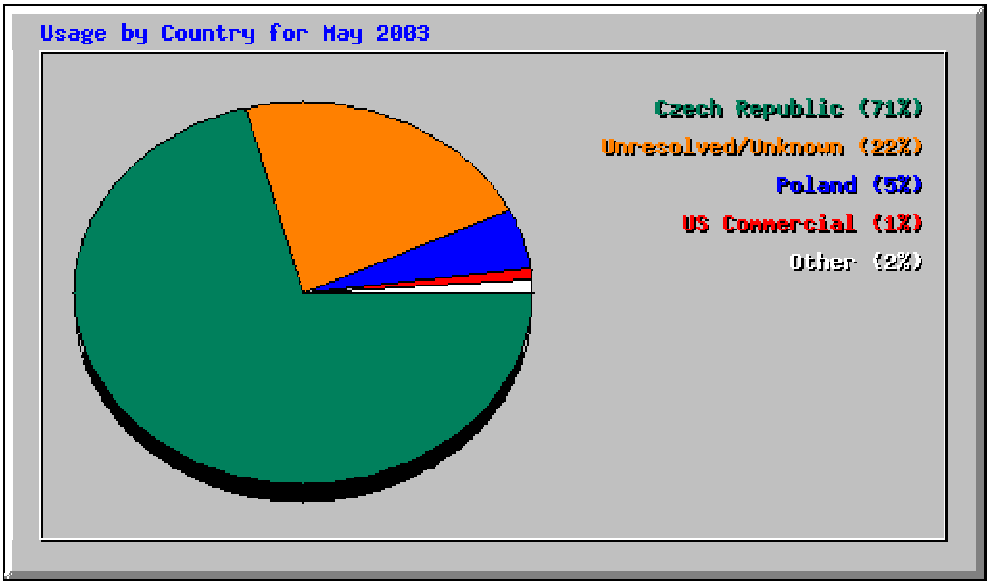
Ten fragment tabeli pokazuje 10 najczęściej pobieranych plików z serwera `math`. W kolumnie HITS podana jest ilość udanych transferów plików, w kolumnie KBytes widać łączny rozmiar tych transferów, a w kolumnie URL jest podana pełna ścieżka z nazwą pliku.

Najczęściej pobierane pliki ze względu na wielkość

Top 10 of 554 Total URLs By KBytes					
#	Hits		KBytes		URL
1	5	0.48%	230704	13.08%	/solaris/x86/5.8/xfree86-4.2.0-Solaris8-x86.pkg.tgz
2	4	0.38%	78744	4.46%	/solaris/x86/5.8/mozilla-1.4a-Solaris8-i686.pkg.tgz
3	1	0.10%	74148	4.20%	/solaris/companion/sparc/5.8/kde-2.2.1-SFW.pkg.gz
4	2	0.19%	73873	4.19%	/solaris/companion/i386/5.8/kde-2.2.1-SFW.pkg.gz
5	2	0.19%	50407	2.86%	/solaris/x86/5.8/mozilla-1.3-Solaris8-i686.pkg.tgz
6	7	0.67%	41146	2.33%	/solaris/x86/5.8/gpc-20011202-Solaris8-x86.pkg.tgz
7	2	0.19%	39772	2.25%	/solaris/companion/i386/5.8/kde-2.0.1-SFW.pkg.gz
8	2	0.19%	34576	1.96%	/solaris/common/5.8/texmf-1.0.7-solaris8.pkg.tgz
9	4	0.38%	34257	1.94%	/solaris/x86/5.8/mplayer-0.90pre8-Solaris8-i686.pkg.tgz
10	1	0.10%	28845	1.64%	/win32/java/j2sdk-1_3_1_01-win.exe

Tabela ta pokazuje pliki, których transfer w miesiącu maj dał największą sumę wysłanych bajtów. O miejscu pliku na liście nie decyduje jego rozmiar ani częstość pobierania, ale oba te parametry naraz.

Kraje - wykres



Ten wykres kołowy przedstawia państwa, z których najczęściej były ściągane pliki z serwisu FTP. Widać że w maju najwięcej danych było pobieranych z Czech 71%. Następne w kolejności są adresy z sieci, których przynależności do konkretnego państw nie można ustalić (Unresolved/Unknown) 22%, natomiast trzecim krajem w kolejności jest Polska 5%.

Kraje - tabela

Top 12 of 12 Total Countries							
#	Hits		Files		KBytes		Country
1	737	70.87%	736	73.67%	1078665	61.15%	Czech Republic
2	224	21.54%	219	21.92%	352520	19.99%	Unresolved/Unknown
3	52	5.00%	26	2.60%	168201	9.54%	Poland
4	11	1.06%	6	0.60%	69620	3.95%	US Commercial
5	5	0.48%	5	0.50%	76998	4.37%	Network
6	3	0.29%	2	0.20%	2856	0.16%	Germany
7	2	0.19%	1	0.10%	7568	0.43%	Non-Profit Organization
8	2	0.19%	1	0.10%	1474	0.08%	Netherlands
9	1	0.10%	1	0.10%	1	0.00%	Switzerland
10	1	0.10%	0	0.00%	112	0.01%	Italy
11	1	0.10%	1	0.10%	1	0.00%	Japan
12	1	0.10%	1	0.10%	5878	0.33%	Taiwan

Tabela ta daje więcej szczegółów na ten sam temat co wykres kołowy w poprzednim podpunkcie. Mamy tutaj ilość odwołań do plików (Hits), ilość pobranych plików (Files), oraz ilości pobranych danych w KB (KBytes).

Rozdział 5

Programy

5.1 Kontrola zajętości dysków

Jednym z zadań administratora, poza analizą zawartości różnego rodzaju logów, jest pilnowanie zajętości dysków. Nigdy nie ma niestety na tyle komfortowej sytuacji, aby zignorować ograniczenia wynikające z rozmiaru twardych dysków, tak jak nie istnieją dyski, których nie można zapełnić. W systemie Solaris, ale podobnie też w innych systemach Unix, fizyczne dyski podzielony jest na logiczne *systemy plików*, czasem zwane *partycjami*. Dobrym zwyczajem, w przypadku serwera, jest wydzielenie następujących systemów plików:

`/` — (root) wystarczy niewielki przydział miejsca, tutaj głównie będą gromadzone pliki konfiguracyjne oraz montowane inne systemy plików.

`/var` — tutaj przechowywane są: poczta, logi, kolejki drukowania. Rozmiar tego systemu plików uzależniony jest od liczby użytkowników i przeznaczenia serwera. Na serwerze wydruków i serwerze pocztowym powinien to być możliwie duży system plików.

`/usr` — system plików prawie wyłącznie do odczytu. Tutaj zainstalowane jest całe oprogramowanie systemu.

`/export/home` — w zależności od tego czy na serwerze zakładane są konta Shell'owe, czy też nie, rozmiar tego systemu plików powinien być odpowiednio duży.

Wydzielenie powyższych niezależnych obszarów w przypadku np. zapełnienia `/export/home` nie spowoduje zawieszenia pracy programów korzystających z `/var`, w tym serwera pocztowego czy też serwera wydruku mimo, że użytkownicy nie zapiszą już żadnego pliku.

Najintensywniej zmieniającym się system plików, jak wskazuje na to nazwa (od ang. variable), jest `/var`. Znacznie wolniej w czasie zmienia się `/export/home`, ale ma tendencję to systematycznego kurczenia się, a to dlatego, że użytkownicy częściej zapisują niż usuwają pliki. Te dwa systemy plików należy regularnie

kontrolować. Wystarczy do tego standardowe polecenie systemu `df -k`. Daje ono jednak tylko zbiorczą informacje o zajętości wszystkich zamontowanych systemów plików.

O ile w przypadku logów możemy zastosować automatyczną rotację, co uchroni nas od przepełnienia systemu plików, to w przypadku poczty, kolejek i katalogów użytkowników można jedynie zastosować tzw. *quota*, które jedynie w bierny sposób nie dopuści do zajęcia większego niż wyznaczony obszar. Dobrym rozwiązaniem wydaje się regularne sprawdzanie rozmiarów skrzynek pocztowych i kont użytkowników. Jest to wystarczające działanie prewencyjne. Aby je maksymalnie uprościć napisałem dwa skrypty Perl'owe:

`chkmailquota` — do kontroli rozmiarów skrzynek pocztowych oraz,

`chkhomequota` — do kontroli rozmiarów katalogów użytkowników.

Oba programy działają bardzo podobnie. Pobierają ten sam zestaw argumentów:

```
program [-nq] [size]
```

o następującym znaczeniu:

`size` — limit rozmiaru skrzynki pocztowej, względnie katalogu domowego, podany w bajtach. Jest to parametr opcjonalny i nie podanie go spowoduje użycie limitu domyślnego, który dla `chkmailquota` wynosi 5MB, a dla `chkhomequota` 45MB.

`n` — (ang. notify) do użytkowników którzy przekroczyli limit zostanie wysłany e-mail informujący o tym fakcie, z podaniem wielkości sumarycznego rozmiaru plików i o ile przekroczono limit. Zbiorcza informacja, łącznie z wynikiem polecenia `df -k`, zostaje dodatkowo wysłana na adres administratora systemu.

`q` — (ang. quiet) bez podania tej opcji program wypisuje wynik w oknie terminala, gdzie go uruchomiono. Jeśli opcja zostanie użyta na ekran nie będzie wypisana żadna informacja. Opcja jest przydatna gdy program jest uruchamiany z tablicy `crontab`.

5.2 Rotacja logów ProFTPD

Przy okazji tworzenia statystyk FTP (por. rozdział 4.2) okazało się, że od czasu uruchomienia ProFTPD jego logi nie były poddane rotacji i osiągnęły dość duże rozmiary. W związku z tym uznaliśmy, że należy napisać skrypty, które będą wołane regularnie przez demon `cron` w celu wykonania rotacji i jednocześnie wygenerowania statystyk.

processftpdlogs

W działaniu skryptu można wydzielić następujące kroki:

1. zatrzymanie serwera ProFTPD,
2. przeniesienie wszystkich trzech logów: `access.log`, `auth.log`, `xferlog`, do podkatalogu `logs`, dopisanie daty w nazwach plików: `access.log`, `auth.log`,
3. uruchomienie serwera ProFTPD, w tym momencie program ProFTPD tworzy wszystkie trzy logi na nowo,
4. wygenerowanie statystyk przy pomocy programu Webalizer na podstawie loga `logs/xferlog`,
5. dopisanie daty w nazwie pliku `logs/xferlog`,
6. kompresja programem `gzip` ostatnio zarchiwizowanych logów w katalogu `logs`.

Kod skryptu zamieszczono na stronie 56.

processftpdlogs-ny

W programie Webalizer jest ograniczenie polegające na tym, że utworzyć można statystyki tylko za ostatnie 12 miesięcy. Dlatego też dzielimy statystyki na poszczególne lata. Służy do tego program `processftpdlogs-ny`.

Skrypt `processftpdlogs-ny` wykonuje najpierw te same zadania, co skrypt `processftpdlogs`. Dodatkowo:

- wszystkie zarchiwizowane logi tego samego typu za dany rok są łączone w jeden plik, w nazwie którego jest podany rok, następnie pliki te są kompresowane przy pomocy `gzip`,
- konfiguracja Webalizer'a jest dostosowywana do zmiany roku,
- do strony głównej, na której umieszczone są statystyki, czyli do `index.html`, dopisywany jest odnośnik do strony ze statystykami za nowy rok.

Kod skryptu zamieszczono na stronie 57.

Na `math` w tablicy `crontab` *root*'a umieszono następujące wpisy:

```
# FTP log rotation
15 4 1 * *  /opt/sbin/processftpdlogs
59 23 31 12 *  /opt/sbin/processftpdlogs-ny
```

które powodują uruchomienie skryptu `processftpdlogs` pierwszego dnia każdego miesiąca o godzinie 4:15, natomiast `processftpdlogs-ny` raz w roku, minutę przed końcem roku.

Dodatek A

Skrypty

Wszystkie skrypty można pobrać ze strony:

http://theta.uwb.edu.pl/projects/loganalysis/

chkmailquota

```
#!/bin/perl
#
# Last modified: Jun 16, 2003
#
# Copyright (c) 2003 Krzysztof Rapczynski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.


my $MAXSIZE = 5*1024*1024;
my $SYSADM = "root";
my $SELF=`basename $0`;
my ($cmd, $size);

my $MAILER  = "/opt/sbin/sendmail -t";

# opcje:
#  1. posortowana lista od max do min userow (domyslnie)
#  2. n wyslanie informacji
#  3. q quiet - nie wypisywanie niczego

sub sendmail {
        my ($user,$size) = @_;
        my $mMAXSIZE;
        ($mMAXSIZE = $MAXSIZE / (1024 * 1024)) =~ s/(.*\.[0-9]{2}).*/$1/ ;
        my $oversize = $size - $mMAXSIZE;

  open(MAIL, "| $MAILER")
```

```
    or warn "$0: cannot open pipe to $MAILER: $!\n" && return;

  print MAIL "To: $user\n";
  print MAIL "Subject: Wiadomosc od administratora\n\n";
  print MAIL "Twoja skrzynka pocztowa zajmuje $size MB i przekracza o\n";
  print MAIL "$oversize MB dopuszczalny rozmiar.\n\n";
  print MAIL "Prosze usunac zbedne wiadomosci, pozostale przeniesc do\n";
  print MAIL "folderow na swoim koncie.\n\n";
  print MAIL "Nie zastosowanie sie do powyzszych zalecen w ciagu 14 dni\n";
  print MAIL "spowoduje zablokowanie poczty uzytkownika. Postarajmy sie\n";
  print MAIL "nie zapychac naszego serwera.\n\n";
  print MAIL "Diekuje\n\n";
  print MAIL "Administrator\n";
  close(MAIL);
}


sub usage {
        printf("usage: $SELF [-nq] [size]\n");
}

if (@ARGV[0] =~ /^[0-9]+$/) {
        $size = @ARGV[0];
} else {
        $cmd = @ARGV[0];
        $size = @ARGV[1];
}

if (($cmd && $cmd !~ /^-[nq]+$/) || @ARGV[2]) {
        usage;
        exit(1);
}

if ($size) {
        $MAXSIZE = $size;
}

my @list = `ls -l /var/mail`;

sub bysize {
        @fa = split(/[ ]+/,$a);
        @fb = split(/[ ]+/,$b);
        chomp(@fa, @fb);
        $fb[4] <=> $fa[4];
}

my @sortedlist = sort bysize @list;

my $i = 0;

foreach $item (@sortedlist) {
        @fields = split(/[ ]+/,$item);
        chomp(@fields);
        if ($fields[4] > $MAXSIZE) {
            ($size = $fields[4] / (1024 * 1024)) =~ s/(.*\.[0-9]{2}).*/$1/ ;
                $wynik[$i++] = sprintf("%10s %sMB\n", $fields[8], $size);
                if ($cmd =~ "n") {
                        sendmail($fields[8], $size);
                }
        }
}

if ($cmd =~ "n") {
        open(MAIL, "| $MAILER")
        or warn "$0: cannot open pipe to $MAILER: $!\n" && return;
```

```
        print MAIL "To: $SYSADM\n";
        print MAIL "Subject: [chkmailquota] Rozmiary skrzynek INBOX\n\n";
        foreach $item (@wynik) {
                print MAIL $item;
        }
        print MAIL "\n\n";
        print MAIL `df -k`;
        close(MAIL);
}

if ($cmd !~ "q") {
        foreach $item (@wynik) {
                print $item;
        }
}
```

chkhomequota

```
#!/bin/perl
#
# Last modified: Jun 16, 2003
#
# Copyright (c) 2003 Krzysztof Rapczynski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.


my $MAXSIZE = 45*1024*1024;
my $SYSADM = "root";
my $SELF=`basename $0`;
my ($cmd, $size, $username, $wynik);

my $MAILER  = "/opt/sbin/sendmail -t";

# opcje:
#  1. posortowana lista od max do min userow (domyslnie)
#  2. n wyslanie informacji
#  3. q quiet - nie wypisywanie niczego

sub sendmail {
    my ($user,$size) = @_;
    my $mMAXSIZE;
    ($mMAXSIZE = $MAXSIZE / (1024 * 1024)) =~ s/(.*\.[0-9]{2}).*/$1/ ;
    my $oversize = $size - $mMAXSIZE;

  open(MAIL, "| $MAILER")
    or warn "$0: cannot open pipe to $MAILER: $!\n" && return;

  print MAIL "To: $user\n";
  print MAIL "Subject: Wiadomosc od administratora\n\n";
  print MAIL "Twoj katalog domowy zajmuje $size MB i przekracza o\n";
  print MAIL "$oversize MB dopuszczalny rozmiar.\n\n";
  print MAIL "Prosze usunac zbedne pliki lub poddac je kompresji.\n\n";
  print MAIL "Nie zastosowanie sie do powyzszych zalecen w ciagu 14 dni\n";
  print MAIL "spowoduje zablokowanie konta. Postarajmy sie nie zapychac\n";
  print MAIL "naszego serwera.\n\n";
  print MAIL "Diekuje\n\n";
  print MAIL "Administrator\n";
  close(MAIL);
}


sub usage {
    printf("usage: $SELF [-nq] [size]\n");
}

if (@ARGV[0] =~ /^[0-9]+$/) {
    $size = @ARGV[0];
} else {
    $cmd = @ARGV[0];
    $size = @ARGV[1];
}
```

```
if (($cmd && $cmd !~ /^-[nq]+$/) || @ARGV[2] || $#ARGV > 1) {
    usage;
    exit(1);
}

if ($size) {
    $MAXSIZE = $size;
}

my @list = `du -s /export/home/[a-z]*`;

sub bysize {
    @fa = split(/[ ]+/,$a);
    @fb = split(/[ ]+/,$b);
    chomp(@fa, @fb);
        $fb[0] <=> $fa[0];
}

my @sortedlist = sort bysize @list;

my $i = 0;

foreach $item (@sortedlist) {
    @fields = split(/[ \t]+/,$item);
    chomp(@fields);
    $size = $fields[0] * 512;
    if ($size > $MAXSIZE) {
        ($size = $size / (1024 * 1024)) =~ s/(.*\.[0-9]{2}).*/$1/;
        ($username = $fields[1]) =~ s/.*\///;
        $wynik[$i++] = sprintf("%10s %sMB\n", $username, $size);
        if ($cmd =~ "n") {
            sendmail($username, $size);
        }
    }
}

if ($cmd =~ "n") {
    open(MAIL, "| $MAILER")
        or warn "$0: cannot open pipe to $MAILER: $!\n" && return;

    print MAIL "To: $SYSADM\n";
    print MAIL "Subject: [chkhomequota] Rozmiary katalogow domowych\n\n";
    foreach $item (@wynik) {
        print MAIL $item;
    }
    print MAIL "\n\n";
    print MAIL `df -k`;
    close(MAIL);
}

if ($cmd !~ "q") {
    foreach $item (@wynik) {
        print $item;
    }
}
```

processftpdlogs

```
#!/bin/sh
#
# Rotate PROftpd logs. Run at most daily.
#
# Last modified: Jun 18, 2003
#
# Copyright (c) 2003 Krzysztof Rapczynski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

PROFTPDDIR=/opt/var/proftpd
DATE=`date '+%Y-%m-%d'`

# Stop proftpd daemon
/etc/init.d/proftpd stop >/dev/null

# Rotate logs
(cd $PROFTPDDIR
    mv access.log logs/access-$DATE.log
    mv auth.log logs/auth-$DATE.log
    mv xferlog logs/xferlog)

# Restart proftpd daemon
/etc/init.d/proftpd start >/dev/null

# Generate statistics
(cd $PROFTPDDIR
    sed 's%/export/home1/ftpd%%' logs/xferlog > logs/xferlog.tmp
    mv logs/xferlog.tmp logs/xferlog
    /opt/bin/webalizer -Q)

# Rename & gzip log files
(cd $PROFTPDDIR/logs
    mv xferlog xferlog-$DATE
    gzip -q access-$DATE.log
    gzip -q auth-$DATE.log
    gzip -q xferlog-$DATE)

exit 0
```

processftpdlogs-ny

```
#!/bin/sh
#
# Rotate PROftpd logs at the end of the year.
#
# Last modified: Jun 18, 2003
#
# Copyright (c) 2003 Krzysztof Rapczynski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

PROFTPDDIR=/opt/var/proftpd
DATE=`date '+%Y-%m-%d'`
CURYEAR=`date +%Y`
NEWYEAR=`expr $CURYEAR + 1`

# Stop proftpd daemon
/etc/init.d/proftpd stop >/dev/null

# Rotate logs
(cd $PROFTPDDIR
    mv access.log logs/access-$DATE.log
    mv auth.log logs/auth-$DATE.log
    mv xferlog logs/xferlog)

# Restart proftpd daemon
/etc/init.d/proftpd start >/dev/null

# Generate statistics
(cd $PROFTPDDIR
    sed 's%/export/home1/ftpd%%' logs/xferlog > logs/xferlog.tmp
    mv logs/xferlog.tmp logs/xferlog
    /opt/bin/webalizer -Q)

# Rename & gzip log files
(cd $PROFTPDDIR/logs
    mv xferlog xferlog-$DATE

    for f in access-$CURYEAR-*.gz; do
        gunzip $f;
    done
    for f in access-$CURYEAR-*.log; do
        cat $f >> access-$CURYEAR.log
        rm $f
    done
    gzip -qf access-$CURYEAR.log


    for f in auth-$CURYEAR-*.gz; do
        gunzip $f;
    done
    for f in auth-$CURYEAR-*.log; do
        cat $f >> auth-$CURYEAR.log
        rm $f
    done
```

```
    gzip -qf auth-$CURYEAR.log


    for f in xferlog-$CURYEAR-*.gz; do
        gunzip $f;
    done
    for f in xferlog-$CURYEAR-*; do
        cat $f >> xferlog-$CURYEAR
        rm $f
    done
    gzip -qf xferlog-$CURYEAR)


(cd $PROFTPDDIR
    # Create new directory for statistics
    mkdir usage/$NEWYEAR

    # Modify Webalizer configuration
    sed "s/$CURYEAR/$NEWYEAR/g" webalizer.conf > /tmp/webalizer.conf.tmp
    mv /tmp/webalizer.conf.tmp webalizer.conf

    # Modify usage/index.html
    sed '/<\/ul>/i\
  <li><a href=\"NEWYEAR\">NEWYEAR</a>' usage/index.html \
        | sed "s/NEWYEAR/$NEWYEAR/g" > /tmp/index.html.tmp
    mv /tmp/index.html.tmp usage/index.html
    )
done

exit 0
```

Spis literatury

[1] R. Eckstein, D. Collier-Brown, P. Kelly, *Samba*, O'Reilly 2000.

[2] Analysis Console for Intrusion Databases, `http://www.cert.org/kb/acid/`.

[3] Apache Software Foundation, `http://www.apache.org/`.

[4] Armoring Solaris, `http://solaris-x86.org/security/armoring/`.

[5] Armoring Solaris II, `http://solaris-x86.org/security/armoringii/`.

[6] `idled` home page, `http://www.darkwing.com/idled/`.

[7] Poppassd: password change daemon, `http://echelon.pl/pubs/poppassd.html`.

[8] `rsync` home page, `http://rsync.samba.org`.

[9] Snort, `http://www.snort.org/`.

[10] Sudo Man page, `http://www.courtesan.com/sudo/`.

[11] The Horde Project, `http://www.horde.org/`.

[12] The ProFTPD Project, `http://www.proftpd.org/`.

[13] The Webalizer, `http://www.webalizer.org/`.