

Projekt z Bezpieczeństwa Operacji w Internecie
Projekt z Administrowania Siecią TCP/IP

System zabezpieczeń oraz projekt sieci komputerowej TCP/IP dla internetowego portalu typu Online Judge

na przykładzie witryny *Sphere Online Judge*

Adrian Kosowski
Informatyka, semestr IX



<http://spoj.sphere.pl>

Gdańsk, Styczeń 2005

Spis treści

1. Wprowadzenie.....	3
1.1. Charakterystyka zabezpieczanego portalu.....	3
1.2. Funkcjonalność oferowana przez portal	3
1.3. Charakterystyka użytkowników systemu	6
1.4. Wartość informacji w systemie oraz konsekwencje jej utraty.....	7
1.5. Projekt architektury systemu komputerów	9
2. Struktura sieci komputerowej portalu	11
2.1. Połączenie z siecią publiczną.....	11
2.2. Architektura sieci komputerowej	12
2.3. Kryteria doboru urządzeń sieciowych	15
2.4. Kosztorys instalacji sieci komputerowej	16
2.5. Konfiguracja urządzeń sieciowych portalu.....	18
Konfiguracja serwera brzegowego.....	18
Konfiguracja serwera RDBMS	19
Konfiguracja końcówek testowych.....	19
Konfiguracja switcha zarządzalnego.....	20
3. Realizacja systemu zabezpieczeń portalu.....	21
3.1. Analiza potencjalnych zagrożeń i zarys polityki bezpieczeństwa	21
Ogólne założenia ochrony systemu.....	21
Zagrożenia wynikające z czynników losowych lub niedopełnienia obowiązków przez osoby trzecie	22
Zagrożenia wynikające z niedopełnienia obowiązków przez administratorów	24
Zagrożenia wynikające z fizycznego naruszenia bezpieczeństwa systemu	25
Zagrożenia związane z atakami z sieci publicznej.....	27
Zagrożenia wynikające ze specyfiki usług realizowanych przez portal	29
3.2. Identyfikacja głównych słabości systemu.....	31
Krytyczne aspekty bezpieczeństwa danych	31
Problemy z wydajnością i wąskie gardła systemu.....	31
3.3. Bezpieczeństwo danych w portalu	32
3.4. Kontrolowanie uruchamianych zadań.....	32
Zabezpieczenie usług www.....	33
Zabezpieczenie końcówek testowych	33
3.5. Uwagi końcowe	35
Załącznik. Reguły firewalla dla serwera brzegowego	36

1. Wprowadzenie

1.1. Charakterystyka zabezpieczanego portalu

Przedmiotem rozważań jest powszechnie dostępny portal internetowy pozwalający na organizację zawodów programistycznych. System taki, określany popularnie jako *online judge* lub *online tester*, udostępnia użytkownikom do rozwiązania zestaw zadań o charakterze algorytmicznym. Reguły konkursów oraz treści i zasady oceny poszczególnych zadań ustalane są przez uprzywilejowanych użytkowników zarządzających swoimi konkursami poprzez przeglądarkę internetową. Uczestnik konkursu, również korzystając z przeglądarki internetowej, ma możliwość nadesłania rozwiązania wybranego przez siebie zadania. Nadesłany kod jest następnie testowany automatycznie przez system i na tej podstawie poddawany ocenie poprawności.

Większość istniejących portali typu *online judge* realizuje zaledwie najbardziej podstawowy podzbiór omówionej funkcjonalności. Ograniczenie takie jest po części uzasadnione względami bezpieczeństwa, poprzez przyjęcie założenia, że duża część operacji wysokiego ryzyka (np. konfigurowanie zadań i konkursów) wykonywana jest wyłącznie przez osoby zaufane, czyli najczęściej administratorów systemu. Rezygnacja z tego założenia zwiększa znacząco zakres potencjalnych zastosowań systemu (np. o laboratoria z przedmiotów akademickich realizowanych na uczelniach), lecz zarazem wymaga bardziej kompleksowego podejścia do procesu opracowywania zabezpieczeń. Analiza zagrożeń i projekt zabezpieczeń zostaną przeprowadzone na przykładzie portalu internetowego *Sphere Online Judge* (<http://spoj.sphere.pl>), realizowanego na Wydziale ETI PG. W chwili obecnej (po pół roku działania) system liczy przeszło 2000 użytkowników z 70 krajów i dynamicznie się rozwija. Przewidziana w projekcie przebudowa systemu planowana jest na koniec bieżącego roku.

1.2. Funkcjonalność oferowana przez portal

Portal *online judge* można rozpatrywać z jednej strony jako środowisko do automatycznej oceny rozwiązań problemów algorytmicznych i symulacyjnych, z drugiej zaś jako repozytorium dokumentów cyfrowych, dostępnych poprzez przeglądarkę internetową i zarządzanych przy wykorzystaniu prostego systemu Content Management System. Użytkownicy uprzywilejowani mogą zakładać własne konkursy programistyczne, opracowywać do nich zadania oraz określać kryteria oceny poprawności zgłaszanych rozwiązań. Możliwe są modyfikacje wyglądu i zachowania witryny konkursowej zgodnie z

upodobaniem jej twórcy, a w szczególności istnieje możliwość zdefiniowania własnych zasad generowania statystyk i rankingów użytkowników dla danego konkursu. W konwencji właśnie takiego konkursu prowadzone są niektóre laboratoria z przedmiotów na kierunku Informatyka Wydziału ETI (przedmioty kierunkowe Komputerowe Modelowanie Systemów, Projektowanie i Analiza Algorytmów, Logika i Teoria Mnogości, Praktyka Programowania; przedmioty specjalnościowe Podstawy Kryptografii oraz Algorytmy Kombinatoryczne). W przyszłości planuje się wdrożenie systemu na laboratoriach na wydziale MFI Uniwersytetu Gdańskiego i na innych uczelniach.

Witryny konkursowe są dostępne dla ich uczestników i umożliwiają przeglądanie zestawów zadań, forów, stron z informacjami o zawodach, zgłaszanie rozwiązań zadań oraz przeglądanie wyników oceny dotychczasowych rozwiązań i generowanych na bieżąco rankingów użytkowników. Prosty scenariusz pracy z portalem zilustrowano na rys. 1.

Zadania konkursowe mają zazwyczaj charakter algorytmiczny i należą do jednej z kilku kategorii: klasycznej (zadania polegające na realizacji algorytmu rozwiązującego poprawnie postawione zagadnienie), optymalizacyjnej (zadania o charakterze problemów otwartych, zachęcające do poszukiwań suboptymalnych rozwiązań postawionych zagadnień) oraz szkoleniowej (zadania służące wyłącznie do treningu). Punktacja przyznawana za rozwiązanie zadania może być binarna (rozwiązanie są oceniane jako poprawne lub błędne), punktowa (rozwiązania są testowane i otrzymują notę), bądź punktowo-opisowa.

Niezależnie od odbywających się konkursów, przez 24 godziny na dobę dostępna jest dla użytkowników baza przeszło 200 zadań. Część z nich ma charakter oryginalny, a część pochodzi ze wcześniejszych konkursów (np. Olimpiad Informatycznych i Zawodów ACM).

Rozwiązania zadań mogą być zgłaszane w jednym z przeszło trzydziestu języków programowania, takich jak C, C++, Pascal, Java, Smalltalk, Nice, Ada, Ocaml, Prolog, Lisp, Clisp, Perl, Python, PHP, Fortran, Icon, Asembler, Nemerle, C#, Bash.

W przyszłości planuje się rozszerzenie funkcjonalności portalu o interakcję uczestnika konkursu z testowanym programem, pozwalając na testowanie programu na własnych danych testowych i prowadzenie z nim rozgrywek.

Log Out

add news
add upload
add problem
add contest
group list

my account

status
submit
problems

news
contests
info
ranks

forum
help
links

Server time: 2004-12-02 00:19:26

advanced...

Sphere Online Judge

Contests

Running contests

CODE	NAME	START	END
CRYPT04	Podstawy Kryptografii (2004)	2004-10-01 00:00:00	2005-09-01 00:00:00
KMS04	Komputerowe Modelowanie Systemów	2004-10-01 00:00:00	2005-01-26 00:00:00
ALGKOM04	Algorytmy Kombinatoryczne (2004)	2004-10-06 00:00:00	2005-01-26 00:00:00
LIGA04	Programming League 2004	2004-10-09 00:00:00	2005-05-30 00:00:00
PP04_L	Zadania Dodatkowe z PP	2004-10-11 00:00:00	2005-02-01 00:00:00

Past contests

CODE	NAME	START	END
TS1CRP04	Testing by conl	2004-01-01 00:00:00	0000-00-00 00:00:00
CONL_TEST	Contest Testowy	2004-01-01 00:00:00	0000-00-00 00:00:00
SB0	Struktury Baz Danych (default)	2004-10-01 00:00:00	2004-11-05 00:00:00
SPAR001	Eliminacje do AMP	2004-10-09 09:30:00	2004-10-09 14:45:00
SPAR002	Trening przed AMP	2004-10-16 09:30:00	2004-10-16 14:30:00
SP00204	Sparging 2 przed AMP22 2004	2004-10-19 00:00:00	2004-10-19 18:10:00



Log Out

add news
add upload
edit contact

status
submit
problems
current
set 1

links

news
info
rules
forum
links

SPOJ time:
2004-12-02
00:20:58

13 Users:
adrian
cvi
dea
gabrie
harv
hayday
janki
kroovechkaja
luki
maruszp
matt
noix
yry

Programming League 2004

list of set2 problems


ID	S	NAME	CODE	LIBERS	ACC %
72		Food Shortage in Byteland	BYTEFOOD	38	61.07
215	✓	Panic in the Plazas	PANIC	11	12.84
225		Nightmare in the Towers of Hanoi	HANOI	17	54.56
226	✓	Jewelry and Fashion	JEWELS	1	1.20
227	✓	Ordering the Soldiers	ORDERS	49	10.83

Document type: [C](#) [html](#) [C++](#) [pdf](#)



page size: 800x600 1024x768 Full theme: olive banana plum
spoj team at sphere © 2004 all rights reserved





Programming League 2004

[Log Out](#)



[add news](#)
[add upload](#)
[edit contest](#)

[status](#)
[submit](#)
[problems](#)

[ranks](#)

[news](#)
[info](#)
[rules](#)
[forum](#)
[links](#)

12 Users:
 adrian
 ovi
 dea
 gabrie
 haytday
 janki
 kroczewskaja
 luki
 mariuszp
 matt
 noix
 sys

submit a solution

Please insert your source code or choose a file:

```
#include <stdio.h>
#include <string.h>

#define SIZE 8100

char s [SIZE];
int A [SIZE];
int L [SIZE];

int main (void)
{
    int q; scanf ("%d", &q); while (q--)
    {
        // Input
        scanf ("%s", s);
    }
}
```

Language:

C++ (g++ 3.3.3)
C (gcc 3.3.3)
C++ (g++ 3.3.3)
Pascal (gpc v20030830)
Java (jgc 1.0.10)
Java (j2sdk 1.5.0_02)
Nice (nicec 0.8)
JAR (Jre 1.5.0b2)
C# (mcs 1.0.1)
Nemerle (ncc 0.2.1)
Smalltalk (gst 2.1.7)
Assembler (nasm 0.98.38i)

Problem code or id:

JEWELS


☒ private

Send

Full theme: olivane bavana plum

© 2004 all rights reserved

ed in: 0.024s



Log Out

add news
add upload
edit contest

status
submit
problems

ranks

news
info
rules
forums
links

SPOJ time:
2004-12-02
00:25:46

9 Users:
adrian
civian
cristian
deaa
janki
luki
maruszp
matej
noix

Programming League 2004

Contest ranking

Standings from Problemset 3

POS	NAME	CRYPTO4 [0.82]	BURNCTY [1.50]	BYTELE [1.50]	HOLIDAY [0.82]	JEWELS [1.00]	SOLVED	SCORE
1	Pascal Zimmer	0.82	1.50 (1)	0.82	-	-	4	4.64
2	Adrian Kuegel	-	1.50 (1)	-	0.82	1.80	2	2.62
3	Roman Sol	-	0.75 (2)	0.50 (3)	-	-	2	1.25
4	Tomasz Czugra	0.82	-	-	-	-	1	0.82
5	Maxim A. Sokhev	-	-	0.75 (2)	-	-	1	0.75
6	Joachim Erdel	-	0.50 (3)	-	-	-	1	0.50
7	Phạm Mỹ Minh	-	0.38 (4)	-	-	-	1	0.38

Standings from Problemset 2

POS	NAME	BYTEFOOD [1.50]	PANIC [0.54]	HANOI [1.50]	JEWELS [1.00]	ORDERS [0.33]	SOLVED	SCORE
1	Ivan Medvedsky	1.50 (1)	-	0.12 (10)	-	0.33	4	2.49
2	Tomasz Czugra	-	0.54	1.50 (1)	-	0.33	3	2.37
3	zyvez	0.21 (7)	0.54	0.30 (5)	-	0.33	4	1.38
4	Adrian Kuegel	0.19 (8)	-	0.25 (6)	-	0.33	4	1.31
5	Pham Mỹ Minh	0.38 (4)	0.54	-	-	0.33	3	1.24
6	Robin Nikita	0.09 (17)	0.54	0.14 (11)	-	0.33	4	1.09
7	H Nguyen Cong Hiep	0.10 (18)	0.54	-	-	0.33	3	0.97
8	Per Austin	-	0.54	0.38 (4)	-	-	2	0.91
9	Piip Holsti	-	0.54	-	-	0.33	2	0.87
10	Frank Raizer	-	-	0.54	-	0.33	2	0.87
11	Lukasz Koszner	-	0.54 (3)	-	-	0.33	2	0.83
12	Pascal Zimmer	0.30 (5)	-	0.17 (9)	-	0.33	3	0.80
13	Hieu Nguyen	-	0.75 (2)	-	-	-	1	0.75
14	Alexey Danchenko	0.75 (2)	-	-	-	-	1	0.75
15	Shashoua GU	0.54 (3)	0.54	-	-	-	1	0.54
16	Bogdanov Osvuch	-	-	-	-	-	1	0.50

Rys. 1. Przykładowy scenariusz pracy z systemem Sphere Online Judge
(udział w konkursie *DASM Programming League*)

1.3. Charakterystyka użytkowników systemu

Wyróżnić należy następujące grupy użytkowników systemu:

- 1) *Administratorzy portalu* mają bezpośredni, fizyczny dostęp do wszystkich maszyn wchodzących w skład systemu. Do ich obowiązków należy lokalna konfiguracja sprzętu oraz zdalne zarządzanie systemem w warstwie oprogramowania (za pomocą kont shellowych) i nadzorowanie jego prawidłowego działania. Dodatkowo, administratorzy muszą zarządzać treścią portalu internetowego za pośrednictwem dedykowanego panelu administracyjnego dostępnego poprzez przeglądarkę WWW. Liczbę administratorów systemu szacuje się na między 3 a 5 osób.
- 2) *Użytkownicy systemu online judge* korzystają z usług portalu, nawiązując interakcję poprzez przeglądarkę WWW. Większość operacji wykonywanych jest korzystając z protokołu HTTP do przesyłania żądań i pobierania danych w formacie HTML. Przewiduje się możliwość nawiązywania dodatkowych połączeń pomiędzy oprogramowaniem po stronie klienta (aplet w języku JAVA) a wchodzącym w skład portalu systemem testującym oprogramowanie, by umożliwić ręczne testowanie nadsyłanych programów (np. interaktywnych gier turowych). Wykorzystywany jest w tym celu specjalny protokół, a komunikacja odbywa się przez gniazdko TCP/IP. W zależności od uprawnień wyróżniamy następujące rodzaje użytkowników:
 - a. *Użytkowników uprzywilejowanych (organizatorów zawodów)*, którzy mają możliwość współtworzenia portalu poprzez dodawanie nowej treści i zarządzanie nią poprzez specjalny panel administracyjny (dostępny poprzez przeglądarkę WWW). Docelowo przewiduje się, że uprawnienia do dodawania zadań do systemu oraz tworzenia stron internetowych dla swoich zawodów będzie miało ok. 100 użytkowników systemu.
 - b. *Użytkowników ze zwykłym kontem*, którzy przeglądają strony portalu i nadsyłają rozwiązania dostępnych zadań poprzez przeglądarkę WWW. Liczbę takich kont należy docelowo szacować na kilkadziesiąt tysięcy (takie oszacowanie jest możliwe poprzez porównanie dynamiki rozwoju portalu z dynamiką konkurencyjnych portali, w czasie, gdy były na podobnym stadium rozwoju).
 - c. *Użytkowników niezarejestrowanych*, przeglądających statyczną zawartość portalu dostępną dla klientów HTTP. W szczególności należy tu wymienić niezarejestrowanych gości korzystających z przeglądarki WWW (kilka tysięcy

osób, generujących ruch na poziomie kilkudziesięciu tysięcy żądań dziennie) oraz programy pobierające automatycznie zawartość witryny (tzw. spidery, oraz boty popularnych wyszukiwarek internetowych).

Konieczne jest zdefiniowanie odpowiednich uprawnień dostępu, by użytkownicy mieli pełen dostęp do informacji wprowadzonej przez siebie oraz ograniczony dostęp do danych innych użytkowników.

1.4. Wartość informacji w systemie oraz konsekwencje jej utraty

Projektowany system można rozważać jako zbiór logicznie niezależnych witryn internetowych, ale wykorzystujących wspólną bazę danych, serwer www i oprogramowanie systemowe oraz zarządzanych przez uprawnionych użytkowników w jednym wspólnym systemie zarządzania treścią (*content management system*). Informację przechowywaną w systemie można podzielić na dwie kategorie: informację dostępną wyłącznie dla administratorów, oraz informację dodawaną przez użytkowników systemu:

- 1) *Informacja dostępna dla administratorów* obejmuje ustawienia konfiguracyjne i hasła wykorzystywane przez system operacyjny i oprogramowanie serwerowe (serwer baz danych, serwer www), a także kody źródłowe witryny internetowej. Musi ona podlegać ścisłej ochronie (nie może być modyfikowana przez nieuprawnione osoby, a niektóre jej elementy muszą pozostać tajne). Niepożądana zmiana lub wpadnięcie w niepowołane ręce informacji o znaczeniu krytycznym może spowodować całkowity upadek systemu oraz częściowy wyciek informacji dotyczących poszczególnych zawodów. Atakujący odnosi wymierną korzyść w sytuacji, gdy uda mu się dokonać zmian lub przejąć informację w sposób niezauważalny dla administratorów systemu. Może w szczególności odsprzedać zdobyte dane uczestnikom konkursów lub samemu z nich skorzystać (np. po to, by wygrać w konkursie główną nagrodę, lub uzyskać zaliczenie z przedmiotu na uczelni). System projektowany jest z myślą o konkursach o zróżnicowanej puli nagród, o wartości nawet ok. 20000PLN (przeciętna suma oferowana przez sponsora, polską firmę informatyczną, wynosi ok. 5000PLN; w przypadku sponsoringu przez firmy zagraniczne może być wymiernie wyższa). Za zagwarantowanie bezpieczeństwa cennych informacji odpowiadają de facto administratorzy (mimo, iż umowa zawierana z użytkownikami zwalnia ich z odpowiedzialności prawnej i nie pozwala na roszczenia odszkodowawcze).
- 2) *Informacja wprowadzana przez organizatorów zawodów* obejmuje wszystkie dane niezbędne do przeprowadzenia konkursów informatycznych korzystając z portalu

online judge. System musi zapewnić możliwość zaawansowanego zarządzania dostępem do informacji (system *access control lists* dla poszczególnych użytkowników i grup użytkowników korzystających z portalu za pomocą przeglądarki www; bardziej zaawansowany od standardowego 9-polowego systemu uprawnień występującego typowo w systemach uniksowych). Zazwyczaj odpowiedzialność za zamieszczaną przez siebie informację ponoszą użytkownicy (organizatorzy zawodów) i tylko od nich zależy jej wartość. Jakiegokolwiek szacunki liczbowe są dosyć trudne, ponieważ ocena wartości takiej informacji jest bardzo subiektywna. Tak jak w przypadku krytycznej informacji dostępnej dla administratorów, straty materialne związane z przypadkową utratą danych są niewielkie, z pewnością natomiast przyczyniają się do pogorszenia wizerunku organizatorów konkursu oraz samego portalu online judge. Największy problem stanowi niezauważony wyciek informacji konkursowej (treści zadań przed rozpoczęciem zawodów, danych testowych w trakcie zawodów). Dla niektórych osób pozyskanie takiej informacji wydaje się mieć nie tylko pewną wartość materialną (dając wymierną przewagę w rozgrywanym konkursie), ale również dawać dużą satysfakcję, co stanowi dodatkową motywację do podejmowania tego typu działań.

- 3) *Informacja wprowadzana przez uczestników konkursów* stanowi (w rozumieniu ilościowym) największą część danych przechowywanych w systemie. Szczegółnej ochronie podlegać muszą dane osobowe użytkowników. Ze względu na regulacje prawne informacje te traktowane są jako dane krytyczne z punktu widzenia bezpieczeństwa. Część z nich jest jedynie sporadycznie wykorzystywana (dane osobowe, adresy, numery kont bankowych użytkowników), i dlatego powinna być przechowywana wyłącznie na oddzielnym serwerze baz danych i niedostępna do odczytu z serwera www. Baza adresów e-mail użytkowników ma potencjalnie dużą wartość dla spamerów, dlatego należy dołożyć wszelkich starań, by masowe odczytywanie adresów e-mail przez stworzony do tego celu automat było niemożliwe. Należy wreszcie zwrócić uwagę na konieczność ochrony kodów źródłowych nadsyłanych przez użytkowników. Ponieważ kod może mieć trudną do oszacowania wartość intelektualną czy materialną (np. w przypadku rozwiązania przez użytkownika problemu uznawanego za otwarty), należy odpowiednio sformułować umowę przedkładaną użytkownikom, by wykluczyć możliwość roszczeń finansowych wobec administratorów w przypadku przypadkowego poznania kodu źródłowego przez osoby trzecie. Z drugiej strony, nie można dopuścić do sytuacji, by użytkownik

korzystał z oferowanych mechanizmów zamieszczania i pobierania danych binarnych jako konta FTP o publicznym dostępie, gdyż zwiększa to obciążenie serwera i może potencjalnie pośrednio narazić administratorów na konflikt z prawem.

Podsumowując przeprowadzoną charakterystykę, można w skrócie stwierdzić, że informacja niejawną zawartą w systemie ma dużą wartość z punktu widzenia osoby niepowołanej, która jest w stanie zdobyć do niej dostęp w sposób niezauważony. Zniszczenie lub publiczne ujawnienie informacji niejawniej może doprowadzić do kompromitacji serwisu, i w efekcie do jego zamknięcia.

1.5. Projekt architektury systemu komputerów

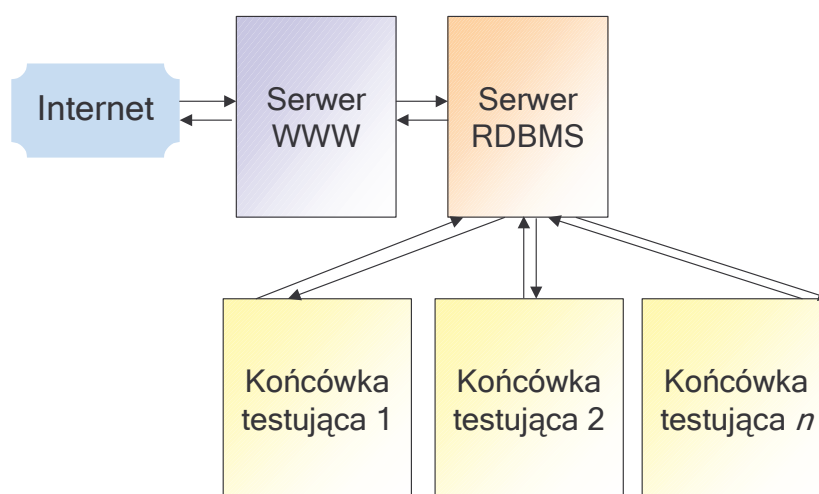
Ze względu na dużą wydajność i moc obliczeniową konieczną do przetwarzania informacji w systemie konieczne jest rozdzielenie realizowanych zadań na kilka komputerów. Ze względu na wykonywane zadania, jednostki należące do systemu można podzielić na trzy zasadnicze kategorie:

- 1) *Komputer odpowiedzialny za komunikację z siecią zewnętrzną* przyjmuje i przetwarza wszystkie żądania otrzymywane od użytkowników korzystających z portalu za pośrednictwem Internetu, pełniąc rolę bramy i serwera podstawowych usług. Do jego zadań należy w szczególności filtrowanie nadchodzących pakietów TCP/IP. Żądania DNS, HTTP/HTTPS oraz związane z obsługą poczty obsługiwane są przez odpowiednio serwer nazw, serwer www oraz serwery POP3, IMAP, SMTP uruchomione na tej samej maszynie. Ruch sieciowy na ściśle określonych portach poddawany jest translacji adresów i kierowany do wybranych komputerów w prywatnej podsieci wewnętrznej (pozwala tu użytkownikowi nawiązać prostą interakcję z programem testowanym na końcówce sprawdzającej). Ruch sieciowy nie związany z usługami realizowanymi przez portal jest wycinany na firewallu.
- 2) *Serwer baz danych* zlokalizowany jest w obrębie podsieci wewnętrznej. Komunikacja z otaczającymi komputerami odbywa się jedynie w bezpiecznych kanałach komunikacyjnych protokołu SSL na kilku dedykowanych portach TCP/IP, w ściśle wyspecyfikowanym protokole w warstwie aplikacji, służącym do przekazywania żądań do bazy danych. Protokół ten stanowi prosty mechanizm wyrażania zapytań do bazy danych oraz uzyskiwania na nie odpowiedzi, uwzględnia jednak specyfikę struktury baz wykorzystywanych w systemie. Żądania przetwarzane są przez centralny moduł zarządzania danymi portalu, który korzysta w sposób przezroczysty dla

żądającego z uruchomionego na komputerze systemu zarządzania relacyjną bazą danych

- 3) *Końcówki sprawdzające* pobierają informacje o zadaniach do realizacji z serwera baz danych za pośrednictwem opisanego wcześniej protokołu. Końcówki podzielone są logicznie na nazwane grupy, a w obrębie pojedynczej grupy uważane są za nierozróżnialne. Końcówki pobierają sekwencyjnie z kolejki programy przydzielone do ich grupy, kompilują je, uruchamiają i testują w bezpiecznym środowisku, i zapisują odpowiedź (wyniki testowania) na serwerze baz danych. Z punktu widzenia konfiguracji sieci maszyny te są niemalże zupełnie odcięte od komunikacji sieciowej, a połączenia inicjują jedynie na określonych portach z serwerem baz danych. Dodatkowo otwarte są porty takich usług jak SSH czy serwera czasu, pozwalające na administrowanie końcówkami przez zdalnie zalogowanych administratorów. Pewne rozszerzenie funkcjonalności stanowi filtrowana komunikacja ze światem zewnętrznym, pozwalająca na testowanie programów na danych testowych nadsyłanych przez użytkownika w czasie rzeczywistym. Liczba końcówek sprawdzających jest uzależniona od aktualnego zapotrzebowania na moc obliczeniową i może ulegać dynamicznym zmianom. W okresach przerw w sprawdzaniu końcówki mogą realizować dodatkowe zadania pomocnicze, np. sprawdzać specjalizowanym programem korelacje pomiędzy nadsyłanymi wcześniej kodami źródłowymi celem wykrycia oszustw i innych nieprawidłowości.

Schemat podstawowej wymiany danych pomiędzy poszczególnymi maszynami podsieci prywatnej portalu został przedstawiony na rys. 2.



Rys. 2. Architektura połączeń serwerów usługowych portalu

2. Struktura sieci komputerowej portalu

2.1. Połączenie z siecią publiczną

Ze względu na zastosowanie portalu w pracy dydaktycznej na wydziale ETI celowe wydaje się umieszczenie jego podsieci prywatnej na terenie Politechniki Gdańskiej. Pozwala to z jednej strony zminimalizować koszty związane z dostarczaniem usług internetowych, z drugiej zaś – zapewnić wystarczającą przepustowość łącz internetowych i zagwarantować niezawodność usługi dla osób korzystających z Internetu z sieci lokalnej Politechniki (np. przez studentów znajdujących się w laboratoriach). Bieżącą lokalizację portalu stanowi laboratorium dyplomowe Katedry Architektury Systemów Komputerowych. Podsieć portalu ma zapewnione miejsce do przechowywania ok. 10 komputerów i innych urządzeń sieciowych, połączenie z publiczną siecią pakietową za pośrednictwem 100Mbit sieci lokalnej Ethernet wydziału ETI (poprzez bramy wydziałowy i routery Ośrodka Informatycznego/TASK), oraz 3 publicznie widoczne adresy IP.

Pewną wadą istniejącego połączenia jest brak bezpiecznej lokalizacji serwerów (w pomieszczeniu o całkowicie kontrolowanym dostępie, z zapewnionym zasilaniem awaryjnym) oraz duża zawodność połączenia z siecią zewnętrzną (poza Politechnikę) ze względu na długą drogę z sieci wydziałowej ETI do głównych przełączników sieci ATM. Za 2 lata planuje się przeniesienie portalu do nowego gmachu wydziału ETI i podjęcie starań w celu podłączenia go do sieci 100Mbit powiązanej bezpośrednio z infrastrukturą ATM 622Mbit głównego centrum TASK. Pozwoliłoby to na poprawę niezawodności połączenia i zwiększenie jego niezawodności. Być może zaistnieje też możliwość realizacji połączenia poprzez rozwijaną obecnie przez CI TASK infrastrukturę Ethernetu gigabitowego w sieci o topologii kratownicy, służącego do realizacji połączeń pomiędzy gdańskimi uczelniami.

Aby zapewnić możliwie dużą elastyczność, niniejszy projekt uwzględniać będzie jako podstawową opcję połączenia portalu z Internetem za pośrednictwem sieci Ethernet o przepustowości do 1Gbit, przy czym wykorzystywane urządzenia będą poprawnie i stosunkowo wydajnie funkcjonowały również w sieci 100Mbit. Ze względu na specyfikę lokalizacji portalu nie jest niestety możliwe zapewnienie alternatywnego połączenia z Internetem, wykorzystującego łącza innego providera.

2.2. Architektura sieci komputerowej

Odcięcie wszystkich komputerów portalu za wyjątkiem gateway'a/bastion hosta od sieci publicznej jest konieczne z kilku powodów. Ukrycie komputerów wykonujących operacje obliczeń i przetwarzania danych w obrębie sieci prywatnej pozwala zmniejszyć ryzyko ich ataku oraz zredukować obciążenie procesora (w przestrzeni jądra systemu) związane z przetwarzaniem ruchu sieciowego. Aspekt bezpieczeństwa jest ważniejszy z punktu widzenia serwera relacyjnej bazy danych, natomiast kwestia wydajności odgrywa szczególną rolę w przypadku końcówek testowych, które dokonują m.in. pomiarów czasu działania programów użytkowników. Ponadto, rozwiązanie takie pozwala ograniczyć pulę adresów IP wykorzystywanych przez system i uniemożliwia próby podsłuchania pakietów wymienianych pomiędzy komputerami portalu z sieci wydziału ETI.

W podstawowej konfiguracji koniecznej do pracy portalu, sieć złożona jest z następujących elementów:

1) Komputerów klasy PC:

- a. Serwera brzegowego (pełniącego rolę firewalla/bastion hosta oraz serwera usług www i pocztowych), wyposażonego w kartę sieciową służącą do komunikacji z siecią zewnętrzną (Ethernet 100Mbit lub 1Gbit, interfejs uzależniony od rodzaju połączenia z siecią, paragraf 2.4) oraz kartę sieciową służącą do komunikacji z podsiecią prywatną (Ethernet 1Gbit 1000Base-TX),
- b. Serwera usługi portalowej i RDBMS, połączonego z podsiecią prywatną poprzez kartę sieciową Ethernet 1Gbit 1000Base-TX,
- c. Końcówek testowych (nierozróżnialnych w obrębie ustalonych podgrup), komunikujących się z siecią prywatną poprzez kartę sieciową Ethernet 1Gbit 1000Base-TX.

2) Osprzętu sieciowego:

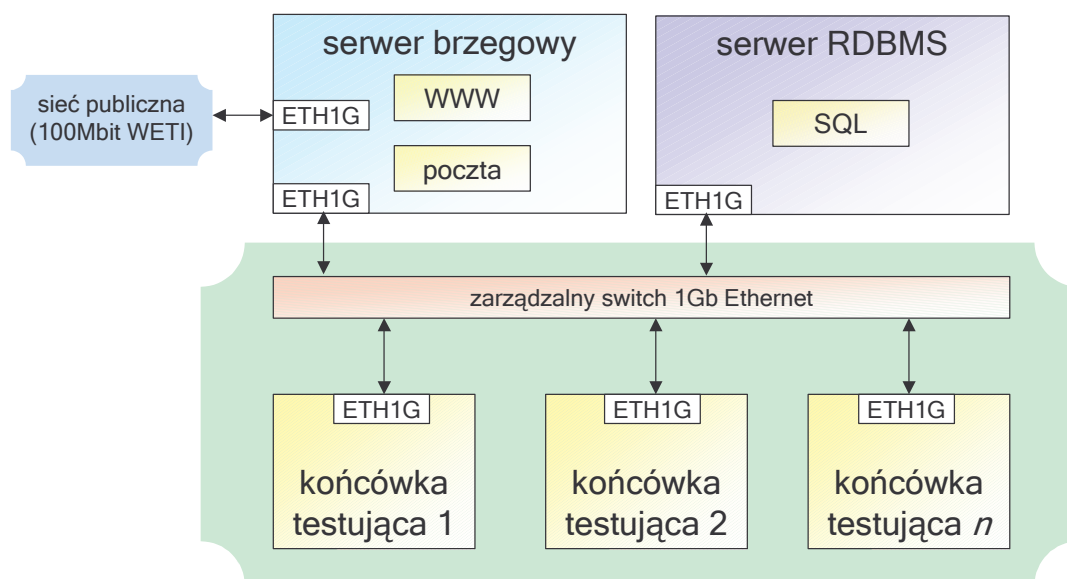
- a. Zarządzalnego switcha Ethernet 1Gbit 1000Base-TX (zarządzanego przez serwer brzegowy), stanowiącego trzon podsieci prywatnej, do którego portów podłączone są wszystkie komputery tworząc sieć o topologii gwiazdy,
- b. Osprzętu dostarczanego przez providera Internetu, a znajdującego się poza systemem, służącego do kształtowania ruchu sieciowego i zapobiegania w możliwie dużym stopniu zalewaniem pakietami i atakom DoS (routery

sprzętowe i software'owe bramy zarządzane przez administratorów Politechniki).

3) Okablowania

- a. Okablowania 1000Base-TX ze złączami RJ-45 (przewody długości nie przekraczającej 2m) służącego do realizacji połączeń pomiędzy switchem a komputerami sieci prywatnej,
- b. Okablowania służącego do połączenia serwera brzegowego z siecią publiczną (okablowanie 1000Base-TX ze złączami RJ-45, bądź światłowodowym łączem złożonym z pary przewodów Duplex-SC, w zależności od przyjętego wariantu połączenia z siecią publiczną),
- c. Kabli szeregowych łączących serwer z portem szeregowym sterującym switcha (wymagane w przypadku, gdy switch nie oferuje możliwości sterowania poprzez port).

Schemat poglądowy takiej konfiguracji sieci komputerowej przedstawiono na rys. 3.



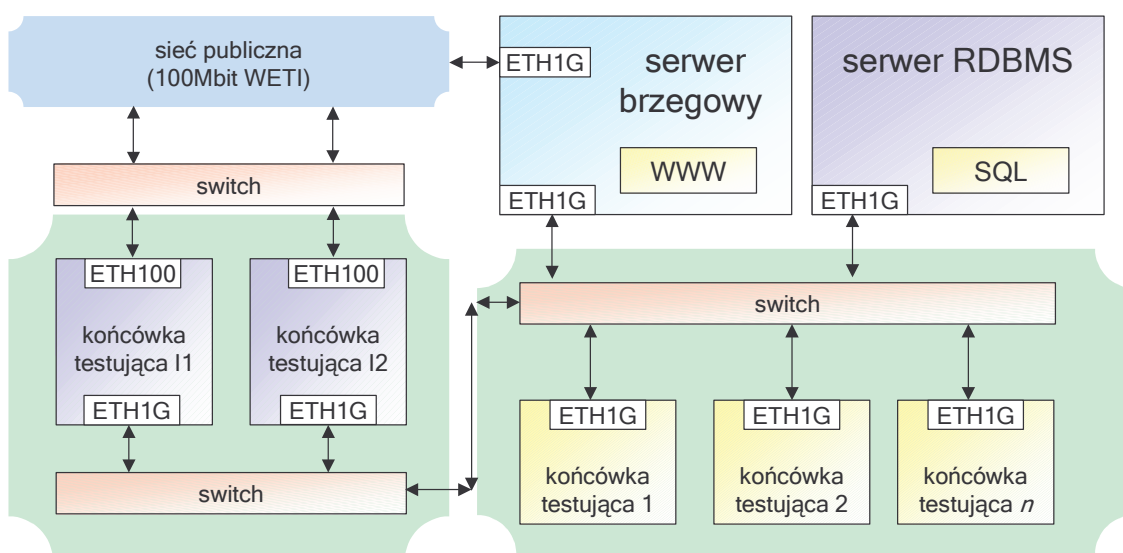
Rys 3. Schemat poglądowy podstawowej konfiguracji sieci komputerowej portalu

Tak zdefiniowana architektura sieci komputerowej pozwala na opracowanie konfiguracji oprogramowania i sprzętu (punkt 2.3) zapewniającej wysoki poziom bezpieczeństwa oraz szybkość działania zadawalającą z punktu widzenia użytkownika przeglądarki internetowej. Funkcjonalności portalu obejmuje możliwość interakcji z użytkownikiem w trakcie procesu

testowania, poprzez filtrowany routing pakietów z sieci publicznej, przez serwer, do końcówek testowych. Należy jednak zwrócić uwagę, że w przypadku testowania gier i programów interaktywnych, zezwalając na przesyłanie informacji multimedialne w czasie rzeczywistym, serwer brzegowy zmuszony jest do routowania ruchu o dużym natężeniu, a wymagającego krótkich czasów odpowiedzi. W tej sytuacji wskazane wydaje się wydzielenie dedykowanych końcówek testowych nawiązujących interakcję z użytkownikiem poprzez bezpośrednie wyjście na świat. Do osprzętu sieciowego należy wówczas dołożyć następujące elementy:

- 1) Dodatkowe interfejsy sieciowe dla interaktywnych końcówek testowania czasu rzeczywistego, wyposażone w identyczne karty sieciowe jak serwer brzegowy.
- 2) Nowe elementy osprzętu mające na celu zwiększenie elastyczności i bezpieczeństwa konfiguracji sieciowej:
 - a. Switch zarządzalny (kontrolowany przez serwer brzegowy), do którego podłączone są interaktywne końcówki testowe oraz łącze sieci zewnętrznej,
 - b. Opcjonalnie, oddzielny switch zarządzalny dla końcówek testowych po stronie podsieci wewnętrznej, połączony z pierwotnym switchem metodą stackowania.

Wprowadzenie tego elementu ma na celu ułatwienie podjęcia akcji odcięcia końcówek interaktywnych w przypadku ataku z zewnątrz (na który są one szczególnie narażone).



Rys 3. Schemat poglądowy rozszerzonej konfiguracji sieci komputerowej portalu

2.3. Kryteria doboru urządzeń sieciowych

Przy wyborze rozwiązań konkretnych producentów należy rozpatrywać szereg kryteriów:

- 1) *Względy niezawodnościowe.* Zakupiony sprzęt powinien być w stanie działać przez kilka lat całkowicie bezawaryjnie. Należy zwrócić uwagę, że warunki pracy systemu są niezwykle stabilne (wentylowane pomieszczenie zamknięte o ustalonej temperaturze) i jedynie w niewielkim stopniu narażone na zdarzenia losowe (burze, przepięcia). Niezawodność jest szczególnie istotna w przypadku końcówek testowych, w których każde, nawet najmniejsze, usterki sprzętowe są zauważane stosunkowo szybko, ale ze względu na niewielką powtarzalność okazują się często trudne do diagnozowania.
- 2) *Względy wydajnościowe.* Wymaga się, aby osprzęt sieciowy pracował wydajnie w warunkach zmiennego obciążenia (nagłe zmiany natężenia ruchu sieciowego). W szczególności, transmisja danych pomiędzy komputerami musi w jedynie niewielkim stopniu obciążać procesory obu maszyn, czego konsekwencją jest wymaganie krótkich czasów transmisji danych w obrębie podsieci prywatnej. Osprzęt sieciowy nie musi natomiast być w stanie realizować wolnozmiennego ruchu ciągłego z prędkością bliską maksymalnej (z analiz obciążenia sieci wynika, że taki ruch w zasadzie w portalu nie występuje). Przy zakupie komputerów należy wymagać szybkiej konfiguracji obliczeniowej na serwerze brzegowym oraz wysokowydajnych i niezawodnych dysków twardych zarówno na serwerze brzegowym, jak i na serwerze RDBMS. Wysoka wydajność końcówek sprzętowych nie jest wymagana, ale nierozróżnialne końcówki (należące do jednej grupy) nie mogą się różnić parametrami wydajnościowymi.
- 3) *Względy bezpieczeństwa.* Switche sterowalne muszą pozwolić na bezwzględne blokowanie połączeń pomiędzy ustalonymi parami portów, udostępniając mechanizmy sterowania warstwy trzeciej ISO/OSI (umożliwiając routing wg IP adresata). Karty sieciowe od strony sieci zewnętrznej muszą być odporne na znane ataki przepełnieniowe w warstwie fizycznej. Osprzęt sieciowy musi umożliwiać realizację polityki bezpieczeństwa zdefiniowanej w rozdziale 3.
- 4) *Względy cenowe.* Koszt kupowanych urządzeń sieciowych powinien być stosunkowo niski w porównaniu do konkurencyjnych urządzeń tej samej klasy. W przypadku dokonywania zakupu w drodze przetargu (z środków finansowych politechniki, np. z

rezerwy dziekańskiej) konieczne jest jasne sprecyzowanie nazw i modeli wymaganych urządzeń.

2.4. Kosztorys instalacji sieci komputerowej

Mając na uwadze zdefiniowane w poprzednim punkcie kryteria, i zakładając ograniczenie nakładów inwestycyjnych do 10000PLN, można zaproponować przykładowe zestawienie komponentów sieciowych:

- 1) Serwer brzegowy i serwer RDBMS: dwa komputery klasy PC (koszt jednostkowy ok. 3500PLN) wyposażone w procesor szybkości Pentium 4 2GHz lub porównywalnej, 1GB pamięci operacyjnej RAM oraz macierze dyskowe pracujące w trybie RAID 0+1, 5 lub 10 o pojemności min. 100GB. Końcówki testowe klasy Pentium 3 500MHz, 128MB pamięci operacyjnej RAM, pozyskane za darmo z laboratoriów wydziału ETI.
- 2) Jako główny przełącznik sieci może posłużyć jeden ze switchy Hewlett-Packard z serii hp procure switch, zarządzanych w warstwie 3 (statyczne zarządzanie adresami IP). Ze względu na wciąż wysoką cenę przełączników gigabitowych 1000-Base TX, można rozważyć zakup switcha wyposażonego w 24 porty 10/100Mbit (podłączane do końcówek testowych) oraz 2 porty 1Gbit (podłączone do serwera brzegowego i serwera RDBMS). Skonstruowany w takiej konfiguracji przełącznik hp procure switch 2626, rys. 4a, kosztuje w chwili obecnej ok. 2500PLN. Działający w oparciu o procesor Motorola MPC8245 PowerPC, pozwala na realizację przełączania z opóźnieniem nie większym niż 12 mikrosekund, przy przepustowości szyny głównej 9.6Gbit/s. Sterowanie switchem odbywa się poprzez port szeregowy RS 232C. W przypadku dokonywania zakupów przyszłości (za ok. 2 lata) poważną alternatywę może stanowić w pełni gigabitowy switch zarządzalny o mniejszej liczbie portów. Przykładowo, switch zarządzalny hp procure switch 6108, rys. 4b, wyposażony jest w 8 portów komunikacji gigabitowej, charakteryzujące się podobnymi parametrami jak model 2626. Jego cena wynosi obecnie 6000PLN i obniża się o ok. 30% rocznie.
- 3) W celu minimalizacji prawdopodobieństwa wystąpienia jakichkolwiek konfliktów, zarówno serwer RDBS i serwer brzegowy od strony podsieci wewnętrznej powinny zostać wyposażone w niezawodne karty sieciowe ze złączem RJ 45. Stawianym wymaganiom dobrze odpowiadają gigabitowe karty sieciowe 3COM (np. 10/100/1000Base-T NIC 3C2000-T), dostępne w cenie ok. 200PLN. Ze względu na

a)



b)



Rys. 4. Ethernetowe przełączniki firmy Hewlett Packard zarządzalne w warstwie 3.

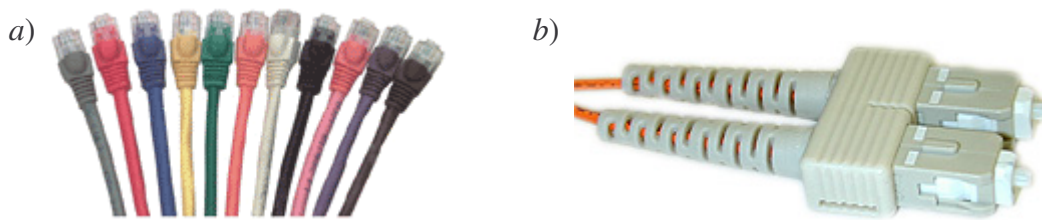
a) 24-portowy switch hp procure switch 2626

b) w pełni gigabitowy 8-portowy switch hp procure switch 6108

jakość wbudowanych układów mikroprocesorowych, karty te cechuje zdecydowanie mniejsze obciążenie procesora niż w przypadku popularnych kart opartych na chipsecie firmy Realtek. W przypadku końcówek testowych dobór karty sieciowej ma nieco mniejsze znaczenie, i prawdopodobnie do działania wystarczą im zainstalowane w momencie wycofania z laboratoriów WETI karty 100Mbit 3COM.

Największe znaczenie z punktu widzenia wydajności ma dobór karty sieciowej używanej w połączeniu serwera brzegowego z siecią zewnętrzną. W przypadku połączenia RJ 45 można wykorzystać te same modele kart 3COM; w przypadku połączenia światłowodowego koszt karty jest odpowiednio wyższy (ponad 900PLN).

- 4) Do okablowania połączeń sieciowych pomiędzy złączami RJ 45 można wykorzystać kable nie krzyżowane skręcone nieekranowane (UTP) kategorii 5e lub 6. Stosowanie kabli kategorii 6, rys. 5a, w połączeniach gigabitowych wydaje się uzasadnione ze względu na niewielki koszt pojedynczego połączenia (ok. 10PLN za 2m kabel) i dobre parametry przy transmisji o wysokiej częstotliwości. W przypadku zaistnienia konieczności podłączenia serwera brzegowego za pomocą światłowodu (łącze parowane SC, rys. 5b), koszt pojedynczego kabla wynosi ok. 150PLN.



Rys. 5. Okablowanie połączeń projektowanej sieci Ethernet

a) końcówki kabli Ethernet UTP kategorii 6. (350MHz pasmo przenoszenia)

b) końcówka pary ethernetowych przewodów światłowodowych, złącze typu SC

Zaprojektowana konfiguracja pozwala na realizację planowanej sieci komputerowej bez znaczącego przekroczenia przewidzianego budżetu 10000PLN. Ponieważ przez pewien okres można korzystać z tańszej infrastruktury znajdującej się na terenie wydziału (np. switchy 100Mbit), koszt inwestycyjny można rozłożyć na szereg przetargów, w okresie nawet ok. 2 lat. Możliwy jest też w terminie późniejszym zakup dodatkowych switchy pozwalających na bezpośrednie zestawianie połączeń z końcówek interaktywnych do sieci publicznej (początkowo można do tego celu wykorzystywać nieużywane porty zakupionego zarządzalnego switcha, jednak może się to wiązać z niewielkim spadkiem wydajności).

2.5. Konfiguracja urządzeń sieciowych portalu

Konfiguracja serwera brzegowego

Serwer brzegowy portalu musi łączyć role wydajnego serwera usług www z funkcją bastion hosta strzegącego podsieci prywatnej. Optymalnym wyborem wydaje się więc jeden z najbezpieczniejszych darmowych systemów wzorowanych na Uniksie, Debian GNU/Linux pracujący na jądrze z serii 2.4, z dodatkowymi łataniami podnoszącymi bezpieczeństwo systemu. Komputer wyposażony jest w dwa interfejsy ethernetowe odpowiadające jego kartom sieciowym: eth/0 do komunikacji z siecią zewnętrzną oraz eth/1 połączony z podsiecią prywatną. Dodatkowo, port szeregowy RS 232C wykorzystywany jest do zarządzania switchem sterowalnym.

Usługi udostępniane przez serwer do sieci zewnętrznej obejmują w szczególności DNS (usługę serwera nazw), WWW (protokół HTTP oraz szyfrowany protokół HTTPS po SSL) oraz SMTP (w celu wysyłania powiadomień do użytkowników portalu). Do celów administracyjnych udostępnione są również usługi SSH i SCP/SFTP, służące do zdalnej konfiguracji serwera i przesyłania kopii zapasowych danych na serwery zewnętrzne. W fazie

rozwoju kodu portalu przydatna jest też zdalnie udostępniona usługa CVS (pserver), pozwalająca na zarządzanie kodem źródłowym. Pewne dodatkowe usługi mogą być udostępniane jako udogodnienie dla administratorów i innych użytkowników kont shellowych (np. POP3/IMAP do odbierania poczty). Opcjonalnie (do celów diagnostycznych) serwer może odpowiadać na pingowanie pakietami ICMP.

Po stronie sieci wewnętrznej komunikacja pomiędzy serwerem brzegowym a serwerem RDBMS odbywa się wyłącznie po wydzielonym tunelu SSL, korzystając ze zdefiniowanego dla portalu w warstwie prezentacji protokołu do przekazywania zapytań i odpowiedzi na zapytania. Nieużywane porty w obrębie zarówno sieci wewnętrznej, jak i zewnętrznej są zamknięte.

Wydaje się celowe wykorzystanie serwera brzegowego jako serwera cache dla pakietów dystrybucji Debian, co pozwala na aktualizację oprogramowania wszystkich komputerów podsieci (np. poprzez polecenie apt-get) bez konieczności udostępniania im połączenia z Internetem.

Reguły firewalla dla serwera brzegowego zostały bardziej szczegółowo opisane w załączniku. W przypadku realizacji interakcji pomiędzy użytkownikiem zdalnym a końcówkami testowym bez wydzielonego połączenia z Internetem (za pośrednictwem serwera brzegowego), bądź też w przypadku tunelowania połączeń pomiędzy końcówkami testowymi a serwerem RDBMS poprzez serwer brzegowy, konieczne jest zdefiniowanie dodatkowych reguł routingu dla tego serwera.

Konfiguracja serwera RDBMS

Serwer RDBMS może działać na podobnie zdefiniowanej platformie jak serwer brzegowy. Reguły firewalla dla serwera RDBMS powinny ograniczać ruch sieciowy do pojedynczego portu, po którym odbywa się cała komunikacja w protokole zdefiniowanym dla portalu (na gniazdkach SSL). Podstawową usługą uruchomioną na serwerze RDBMS jest główny serwer danych portalu (oprogramowanie napisane np. w języku JAVA), wykorzystujący bazę danych SQL na lokalnie uruchomionym serwerze.

Konfiguracja końcówek testowych

Końcówki testowe pracują w oparciu o ten sam system operacyjny Debian GNU/Linux co pozostałe komputery, ale w jeszcze bardziej restrykcyjnej konfiguracji. Wszystkie usługi normalnie dostępne dla użytkowników shellowych (np. związane z komunikacją w sieci,

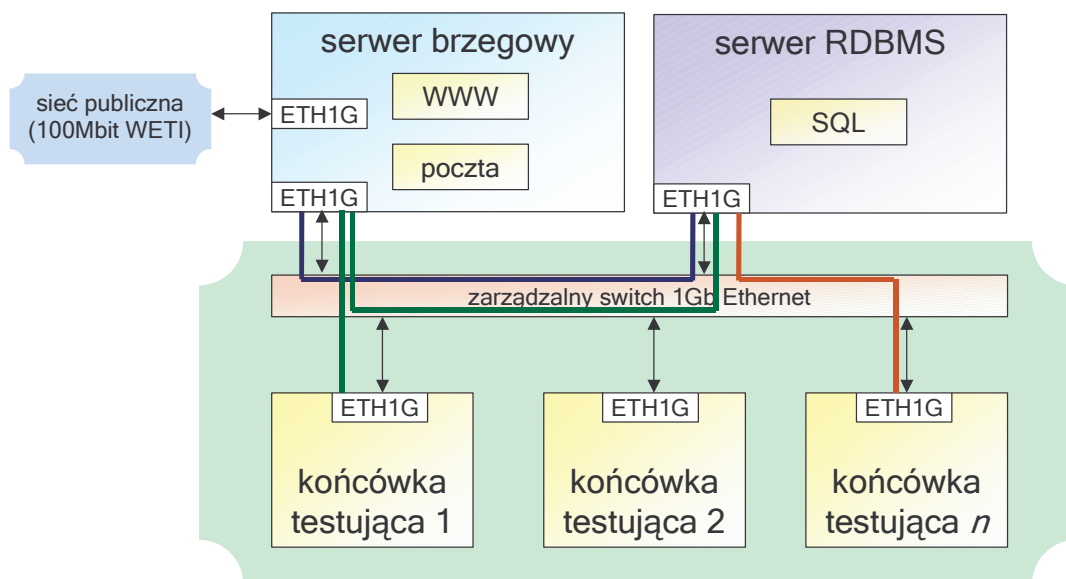
zakładaniem socketów, etc.) są wyłączone, a reguły firewalla zezwalają jedynie na komunikację na wydzielonym porcie z serwerem RDBMS (komunikacja realizowana jest wyłącznie przez oprogramowanie działające w trybie użytkownika uprzywilejowanego root). W przypadku dopuszczenia interakcji pomiędzy końcówkami testowymi a użytkownikiem internetowym (korzystającym np. z apletu Javy w przeglądarce internetowej), konieczne jest zdefiniowanie dodatkowego protokołu komunikacyjnego oraz bezpiecznego kanału komunikacji. Jeżeli komunikacja ta odbywa się poprzez podsieć prywatną i serwer brzegowy, konieczne jest uwzględnienie tego elementu w konfiguracji routingu i firewalla dla serwera brzegowego; jeżeli komunikacja odbywa się bezpośrednio, dla końcówek testowych należy zdefiniować odrębny interfejs sieciowy, powiązany z kartą sieciową mającą bezpośrednie wyjście na świat.

Konfiguracja switcha zarządzalnego

Zarządzanie switchem w warstwie 3 odbywa się poprzez port szeregowy, z którym łączy się serwer brzegowy (traktując go jak zdalny terminal). Głównym celem zarządzania ruchem jest zapewnienie odpowiedniego poziomu bezpieczeństwa danych (rozdział 3) i uniemożliwienie końcówce testowej ataku na inne komputery podsieci prywatnej i komunikacji z komputerami sieci publicznej, w przypadku przejęcia końcówki przez atakujący ją program.

Zarządzanie switchem na poziomie protokołu IP powinno obejmować dwa zasadnicze elementy. Po pierwsze, należy dla każdego portu switcha określić jedyny dopuszczalny adres IP, który może być do niego przypisany. Ponadto konieczne jest podanie dokładnych par adresów IP, które mogą się ze sobą komunikować. W szczególności, należy zezwolić na wymianę pakietów pomiędzy serwerem brzegowym oraz serwerem RDBMS, oraz pomiędzy końcówkami testowymi a serwerem RDBMS (bądź serwerem brzegowym, przy założeniu polityki podwyższonego bezpieczeństwa). Należy całkowicie wykluczyć możliwość komunikacji pomiędzy parami końcówek testowych. Serwer brzegowy musi być wyposażony w opracowane przez administratora narzędzia pozwalające na rekonfigurację switcha i natychmiastowe odcięcie końcówki testowej w przypadku wykrycia utraty kontroli nad jej systemem.

Dozwolony ruch sieciowy przechodzący przez switch został zilustrowany na rys. 6.



Rys. 6. Konfiguracja głównego switcha systemu (linia niebieska – trasa pomiędzy serwerem brzegowym a BMS, linia czerwona – trasa pomiędzy końcówką testową a serwerem RDBMS w konfiguracji normalnej, linia zielona – trasa pomiędzy końcówką testową a serwerem RDBMS w konfiguracji o podwyższonym bezpieczeństwie)

3. Realizacja systemu zabezpieczeń portalu

3.1. Analiza potencjalnych zagrożeń i zarys polityki bezpieczeństwa

Ogólne założenia ochrony systemu

Przeprowadzona w poprzednim punkcie analiza zagrożeń pozwala na zdefiniowanie jasnej, jednoznacznej polityki bezpieczeństwa dla systemu. Poniżej podano jej najważniejsze założenia:

- 1) mianem katastrofy systemu można określić sytuację, w której osoba nieuprawniona uzyska dostęp do informacji w systemie, a następnie będzie w stanie ją w sposób niekontrolowany przesłać do sieci globalnej i odczytać
- 2) serwer musi być zabezpieczony przed następującymi zdarzeniami:
 - a. utratą danych z przyczyn losowych (zniszczenie danych na skutek awarii sprzętu lub błędu oprogramowania)
 - b. atakiem z sieci zewnętrznej (włamanie z innego hosta lub hostów internetowych mające na celu przejęcie lub naruszenie danych, bądź też sparaliżowanie systemu)

- c. atakiem z sieci wewnętrznej (włamanie przeprowadzone poprzez proces uruchomiony na maszynie w podsieci lokalnej portalu, który realizuje kod nadesłany przez użytkownika internetowego korzystającego z portalu).
- 3) naruszenie bezpieczeństwa pojedynczej końcówki testowej (służącej do uruchamiania procesów użytkowników) musi zostać natychmiastowo wykryte i musi istnieć możliwość logicznego odcięcia końcówki od pozostałych komputerów sieci wewnętrznej oraz od sieci zewnętrznej
- 4) przerwy systemu spowodowane awarią sprzętu, odcięciem od sieci internetowej lub energetycznej, przeciążeniem serwera bądź atakiem powinny być możliwie najkrótsze, a w pewnych określonych terminach (w trakcie zawodów krótkoterminowych) są w zasadzie niedopuszczalne, a jeżeli wystąpią, nie mogą przekraczać 2-3 minut.

Poniżej wyszczególniono najważniejsze zagrożenia, na które narażony być może system.

Zagrożenia wynikające z czynników losowych lub niedopełnienia obowiązków przez osoby trzecie

Zagrożenie: Poważna awaria sprzętowa komputera wchodzącego w skład systemu.

Osoba odpowiedzialna za spowodowanie zagrożenia: Brak albo trudno określić.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu (może być wymagana bezpośrednia, fizyczna interwencja) .

Przyczyny: Usterka techniczna, wynikła prawdopodobnie ze „zmęczenia materiału”, (komputery wchodzące w skład systemu pełnią rolę mocno obciążonych jednostek obliczeniowych), często w momencie zaburzenia normalnej pracy (np. przepięcia sieci energetycznej).

Skutki: Przerwa w działaniu systemu, jeżeli jednostką uszkodzoną jest serwer. Zmniejszenie przepustowości obliczeniowej systemu, jeżeli uszkodzona jest jedna lub część końcówek sprawdzających. Utrata danych, jeżeli jednostką uszkodzoną jest serwer baz danych.

Ocena zagrożenia: Poważne.

Likwidacja skutków: Przywrócenie systemu do działania w możliwie krótkim czasie. W przypadku uszkodzenia serwera, zastąpienie go przez komputer zapasowy. W przypadku uszkodzenia serwera baz danych, odtworzenie bazy danych (w razie konieczności z zewnętrznej kopii zapasowej) oraz przejęcie roli serwera baz danych przez komputer zapasowy lub serwer www. Brak konieczności natychmiastowej interwencji w przypadku

uszkodzenia końcówki sprawdzającej. W terminie późniejszym naprawa sprzętu i przywrócenie oryginalnej konfiguracji.

Środki zaradcze: Regularne sprawdzanie stanu sprzętu narzędziami diagnostycznymi oraz tworzenie kopii zapasowych na nośnikach zewnętrznych.

Zagrożenie: Awaria innego elementu infrastruktury podsieci wewnętrznej.

Osoba odpowiedzialna za spowodowanie zagrożenia: Brak albo trudno określić.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Niepoprawne działanie switcha lub uszkodzenie elementu okablowania.

Skutki: Przerwa w działaniu systemu, jeżeli awaria powoduje odcięcie serwera www i baz danych od sieci publicznej. Zmniejszenie przepustowości obliczeniowej systemu, jeżeli niemożliwa jest komunikacja z końcówkami sprawdzającymi (w sytuacji, gdy żadna nie jest dostępna ich rolę może przejąć jeden z serwerów)

Ocena zagrożenia: Umiarkowane.

Likwidacja skutków: Przywrócenie systemu do działania w możliwie krótkim czasie. Kontrola poprawności działania interfejsów sieciowych poszczególnych maszyn.

Środki zaradcze: Trudne do podjęcia. Możliwy ciągły monitoring infrastruktury sieciowej.

Zagrożenie: Przerwa w działaniu systemu z przyczyn zewnętrznych.

Osoba odpowiedzialna za spowodowanie zagrożenia: Zewnętrzny dostawca usług.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Osoba uprawniona do podjęcia stosownej interwencji u dostawcy usług celem przyspieszenia przywrócenia usług.

Przyczyny: Różne, np. przerwa w zasilaniu, lub odcięcie Wydziału ETI na routerach PG (komunikacja z siecią zewnętrzną jest niemożliwa przez ok. 1% czasu).

Skutki: Przerwa w działaniu systemu trudnej do określenia długości (do kilku godzin).

Ocena zagrożenia: Umiarkowane.

Likwidacja skutków: Poinformowanie dostawcy usług o problemie. Naprawa podejmowana przez osoby trzecie.

Środki zaradcze: Regularna wymiana informacji (np. o planowanych remontach, przeglądach) z dostawcami usług i władzami Wydziału.

Zagrożenia wynikające z niedopełnienia obowiązków przez administratorów

Zagrożenie: Upadek systemu na skutek krótkotrwałego zaniedbania nadzoru.

Osoba odpowiedzialna za spowodowanie zagrożenia: Dyżurny administrator systemu.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Brak osób nadzorujących system (np. w okresie wakacyjnym)

Skutki: Brak możliwości podjęcia skutecznych działań w wypadku zaistnienia innego zagrożenia. Jeżeli okres nieobecności administratora nie przekracza doby, szansa zaistnienia poważnych problemów jest niewielka. System posiada bowiem odpowiednio dużą „rezerwę mocy” obliczeniowej w jednostkach najbardziej podatnych na uszkodzenie, przez co zazwyczaj uszkodzenia systemu wywołają jedynie jego częściowe spowolnienie (*graceful degradation of performance*).

Ocena zagrożenia: Umiarkowane.

Likwidacja skutków: Uzależniona od problemów, które wystąpiły pod nieobecność administratora.

Środki zaradcze: Zwiększenie liczby lub zacieśnienie współpracy między administratorami.

Zagrożenie: Upadek systemu na skutek długotrwałego zaniedbania nadzoru.

Osoba odpowiedzialna za spowodowanie zagrożenia: Dyżurny administrator systemu.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Brak nadzoru systemu przez kilka tygodni. Z obserwacji wynika, że rozwijany system pozostawiony bez opieki przez okres ok. 1 miesiąca przestanie działać, zazwyczaj z przyczyn losowych zewnętrznych (logiczne odcięcie serwera na routerach PG, przypadkowe odłączenie systemu od zasilania przez osobę niepowołaną, etc.). Ze względu na lokalizację systemu prawdopodobieństwo takiego zdarzenia jest dużo większe niż w przypadku innych serwisów o podobnym charakterze.

Ocena zagrożenia: Poważne.

Likwidacja skutków: Uzależniona od problemów, które wystąpiły pod nieobecność administratora. Zazwyczaj pojawia się dodatkowy problem poinformowania użytkowników o powrocie systemu do stanu działającego.

Środki zaradcze: Przeniesie serwera ze 100Mbit sieci lokalnej wydziału ETI do sieci ATM nowego centrum komputerowego TASK (nowy gmach ETI) za ok. 2 lata powinno zmniejszyć prawdopodobieństwo wystąpienia tego typu problemów.

Zagrożenie: Przejęcie hasła administratora systemu przez osobę niepowołaną.

Osoba odpowiedzialna za spowodowanie zagrożenia: Dowolny z administratorów systemu.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Nieodpowiedzialne działania administratora (np. logowanie do systemu korzystając z niezabezpieczonych połączeń) mogą doprowadzić do poznania jego hasła i przejęcia kontroli nad systemem przez osobę niepowołaną. W szczególności, bezpośrednią przyczyną utraty hasła może być logowanie do systemu z laboratoriów studenckich na wydziale ETI (ze względu na identyczność kont użytkowników na wszystkich komputerach, użytkownik dowolnego komputera dowolnego laboratorium jest w stanie podejrzeć i przejąć sesję użytkownika, poznając zarazem jego hasło, jeżeli uczyni to w momencie logowania).

Ocena zagrożenia: Poważne.

Likwidacja skutków: Uciążliwa, ze względu na niemożliwość stwierdzenia zakresu szkód wyrządzonych przez osobę mającą uprawnienia administratora. Może wymagać reinstalacji systemu z kopii zapasowych oraz zmiany haseł systemowych.

Środki zaradcze: Wyłączenie możliwości logowania do systemu przy wykorzystaniu niezabezpieczonych portów. Wprowadzenie opcjonalnych haseł jednorazowych o wysokiej sile szyfrowania, które można wykorzystać przy łączeniu się z systemem z miejsc zagrożonych atakiem.

Zagrożenia wynikające z fizycznego naruszenia bezpieczeństwa systemu

Zagrożenie: Fizyczna inwazja zaburzająca pracę systemu.

Osoba odpowiedzialna za dopuszczenie do ataku: Administrator budynku.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Naruszenie konfiguracji sprzętowej lub programowej systemu przez osobę niepowołaną, która uzyska fizyczny dostęp do pomieszczenia z węzłem systemu. W szczególności: próba odłączenia, kradzieży lub rozmontowania komputerów i osprzętu sieciowego; próba zmiany konfiguracji dyskowej.

Skutki: Atak zazwyczaj oznacza chwilowy paraliż systemu, czasami również utratę danych. Przerwa w działaniu systemu może być długotrwała.

Ocena zagrożenia: Umiarkowane (ze względu na niewielkie prawdopodobieństwo zdarzenia, łatwość wykrycia sprawcy).

Likwidacja skutków: Konieczne jest odtworzenie całej konfiguracji z kopii zapasowych oraz uzupełnienie brakującego sprzętu. Zdarzenie musi zostać zgłoszone odpowiednim organom ścigania.

Środki zaradcze: Ciągłe monitorowanie pracy serwera, pozwalające na natychmiastową kontrolę fizycznego stanu maszyny w przypadku nieprawidłowego działania. Współpraca administratorów serwera z administracją budynku celem zwiększenia bezpieczeństwa maszyn.

Zagrożenie: Fizyczna inwazja niezauważalna dla systemu

Osoba odpowiedzialna za dopuszczenie do ataku: Administrator budynku i Administrator systemu w wyniku niedopatrzeń.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Niedostateczne zabezpieczenie systemu w warstwie fizycznej, pozwalające atakującemu podsłuchać pakiety danych w obrębie sieci wewnętrznej, bądź uzyskać w sposób niezauważalny dostęp do danych na dyskach i w pamięci komputerów.

Skutki: Atak pozwala atakującemu uzyskać informacje niedostępne dla innych użytkowników systemu, co może oznaczać dla niego wymierną korzyść materialną.

Ocena zagrożenia: Poważne.

Likwidacja skutków: Uzależniona od rodzaju informacji zdobytej przez atakującego; może wymagać unieważnienia odbywających się zawodów.

Środki zaradcze: Uniemożliwienie ataku polegającego na podejrzeniu danych przetwarzanych przez komputer klasy PC jest w zasadzie niemożliwe w przypadku odpowiedniej determinacji atakującego. Można natomiast wprowadzić podstawowe środki ostrożności, które zmniejszą znacząco skuteczność ataków nie wykorzystujących specjalistycznego sprzętu. Należy tu wymienić przede wszystkim odpowiednie zabezpieczenie administracyjnych kont dostępowych (m.in. poprzez wzmocniony mechanizm autentykacji dla logujących się użytkowników), szyfrowaniu danych na dyskach twardych (wraz z plikiem wymiany) oraz eliminację informacji wysyłanej w postaci nieszyfrowanej na porty łatwe do podsłuchania (monitor, interfejs sieciowy podsieci wewnętrznej). Cała komunikacja w obrębie podsieci lokalnej powinna odbywać się w trwałych bezpiecznych kanałach, szyfrowanych w warstwie transportowej (lub wyższej), a informacja o otwieraniu i zamykaniu kanałów powinna być dostępna dla administratora. Dzięki takim zabezpieczeniom również atak polegający na podszyciu po stronie sieci wewnętrznej okaże się nieskuteczny.

Zagrożenia związane z atakami z sieci publicznej

Zagrożenie: Przechwycenie poufnej informacji przez osobę niepowołaną

Osoba odpowiedzialna za dopuszczenie do ataku: Nieostrożny użytkownik witryny internetowej.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Informacja przesyłana protokołem jawnym (typowo HTTP) w sieci publicznej może zostać przechwycona, np. przez osobę nasłuchującą ruch sieciowy w sieci wydziału ETI.

Skutki: Atak pozwala atakującemu uzyskać informację, która powinna być znana wyłącznie użytkownikowi komunikującemu się z systemem. W szczególności możliwe jest przechwycenie informacji wymaganej do uwierzytelnienia (co pozwala atakującemu na przejście uprawnień do konta w witrynie), lub innej informacji niejawnej (kodu źródłowego programu, danych testowych zadania).

Ocena zagrożenia: Poważne.

Likwidacja skutków: Uzależniona od rodzaju informacji zdobytej przez atakującego; może wymagać odłączenia konta użytkownika lub unieważnienia odbywających się zawodów.

Środki zaradcze: Wymaganie od użytkowników o szerszych uprawnieniach (m.in. organizatorów zawodów), by cała komunikacja następowała przy użyciu zabezpieczonej komunikacji w protokole HTTPS. Użytkownicy o mniejszych uprawnieniach powinni również wykorzystywać protokół HTTPS, przynajmniej do logowania.

Zagrożenie: Podszycie się przez atakującego pod innego użytkownika

Osoba odpowiedzialna za dopuszczenie do ataku: Nieostrożny użytkownik witryny internetowej, lub programista tworzący witrynę internetową.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Atak może zostać przeprowadzony na kilka różnych sposobów, m.in. poprzez przejście (przechwycenie) sesji komunikacyjnej lub innego rodzaju atak typu *man in the middle*, bądź poprzez wykorzystanie niedoskonałości oprogramowania witryny www do podmienienia informacji o zalogowanym użytkowniku (np. w zmiennych sesji serwera).

Skutki: Zbliżone do przypadku „przechwycenie poufnej informacji”.

Ocena zagrożenia: Poważne.

Likwidacja skutków: Zbliżona do przypadku „przechwycenie poufnej informacji”.

Środki zaradcze: Ograniczenie całej istotnej komunikacji w sieci publicznej do protokołu połączeniowego szyfrowanego SSL. Wykupienie wiarygodnego certyfikatu gwarantującego tożsamość serwera przy nawiązywaniu połączenia przez użytkowników.

Zagrożenie: Pozyskanie przez atakującego informacji z bazy danych na skutek błędu oprogramowania witryny.

Osoba odpowiedzialna za dopuszczenie do ataku: Programista tworzący witrynę internetową.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Błędy w kodach witryny internetowej (lub w platformie systemowej przez nią wykorzystywanej, np. w serwerze www) mogą pozwolić na pozyskanie przez atakującego haseł dostępowych do baz danych oraz pobranie lub modyfikację informacji w bazie danych.

Skutki: Przejęcie informacji w bazie danych przez atakującego w sposób niezauważalny dla systemu, lub zniszczenie tej informacji.

Ocena zagrożenia: Krytyczne.

Likwidacja skutków: Przywrócenie naruszonej bazy danych z kopii zapasowych, zmiana danych niejawnych, do których atakujący uzyskał dostęp.

Środki zaradcze: Opracowanie oprogramowania witryny w taki sposób, by błędy w większości modułów nie pozwalały atakującemu na dostęp do bazy danych. Przechowywanie rzadko wykorzystywanych danych o znaczeniu krytycznym (np. numerów kont bankowych użytkowników) w oddzielnej, zewnętrznej bazie danych. Nie przechowywanie w bazie danych haseł użytkowników, a jedynie ich funkcji skrótu.

Zagrożenie: Atak blokujący dostęp do systemu poprzez zalanie żądaniami.

Osoba odpowiedzialna za dopuszczenie do ataku: Administrator odpowiedzialny za konfigurację interfejsów sieciowych.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Nadmierna liczba żądań może spowodować spowolnienie lub zablokowanie dostępu do serwera www.

Ocena zagrożenia: Umiarkowane.

Likwidacja skutków: Wymaga interwencji administratora, w pewnych przypadkach (zmasowany atak DoS) może się okazać niemożliwa.

Środki zaradcze: Mechanizmy filtrowania i kształtowania ruchu podjęte na routerze dostępowym do systemu (ośrodek informatyczny TASK) oraz przez administratora wydziału

ETI. Odpowiednia konfiguracja firewalla na komputerze komunikującym się z siecią zewnętrzną (pełnienie roli bastion hosta odpornego na wiele rodzajów ataków).

Zagrożenie: Atak wykorzystujący słabość oprogramowania systemowego.

Osoba odpowiedzialna za dopuszczenie do ataku: Administrator odpowiedzialny za konfigurację interfejsów sieciowych.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Błędy w oprogramowaniu systemowym pozwalają atakującemu na naruszenie bezpieczeństwa systemu poprzez wysłanie odpowiednich pakietów do serwera.

Ocena zagrożenia: Poważne.

Środki zaradcze: Wielopoziomowy mechanizm filtrowania pakietów. Wysoki poziom zabezpieczenia firewall. Bezpieczna konfiguracja bastion hosta (jądro systemowe o podwyższonym bezpieczeństwie, regularnie aktualizowane oprogramowanie systemowe).

Zagrożenia wynikające ze specyfiki usług realizowanych przez portal

Zagrożenie: Nadesłanie przez użytkownika witryny internetowej złośliwego kodu do uruchomienia na maszynie testowej.

Osoba odpowiedzialna za dopuszczenie do ataku: Programista tworzący kod do testowania programów.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Administrator odpowiedzialny za bezpieczeństwo końcówek testowych.

Przyczyny: Błędy w oprogramowaniu zabezpieczającym uruchamianie programów na maszynach testowych, bądź w mechanizmach systemowych pozwalających na kontrolowane uruchamianie programów.

Ocena zagrożenia: Krytyczne.

Środki zaradcze: Wielopoziomowe zabezpieczenie systemu testującego przed atakiem przez program wykonywany na maszynie lokalnej przez użytkownika o ograniczonych przywilejach. Konfiguracja interfejsów sieciowych pozostałych maszyn, tak, by przejęcie kontroli przez atakującego nad maszyną testową nie pozwalało na naruszenie bezpieczeństwa pozostałych maszyn w prywatnej podsięci.

Zagrożenie: Blokowanie przez użytkownika witryny internetowej dostępu do końcówek sprawdzających poprzez zalewanie żadaniami testowania kodu.

Osoba odpowiedzialna za dopuszczenie do ataku: Programista tworzący kod witryny internetowej.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Skutki: Ciągłe nadsyłanie programów do testowania w kilkusekundowych odstępach może spowodować wyłączenie innych użytkowników i utworzyć kolejkę programów do sprawdzenia.

Ocena zagrożenia: Poważne.

Środki zaradcze: Zabezpieczenie w warstwie logiki systemu (ograniczanie liczby nadsyłanych rozwiązań lub liczby żądań www (dla pojedynczego adresu IP lub podsięci adresów IP). Ograniczenie możliwości masowego zakładania kont użytkowników i ograniczenie liczby rozwiązań nadsyłanych przez użytkownika w określonej jednostce czasu. Priorytetyzacja użytkowników w zależności od generowanego przez nich obciążenia.

Zagrożenie: Nadużycie uprawnień przez organizatora zawodów na serwerze.

Osoba odpowiedzialna za dopuszczenie do ataku: Programista tworzący kod witryny internetowej.

Osoba odpowiedzialna za przeciwdziałanie skutkom: Dyżurny administrator systemu.

Przyczyny: Zezwolenie organizatorom zawodów na opracowywanie własnych skryptów do generowania stron WWW na serwerze może doprowadzić do ataku na serwer lub zmniejszenia jego wydajności.

Skutki: Naruszenie bezpieczeństwa systemu lub zmniejszenie jego wydajności.

Ocena zagrożenia: Poważne.

Środki zaradcze: Zdefiniowanie własnego, prostego języka skryptowego do zastosowania przez osoby przygotowujące strony internetowe zawodów. Właściwe zabezpieczenie interpretera tego języka (np. poprzez jego realizację i uruchomienie w zabezpieczonej maszynie wirtualnej języka Java) oraz modułów kodu ściśle z nim powiązanych.

3.2. Identyfikacja głównych słabości systemu

Krytyczne aspekty bezpieczeństwa danych

Na podstawie przeprowadzonej w punkcie 3.1 analizy nietrudno stwierdzić, że w celu zagwarantowania bezpieczeństwa portalu konieczna jest przede wszystkim:

- 1) wielostopniowa ochrona danych przez sieciowe i systemowe mechanizmy zabezpieczeń, realizowane przez administratorów,
- 2) bezbłądność kodu witryny i kodu do testowania programów w sekcjach o krytycznym znaczeniu (zarządzanie danymi w bazie oraz uruchamianie procesów użytkownika),
- 3) zachowanie podstawowych zasad ostrożności przez administratorów oraz uprzywilejowanych użytkowników systemu.

Aspekty 1) i 2) zostały omówione bardziej szczegółowo w punkcie 2.5.

Problemy z wydajnością i *wąskie gardła* systemu

Wydajna realizacja komponentów oprogramowania wchodzących w skład portalu stanowi stosunkowo obszerne zagadnienie. Z punktu widzenia bezpieczeństwa i ewentualnej podatności systemu na ataki Denial of Service, problem należytego zabezpieczenia wszystkich trzech rodzajów komputerów wchodzących w skład portalu jest jednakowo istotny. Serwer brzegowy musi być, poprzez odpowiednią konfigurację oprogramowania sieciowego, przystosowany do wydajnego wycinania ruchu sieciowego nie przedstawiającego żadnej wartości informacyjnej (pakiety kierowane na zamknięte porty, błędne żądania, etc.). Wycinanie takiego ruchu powinno odbywać się wieloetapowo – na routerach dostępowych providera Internetu, w niskopoziomowych mechanizmach zabezpieczeń serwera w przestrzeni jądra (reguły firewalla), a dopiero w ostateczności – na poziomie aplikacji serwera WWW.

Szczególnie nieprzyjemny rodzaj ataku zorientowany jest w serwer RDBMS i polega na zalewaniu serwera WWW prawidłowymi, ale zróżnicowanymi żadaniami pobrania stron internetowych, które wymagają realizacji skomplikowanych zapytań w bazie danych. Jedyną obroną wobec takiego ataku wydaje się być połączenie zaawansowanych reguł kształtowania ruchu (obniżanie priorytetu żądań pochodzących od jednego hosta w miarę ich napływania) oraz mechanizmów przechowywania wyników zapytań w pamięci tymczasowej, zarówno po stronie serwera WWW jak i serwera SQL. Odpowiednia realizacja takich zabezpieczeń

pozwole również na uchronienie serwisu przed przeciążeniem na skutek nagłej, niespodziewanej dużej liczby wizyt (zjawisko popularnie zwane efektem serwisu slashdot).

Zapewnienie sprawnego funkcjonowania końcówek testowych jest również niezwykle istotne. W przeciwieństwie do pozostałych serwerów usług, dopuszcza się tutaj tworzenie krótkich (kilkudziesięciosekundowych) kolejek programów oczekujących na testowanie i sprawdzenie. Zagrożenie atakiem DoS można skutecznie zmniejszyć poprzez programową priorytetyzację zgłoszeń w zależności od wcześniejszej aktywności danego użytkownika (lub jego lokalnej podsieci adresów IP).

3.3. Bezpieczeństwo danych w portalu

Większość niejawniej przetwarzanej w portalu przechowywane jest w głównej bazie danych na serwerze RDBMS. Dostęp do bazy możliwy jest wyłącznie przez lokalny serwer SQL. Usługa serwera danych portalu uruchomiona na serwerze RDBMS ma za zadanie przetwarzanie i realizację zapytań kierowanych przez usługi działające na innych komputerach, stanowiąc zarazem warstwę pośredniczącą zabezpieczającą operacje wykonywane na bazie danych. W szczególności, zadaniem serwera danych jest kontrola uprawnień użytkowników operujących na danych poprzez system uprawnień typu Access Control Lists przypisanych do struktury wirtualnych plików zapisanych w bazie danych.

Mając na uwadze możliwość wystąpienia potencjalnych błędów w oprogramowaniu, niektóre dane o znaczeniu krytycznym z punktu widzenia bezpieczeństwa są dodatkowo zabezpieczane. Przykładowo, hasła przechowywane są w bazie w postaci 160-bitowych funkcji skrótu SHA1, natomiast rzadko wykorzystywane dane osobowe podlegające ochronie prawnej przechowywane są w oddzielnej bazie danych.

3.4. Kontrolowanie uruchamianych zadań

Brak zaufania do wykonywanych procesów leży u podstaw polityki bezpieczeństwa portalu. Najistotniejsze jest zabezpieczenie komputerów mających bezpośrednią styczność z kodem nadesłanym przez użytkownika, tj. zabezpieczenie usług www serwera brzegowego oraz środowisk uruchamiania programów na końcówkach testowych

Zabezpieczenie usług www

Przyjęty model serwera www jest oparty o architekturę kilkuwarstwową. Przetwarzaniem żądań HTTP zajmuje się serwer www Apache w wersji 2, wraz z zainstalowanymi rozszerzeniami (np. mod_gzip do zmniejszania rozmiaru wysyłanego strumienia danych). Generowaniem kodu HTML dla żądania zajmuje się rozszerzenie serwera Apache Jakarta/Tomcat, pozwalające na realizację żądań poprzez skrypty JSP i serwlety Javy. W szczególności, bezpieczny serwlet wykorzystywany jest do interpretowanego uruchomienia skryptów stosowanych w zawodach przez ich organizatorów, pozwalając im na wykorzystywanie zewnętrznych zasobów jedynie w ograniczonym zakresie. W szczególności, dopuszcza się możliwość kontrolowanej interakcji skryptu z serwerem danych portalu. Pobieranie i modyfikacja danych przez skrypt odbywa się przy założeniu uprawnień albo użytkownika wykonującego skrypt, albo jego autora, co praktycznie uniemożliwia ataki polegające na przekroczeniu uprawnień.

Zabezpieczenie końcówek testowych

Uruchamianie kodu aplikacji nadsyłanych przez użytkownika jest bez wątpienia najbardziej niebezpiecznym elementem pracy systemu. Aby zapewnić możliwie wysoki poziom bezpieczeństwa podejmuje się następujące środki ostrożności:

- Wszystkie fazy związane z przetwarzaniem kodu użytkownika są traktowane jako niebezpieczne, włącznie z kompilacją kodu, która odbywa się z analogicznymi zabezpieczeniami, jak uruchamiany program.
- Przed uruchomieniem każdego programu wszelkie dane są archiwizowane na bezpiecznym komputerze.
- Komputery wykorzystywane do sprawdzania programów są możliwie jak najbardziej odizolowane od świata. Nie mają połączenia z siecią zewnętrzną, a komunikacja w sieci prywatnej ma ograniczony zakres (punkt 2.5). Komputery te nie pełnią też żadnych funkcji poza testowaniem poprawności programów.
- Ze względu na dużą liczbę języków programowania dostępnych do wykorzystania, jako gwarancję bezpieczeństwa służą mechanizmy systemu operacyjnego a nie ograniczenia narzucane w kompilatorze, interpreterze, czy bibliotekach. Pozwala to uniknąć podatności systemu na błędy w oprogramowaniu uruchomieniowym poszczególnych języków.
- Realizacja odpowiednich zabezpieczeń w systemie operacyjnym Debian GNU/Linux (jądro z serii 2.4 z dodatkowymi łataniami) wykracza poza możliwości zdefiniowane w

standardzie POSIX. Ograniczenie uruchamianemu programowi dostępu do usług sieciowych na tym samym lub innych komputerach realizowane jest przez analogię do wywołania systemowego *jail()* znanego z systemów BSD (niedostępnego w systemie Linux). W systemie zastosowano łąty jądra zwiększające bezpieczeństwo (*grsecurity*) oraz ograniczenia w tworzeniu gniazd przez konkretnych użytkowników. Dodatkowo, dodano specjalny moduł do mechanizmu *iptables* i wprowadzono blokowanie pakietów konkretnych użytkowników na firewallu komputera testowego.

- Do ograniczenia swobody działania testowanego procesu wykorzystano podstawowy mechanizm systemu Linux, *RLIMIT*. Pozwala on na ograniczenie całkowitej alokowanej pamięci, czasu procesora, wielkości tworzonych plików oraz ilości procesów potomnych tworzonych przez proces.
- Dodatkowe zabezpieczenia wprowadzono w systemie plików. Ustawiono limit *quota* na sumaryczną wielkość plików i liczbę plików, a dla uruchamianych programów ograniczono zakres widoczności systemu plików poprzez polecenie *chroot()*. Dla każdego dostępnego w systemie kompilatora, interpretera i środowiska dla plików wykonywalnych istnieje specjalnie przygotowane „więzienie” *chroot*, zawierające minimalną konfigurację bibliotek i usług. W szczególności, więzienia mają okrojoną liczbę plików urządzeń (katalog */dev*) oraz nie posiadają systemu plików *proc*.
- Ograniczenie czasu działania procesu realizowane jest zasadniczo poprzez *RLIMIT_CPU*. Jednak ze względu na sytuacje, w których program może działać długo przy krótkim czasie użytkownika (np. w przypadku częstego używania wywołania systemowego *sleep()* lub korzystania z blokującego wejścia/wyjścia) konieczne jest również ograniczenie rzeczywistego czasu działania programu (wall time).
- Dopuszczenie języków wykorzystujących wiele procesów przy uruchomieniu prostego programu (np. C#, Java) wprowadza znaczące utrudnienia dla polityki bezpieczeństwa uruchomienia. W szczególności, w systemie Linux brak bezpiecznego mechanizmu zliczania użytych przez proces wraz z wszystkimi potomkami zasobów. Rozwiązanie stanowi tutaj wykorzystanie mechanizmu *BSD Process Accounting* (dostępnego również na platformie Linux), które loguje działanie wszystkich procesów w systemie. Analiza danych zapisanych w logach możliwa jest poprzez ich filtrowanie względem identyfikatora właściciela procesu (*UID*). Efektem ubocznym jest konieczność testowania kolejnych programów korzystając z różnych kont w pewnej puli użytkowników (pozwala

to uniknąć wyścigów i kłopotów związanych z ustaleniem faktycznej przynależności procesu do bieżącego testowania).

- W przypadku konieczności zabicia programu użytkownika (np. na skutek wyczerpania limitu czasu), stosowany jest probabilistycznie pewny mechanizm wyłączania wszystkich procesów użytkownika. Atomowa operacja wysłania sygnału *SIGKILL* do wszystkich procesów danego użytkownika przez program działający z uprawnieniami nadzorcy systemu wymaga przejścia programu ubijającego do trybu odpowiedniego użytkownika (*setuid()*), a następnie wywołanie polecenia zabicia wszystkich procesów tego samego użytkownika (*kill(-1, SIGKILL)*). Aby zminimalizować ryzyko ataku jednego z ubijanych procesów na proces ubijający, w chwili, gdy znajduje się on już w puli procesów nieuprzywilejowanego użytkownika, konieczna jest losowa rotacja numerów procesów *PID*, możliwa dzięki mechanizmom *grsecurity*.

Dodatkowe środki bezpieczeństwa obejmują m.in. zamykanie wszystkich, poza określonymi, deskryptorów plików, czyszczenie zmiennych środowiskowych oraz uczynienie testowanego programu *session leaderem* swojej własnej sesji. Wszystkie próby naruszenia bezpieczeństwa lub przekroczenia limitów przez program testowany są zapisywane w logach systemu sterującego testowaniem.

3.5. Uwagi końcowe

Niniejszy dokument projektowy zawiera specyfikację konfigurację urządzeń sieciowych i oprogramowania systemowego, oraz szczegółowy opis polityki bezpieczeństwa dla portalu Sphere Online Judge. W chwili obecnej sieć portalu złożona jest z pięciu komputerów (komputera łączącego funkcje serwera brzegowego i RDBMS oraz czterech końcówek testowych) i niezarządzalnego switcha. Rozbudowa portalu w sposób zarysowany w tym projekcie nastąpi prawdopodobnie pod koniec bieżącego roku, i będzie przebiegać stopniowo, w miarę pozyskiwania funduszy na inwestycję. W pierwszej kolejności przewiduje się rozdzielenie serwera brzegowego i serwera danych, a w dalszym ciągu zwiększenie liczby końcówek testowych i przystosowanie niektórych do pracy w trybie interaktywnym, oraz zakup switcha zarządzalnego w celu podniesienia bezpieczeństwa systemu.

Załącznik. Reguły firewalla dla serwera brzegowego

```
#!/bin/bash

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Ograniczenie ruchu ICMP/PING
ICMP_LIMIT_INPUT="50/second"
ICMP_LIMIT_OUTPUT="50/second"

IPT=/usr/local/sbin/iptables
MODPROBE=/sbin/modprobe
LO_IF="lo"

# Interfejs eth0 - połączenie z siecią zewnętrzną
EXT_IF="eth0"
EXT_IP="153.19.53.9"

# Interfejs eth1 - połączenie z podsiecią prywatną
INT_IF="eth1"
INT_NETWORK="192.168.1.0/24"

# Porty serwera otwarte dla sieci zewnętrznej (SSH, HTTP, poczta, inne)
IN_EXT_TCP1=ssh,http,https,ftp,ftp-
data,smtp,pop3s,imap2,imaps,auth,cvspserver,domain
IN_EXT_TCP2=6881:6999,40000:45000
IN_EXT_UDP1=ssh,pop3s,imap2,imaps,cvspserver,6881:6999,domain,40000:45000

# Porty serwera otwarte dla sieci wewnętrznej (DNS, serwer czasu, serwer
# proxy aktualizacji pakietów systemowych, serwer danych SPOJ)
IN_INT_TCP=apt-proxy,domain,spojs
IN_INT_UDP=domain,spojs,ntp

OUT_INT_TCP=
OUT_INT_UDP=

# Flaga zezwalająca na komunikację pomiędzy podsiecią prywatną
# a siecią publiczną
INT_INET=1

case "$1" in
    start|restart|reload|force-reload)

        echo "Loading iptables rules."
```

```

# --- POCZĄTEK ŁADOWANIA REGUŁ ---

# Czyszczenie reguł iptables
$IPT -F
$IPT -F -t nat
$IPT -F -t filter
$IPT -F -t mangle
$IPT -X
$IPT -X -t nat
$IPT -X -t filter
$IPT -X -t mangle

# Ustawienie domyślnej polityki filtrowania pakietów
$IPT -P FORWARD DROP
$IPT -P INPUT DROP
$IPT -P OUTPUT ACCEPT

# Reguły końcowe łańcuchów
$IPT -N drop-and-log-it
$IPT -A drop-and-log-it -m limit --limit 1/hour -j ULOG --ulog-prefix \
REJECT
$IPT -A drop-and-log-it -j REJECT

$IPT -N port_scan
$IPT -A port_scan -m limit --limit 1/minute -j ULOG --ulog-prefix psd

# ROUTING WSTĘPNY (podsieć prywatna)

# Ochrona przed spoofingiem IP w podsieci prywatnej
$IPT -t nat -A PREROUTING -i $EXT_IF -s $INT_NETWORK -j DROP
$IPT -t nat -A PREROUTING -i $INT_IF -s ! $INT_NETWORK -j DROP

# Ustawianie nagłówka typu usługi (TypeOfService), część 1/2
$IPT -t mangle -A PREROUTING -p tcp --dport ftp-data -j TOS --set-tos 8
$IPT -t mangle -A PREROUTING -p tcp --dport ftp -j TOS --set-tos 16
$IPT -t mangle -A PREROUTING -p tcp --dport ssh -j TOS --set-tos 16
$IPT -t mangle -A PREROUTING -p tcp --dport smtp -j TOS --set-tos 16
$IPT -t mangle -A PREROUTING -p tcp --dport http -j TOS --set-tos 8
$IPT -t mangle -A PREROUTING -p tcp --dport https -j TOS --set-tos 8

# Forwardowanie portu do podsieci prywatnej (na przykładzie końcówki
# testowej 192.168.1.2) w celu umożliwienia interakcji
$IPT -t nat -A PREROUTING -d 153.19.53.141 -p tcp --dport 10001 -j DNAT\
--to 192.168.1.2:80

# REGUŁY: FORWARD (podsieć prywatna)

```

```

if [ $INT_INET == 1 ]; then
    $IPT -t filter -A FORWARD -i $INT_IF -o $EXT_IF -s $INT_NETWORK -j ACCEPT
    $IPT -t filter -A FORWARD -o $INT_IF -i $EXT_IF -d $INT_NETWORK -j ACCEPT
fi

$IPT -t filter -A FORWARD -j drop-and-log-it

# REGUŁY: INPUT

$IPT -A INPUT -i $LO_IF -j ACCEPT

$IPT -A INPUT -m state --state INVALID -j DROP
$IPT -A INPUT -m unclean -j DROP

# Ograniczenie ruchu ICMP
$IPT -A INPUT -p icmp -m limit --limit $ICMP_LIMIT_INPUT -j ACCEPT
$IPT -A INPUT -p icmp -j DROP

# Ustanowione połączenia
$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Skanowanie portów
$IPT -A INPUT -m psd -j port_scan

# Zezwalanie na otwarcie portów
$IPT -A INPUT -i $EXT_IF -m mport -p tcp --dports $IN_EXT_TCP1 -j ACCEPT
$IPT -A INPUT -i $EXT_IF -m mport -p tcp --dports $IN_EXT_TCP2 -j ACCEPT
$IPT -A INPUT -i $EXT_IF -m mport -p udp --dports $IN_EXT_UDP1 -j ACCEPT
$IPT -A INPUT -i $INT_IF -m mport -p tcp --dports $IN_INT_TCP -j ACCEPT
$IPT -A INPUT -i $INT_IF -m mport -p udp --dports $IN_INT_UDP -j ACCEPT

# Zamykanie pozostałych portów
$IPT -A INPUT -j drop-and-log-it

# REGUŁY: OUTPUT

$IPT -A OUTPUT -o $LO_IF -j ACCEPT

$IPT -A OUTPUT -p icmp -m limit --limit $ICMP_LIMIT_OUTPUT -j ACCEPT
$IPT -A OUTPUT -p icmp -j DROP

# Ustawianie nagłówka typu usługi (TypeOfService), część 2/2
$IPT -t mangle -A OUTPUT -p tcp --dport ftp-data -j TOS --set-tos 8
$IPT -t mangle -A OUTPUT -p tcp --dport ftp -j TOS --set-tos 16
$IPT -t mangle -A OUTPUT -p tcp --dport ssh -j TOS --set-tos 16
$IPT -t mangle -A OUTPUT -p tcp --dport smtp -j TOS --set-tos 16
$IPT -t mangle -A OUTPUT -p tcp --dport http -j TOS --set-tos 8

```



```

$IPT -t mangle -A OUTPUT -p tcp --dport https -j TOS --set-tos 8

# Ustawione połączenia
$IPT -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Otwarte porty
if [ $OUT_INT_TCP ]; then
    $IPT -A OUTPUT -o $INT_IF -m mport -p tcp --dports $OUT_INT_TCP -j ACCEPT
fi
if [ $OUT_INT_UDP ]; then
    $IPT -A OUTPUT -o $INT_IF -m mport -p udp --dports $OUT_INT_UDP -j ACCEPT
fi

# Zamykanie pozostałych portów
$IPT -A OUTPUT -o $INT_IF -j REJECT

# REGUŁY: POSTROUTING (podsieć prywatna)

if [ $INT_INET == 1 ]; then
    $IPT -t nat -A POSTROUTING -o $EXT_IF -s $INT_NETWORK -j SNAT \
        --to-source $EXT_IP
fi

# --- KONIEC ŁADOWANIA REGUŁ ---

;;
stop)
;;
*)
    echo "Aborting iptables initd: unknown command(s): \"$@\"."
;;
esac

exit 0

```