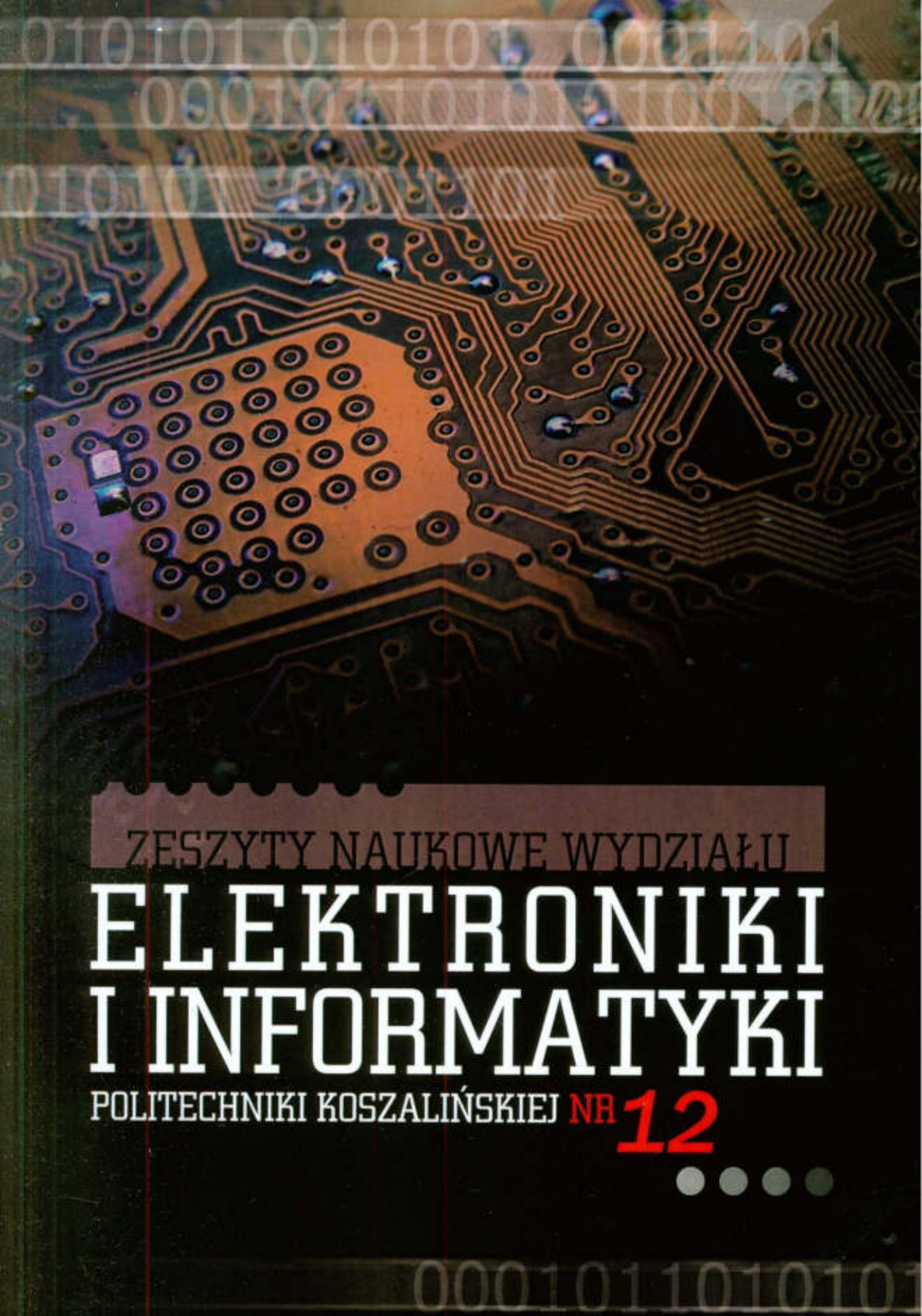


ZESZYTY NAUKOWE WYDZIAŁU

ELEKTRONIKI I INFORMATYKI

POLITECHNIKI KOSZALIŃSKIEJ NR 12

POLITECHNIKA KOSZALIŃSKA

Zeszyty Naukowe
Wydziału Elektroniki i Informatyki

Nr 12

KOSZALIN 2018

Zeszyty Naukowe Wydziału Elektroniki i Informatyki Nr 12

ISSN 1897-7421
ISBN 978-83-7365-492-1

Strona internetowa
https://weii.tu.koszalin.pl/nauka/zeszyty-naukowe

Projekt okładki
Tadeusz Walczak

Skład, łamanie
Maciej Bączek

Wydawnictwo Uczelniane Politechniki Koszalińskiej
75-620 Koszalin, ul. Racławicka 15-17

Koszalin 2018, wyd. I, ark. wyd. 2,92, format B-5, nakład 100 egz.
Druk: INTRO-DRUK, Koszalin

Spis treści

Jakub Ślepecki
Michał Rydzewski
Paweł Kisiel
Paweł Poczekajło
Studenckie Koło Naukowe Pasjonatów Elektroniki
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. JJ Śniadeckich 2, 75-453 Koszalin

Prosta gra zręcznościowa typu "arcade" w oparciu o moduły sterujące z mikroprocesorami AVR

Słowa kluczowe: automat do gier, arcade, Arduino, AVR

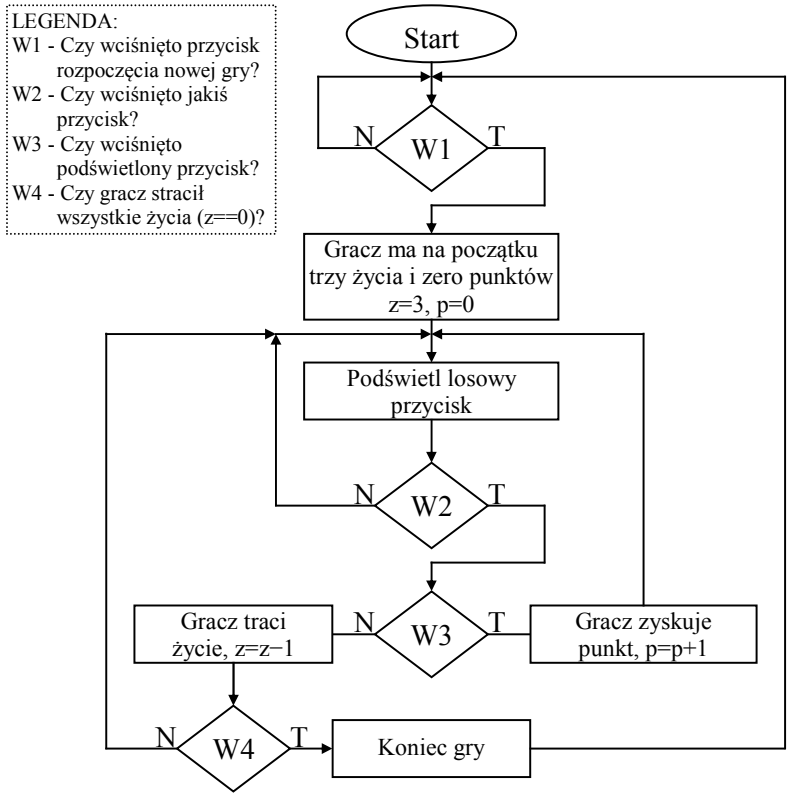
1. Wstęp

Automaty do gier typu "arcade" to niezależne urządzenia przeznaczone do celów rozrywkowych. Zazwyczaj elementy sprzętowe (np. przyciski, wyświetlacze) dostosowane są do danej gry. Urządzenia tego typu najczęściej można spotkać we wszelkiego rodzaju salonach gier. W ostatnich latach automaty do gier przeżywają renesans i coraz chętniej są wybierane jako ciekawa alternatywa wobec gier konsolowych i PC. Popularyzacja techniki mikroprocesorowej i znaczny spadek kosztów stosowania takich układów (idealnym przykładem jest tu Arduino [1]) sprawiły, że powstaje coraz więcej amatorskich konstrukcji typu "arcade". Urządzenia takie są również popularnym tematem projektów studenckich oraz elementem prac inżynierskich.

Niniejszy artykuł prezentuje projekt Studenckiego Koła Naukowego Pasjonatów Elektroniki, które działa przy Katedrze Systemów Cyfrowego Przetwarzania Sygnałów na Wydziale Elektroniki i Informatyki Politechniki Koszalińskiej. Celem przedsięwzięcia było skonstruowanie prostego automatu do gier typu "arcade" realizującego wybraną grę zręcznościową, bazując na mikrokontrolerach z rodziny AVR [2, 3]. Rozgrywka polega na jak najszybszym wciśnięciu wskazanych przycisków z macierzy o wymiarach 3×3 (włączniki są podświetlane). W kolejnych rozdziałach przedstawiono algorytm gry, projekt automatu oraz gotowe urządzenie.

2. Projekt gry

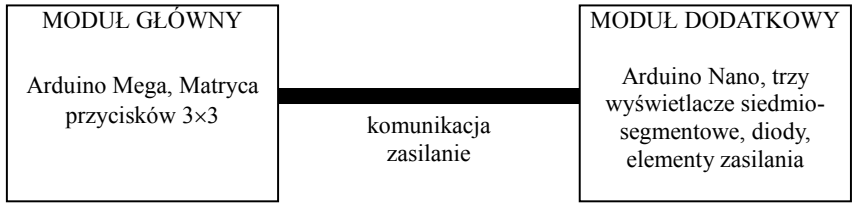
Automat realizuje prostą grę zręcznościową, w której gracz musi wykazać się refleksem (szybkością reakcji). Przyciski umieszczone w macierzy 3×3 są podświetlane w pseudolosowej kolejności przez krótki czas. W danej chwili świeci się tylko jeden przycisk. Zadaniem gracza jest naciśnięcie go. Za każdy naciśnięty zapalony przycisk otrzymuje on jeden punkt, za naciśnięcie niezapalonego – traci jedno z trzech żyć, a za ominięcie (pozwolenie, by przycisk sam zgasł z upływem czasu) traci punkt. Co dziesięć zdobytych punktów, na 100 kończąc, czas świecenia się przycisków jest skracany, co zwiększa trudność gry. Rozgrywka kończy się po utracie ostatniego z trzech żyć. Wówczas przyciski zaczynają mrugać w sekwencji ustalonej dla stanu oczekiwania na kolejną grę. Aby ją rozpocząć należy wcisnąć środkowy przycisk. Schemat blokowy przebiegu gry (algorytm) zaprezentowano na rysunku 1.



Rys. 1. Schemat blokowy algorytmu gry

3. Projekt automatu

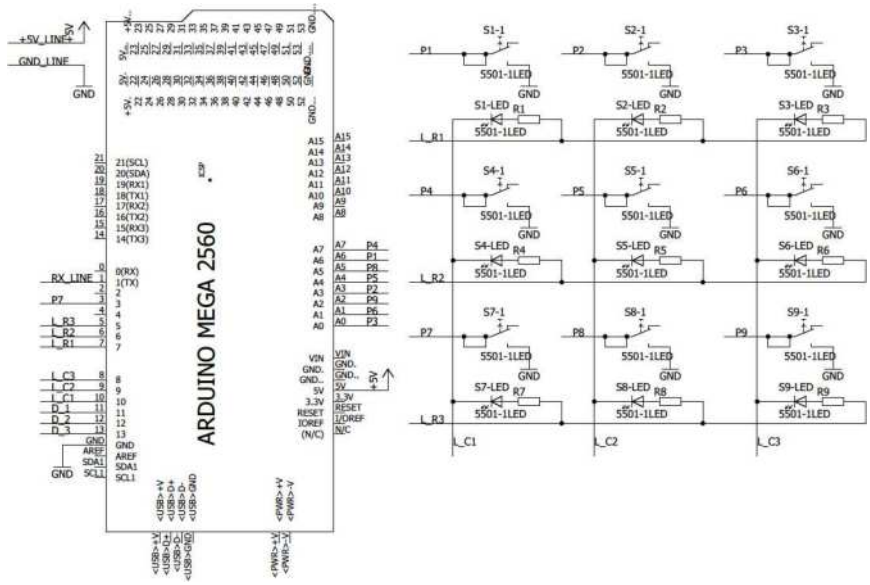
Kluczowym etapem realizacji przedstawionego projektu było wykonanie części sprzętowej automatu. Zdecydowano się na maksymalne wykorzystanie posiadanych już elementów, tak aby zminimalizować koszty. Takie założenie niestety narzucało niektóre rozwiązania, co z kolei wymagało nieszablonowego podejścia do wybranych problemów konstrukcyjnych. Generalnie, cały projekt został podzielony na dwie części. Pierwsza (główna) to moduł sterujący, który kontroluje grę oraz pracę przycisków (w tym również diod podświetlających przyciski). Druga (dodatkowa) to moduł obsługujący m.in. wyświetlacze siedmio-segmentowe. Na rysunku 2 zaprezentowano ogólny schemat automatu, a poniżej szczegółowo przedstawiono poszczególne rozwiązania.



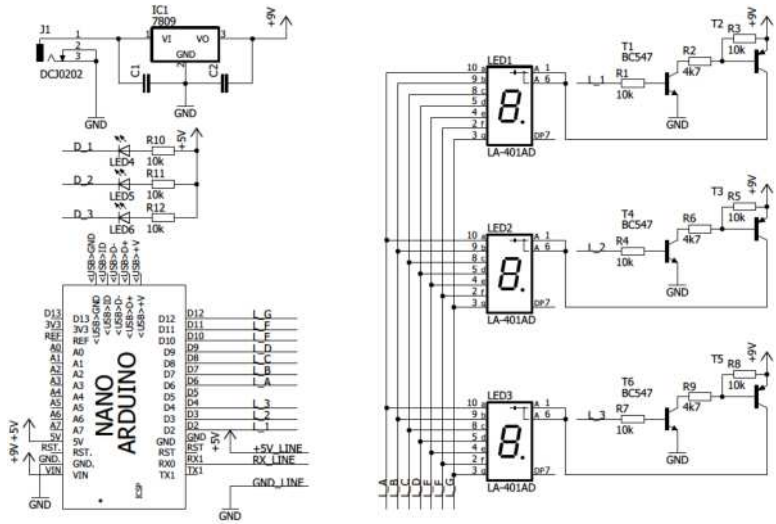
Rys. 2. Ogólny schemat blokowy automatu

3.1. Moduł sterujący (główny)

Moduł główny składa się z dwóch elementów bazowych: płytki mikro-procesorowej oraz macierzy przycisków. Zastosowane przyciski są włącznikami mono-stabilnymi typu "big button" z wbudowanym podświetleniem, które w tym wypadku jest sygnalizatorem wskazanego do naciśnięcia przycisku. Przełączniki są typu NO (ang. normally open). Całym urządzeniem steruje płytka prototypowa Arduino Mega (klon) z mikroprocesorem AVR Atmega2560. Wykorzystanie gotowej płytki pozwoliło znacznie uprościć i przyspieszyć pracę nad urządzeniem. Na etapie prac prototypowych zastosowano układ Arduino Uno z procesorem Atmega328, jednak w miarę rozwoju kodu programu, okazało się że 32kB pamięci FLASH (pamięć programu) i 2kB pamięci SRAM (pamięć zmiennych) to niestety za mało. Naturalnym byłoby zastosowanie nieco "większego" procesora, tj. Atmega64 lub Atmega128, jednak w ofercie gotowych płytek z rodziny Arduino był już tylko zestaw z procesorem Atmega2560. Na niekorzyść rozmiaru programu i zmiennych działał również fakt zastosowania środowiska Arduino IDE, które przy stosowaniu standardowych bibliotek nie jest zbyt ekonomiczne pod kątem zużywania zasobów pamięciowych procesora. Na rysunku 3 przedstawiono schemat elektroniczny modułu sterującego.



Rys. 3. Schemat elektroniczny modułu głównego



Rys. 4. Schemat elektroniczny modułu dodatkowego

3.2. Moduł wyświetlaczy (dodatkowy)

Wyświetlanie wyniku odbywa się za pomocą trzech wyświetlaczy siedmiosegmentowych o wymiarach 47×70mm, które wraz ze sterownikiem wyświetlaczy

stanowią moduł dodatkowy. Za obsługę tego modułu odpowiada płytka Arduino Nano z procesorem Atmega328. Wszystko zawarte jest w osobnej obudowie, gdzie dodatkowo umieszczono trzy diody LED sygnalizujące bieżącą liczbę żyć oraz elementy zasilania. Na rysunku 4 przedstawiono schemat modułu wyświetlaczy.

4. Gotowe urządzenie

Ze względu na ograniczone środki finansowe, obudowa finalnego urządzenia została okrojona do minimum. W obecnej wersji składa się z podstawy wykonanej ze sklejki umieszczonej na regulowanych nóżkach. Moduł główny przymocowany jest od spodu. Przyciski z podświetleniem były oryginalnie dostosowane do montażu w ściance obudowy, więc nie było kłopotu z zainstalowaniem ich na sklejce. Moduł dodatkowy wraz z wyświetlaczami w odpowiedniej obudowie umieszczono na górze, tak aby bieżący wynik był dobrze widoczny dla gracza. Na rysunku 5 przedstawiono zbudowany automat do gry.

Rys. 5. Gotowy automat do gry zręcznościowej

5. Podsumowanie

Wykonany automat do gry działa prawidłowo i zgodnie z zamierzeniami projektantów. Jego funkcjonowanie zostało zweryfikowane praktycznie podczas

Dnia Otwartego Politechniki Koszalińskiej, który odbył się 8 marca 2018 r. Zwiedzający i goście mieli możliwość sprawdzenia swoich umiejętności i refleksu w rozgrywce na zbudowanym automacie.

Jest to pierwszy automat tego typu wykonany przez Koło, więc projekt ten był dla autorów doskonałym wstępem do techniki mikroprocesorowej i znakomitą okazją do poszerzenia swojej wiedzy z dziedziny szeroko pojętej elektroniki i informatyki. W dalszych planach jest stworzenie kolejnych automatów oraz wystawienie ich podczas corocznych Juwenaliów Politechniki Koszalińskiej.

Bibliografia

1 Online: Arduino - Home https://www.arduino.cc/ (14.03.2018)
2 Doliński J.: *Mikrokontrolery AVR w praktyce*, Wydawnictwo BTC, Warszawa (2004), ISBN: 83-910067-6-X
3 Kardaś M.: *Mikrokontrolery AVR Język C Podstawy programowania wyd. II*, Wydawnictwo Atnel, Szczecin (2013), ISBN: 978-83-931797-2-5

Streszczenie

W artykule przedstawiono projekt wykonany przez Studenckie Koło Naukowe Pasjonatów Elektroniki, które działa przy Katedrze Systemów Cyfrowego Przetwarzania Sygnałów na Wydziale Elektroniki i Informatyki Politechniki Koszalińskiej. Niniejsze przedsięwzięcie polegało na zaprojektowaniu i skonstruowaniu prostego automatu do gry typu "arcade". Urządzenie ma realizować rozgrywkę zręcznościową sprawdzającą refleks gracza. Automat bazuje na procesorach Atmel AVR i płytkach prototypowych z rodziny Arduino. Zastosowano również wyświetlacze siedmio-segmentowe oraz przyciski typu "big button".

Abstract

In this paper, the arcade game project is presented. It was created by Students' Science Club of Enthusiasts of Electronics in Faculty of Electronics and Computer Science, Koszalin University of Technology. The gameplay was made to test reaction times of the players. Atmel AVR microcontrollers and Arduino prototype boards were used in the machine. Additionally seven-segments displays and illuminate push buttons were used.

Keywords: arcade game, Arduino, AVR

Paweł Poczekajło
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. JJ Śniadeckich 2, 75-453 Koszalin

Prototyp sterownika jednokolorowego wyświetlacza LED opartego na mikrokontrolerze AVR i układach MAX7219

Słowa kluczowe: wyświetlacz LED, AVR, MAX7219, macierz LED 8×8

1. Wprowadzenie

Wszelkiego rodzaju wyświetlacze LED (zarówno te jednokolorowe jak i RGB) stały się w ostatnich latach jedną z najpopularniejszych form prezentowania różnych informacji. Zastosowania są bardzo szerokie, od reklam i interaktywnych bilbordów, po tablice informacyjne (np. na autostradach lub w urzędach). Nawet na krajowym rynku dostępnych jest wiele gotowych rozwiązań np. takich firm jak RGB Technology [1], czy GilBT [2].

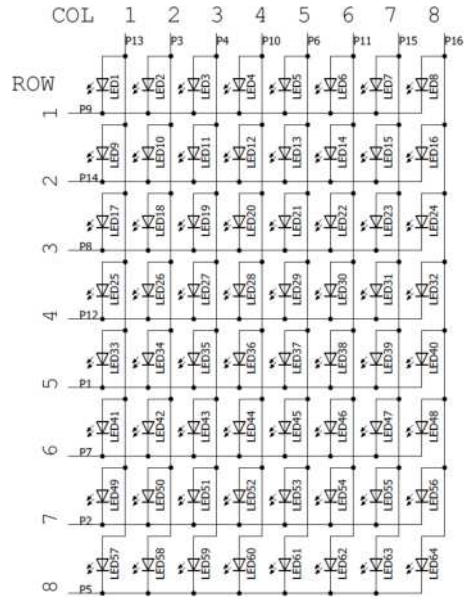
Niestety dużym minusem takich urządzeń jest ich cena, zwłaszcza w przypadku dużych wielokolorowych wyświetlaczy może ona wynosić nawet kilkadziesiąt tysięcy złotych. Wyświetlacze jednokolorowe mogą być sporo tańsze, jednak przy nieco większych wymiarach ich cena również może być znaczna i wynosić nawet kilkanaście tysięcy złotych. Budowa tego typu wyświetlaczy jest zazwyczaj bardzo prosta. Podstawowe elementy składowe to macierze LED (o różnych wymiarach od 8×8 do 64×32), sterowniki pojedynczych macierzy oraz sterownik główny (zapewniający użytkownikowi kontrolę nad wyświetlaczem) i moduł zasilania. Dostępność poszczególnych części jest bardzo dobra, a ceny większości z nich są konkurencyjne. Z tego też względu poniżej przedstawiono prototyp jednokolorowego wyświetlacza LED opartego na wybranym mikroprocesorze firmy Microchip (Atmel) [3], skupiając się głównie na doborze elementów bazowych, optymalnym podłączeniu poszczególnych układów oraz opracowaniu algorytmu sterującego wyświetlaniem. Z racji wcześniejszego doświadczenia [4], zdecydowano się na mikrokontroler z rodziny AVR.

2. Projekt wyświetlacza

Kluczowymi zagadnieniami przy poniższym projekcie są prostota budowy i niska cena podstawowych elementów konstrukcyjnych. Jednocześnie na tym etapie pominięto też kwestię obudowy, skupiając się głównie na elementach elektronicznych. Budowa wewnętrzna urządzenia wygląda podobnie jak w innych tego typu konstrukcjach. Wyświetlacz został podzielony na segmenty o wymiarach 8×8, gdzie każdy z nich jest obsługiwany przez osobny układ MAX7219. Pracą całego urządzenia steruje prosty 8-bitowy mikrokontroler AVR Atmega32. Poniżej przedstawiono bardziej szczegółowe dane dotyczące poszczególnych podzespołów wyświetlacza oraz organizacji połączeń i samego algorytmu wyświetlania.

2.1. Macierz LED

Cały wyświetlacz składa się z osobnych macierzy LED o wymiarach 8×8. Wymiar tych macierzy jest podyktowany wybranym sterownikiem MAX7219. Odpowiednio wymiary całego wyświetlacza są wielokrotnością wymiarów pojedynczej macierzy składowej (jednocześnie przekłada się to na rozdzielczość wyświetlacza). Do testów w ramach prototypu wybrano gotowe, scalone wyświetlacze z serii 1088AS, których schemat przedstawiono na rysunku 1. W układzie tym zastosowane są czerwone diody LED o średnicy 3 mm.



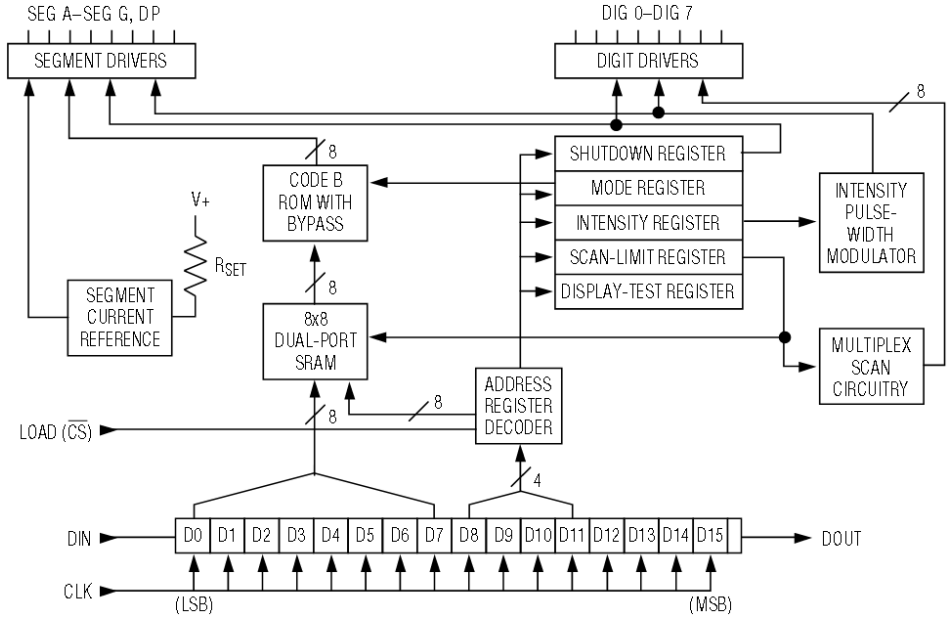
Rys. 1. Schemat modułu 1088AS

2.2. Sterownik macierzy LED – MAX7219

Do sterowania pojedynczymi macierzami LED zastosowano popularny układ MAX7219 [5]. Jego główna zaleta to komunikacja oparta na interfejsie SPI (ramka 16 bitowa) oraz możliwość szeregowego łączenia wielu takich układów. Układ jest jednocześnie dostosowany do obsługi ośmiu wyświetlaczy 7/8-segmentowych. Dodatkową zaletą jest, że sterownik nie musi być odświeżany cyklicznie (raz załadowane dane, są wyświetlane dopóki nie wgramy nowych danych). Jest również możliwość sterowania jasnością świecenia diod. Podstawowe parametry układu przedstawiono w tabeli 1. Poza zasilaniem, do kontroli i obsługi układu wystarczy wykorzystać trzy linie sterujące:

- LOAD(CS) – sygnał sterujący rejestrem przesuwnym, gdy stan jest niski dane szeregowo ładowane są do rejestru (dane podajemy od bitu najstarszego do najmłodszego), zbocze narastające blokuje (zamyka) rejestr i wprowadza dane do układu,
- CLK – sygnał zegarowy dla rejestru przesuwnego, dane wpisywane są przy zboczu narastającym,
- DIN – linia danych.

Na rysunku 2 przedstawiono funkcjonalny schemat blokowy sterownika [5].



Rys. 2. Funkcjonalny schemat blokowy układu MAX7219

Tabela 1. Podstawowe parametry układu MAX7219

Napięcie zasilania	4,0 V - 5,5 V
Interfejs szeregowy	SPI
Max częstotliwość interfejsu komunikacyjnego	10 MHz
Pobór prądu w trybie uśpienia	150 μA
Kontrola jasności diod LED	Analogowa (rezystor) i cyfrowa (instrukcje)
Sterowanie prądem diod LED	Zewnętrzny rezystor podciągający

2.3. Główny sterownik – Atmega32

Jako główny procesor sterujący pracą całego wyświetlacza wybrano mikrokontroler Atmega32. Podstawowe jego parametry zebrano w tabeli 2. Układ ten inicjalizuje wyświetlacze, generuje aktualny do wyświetlenia obraz oraz przesyła go w odpowiedni sposób do sterowników poszczególnych segmentów macierzy LED.

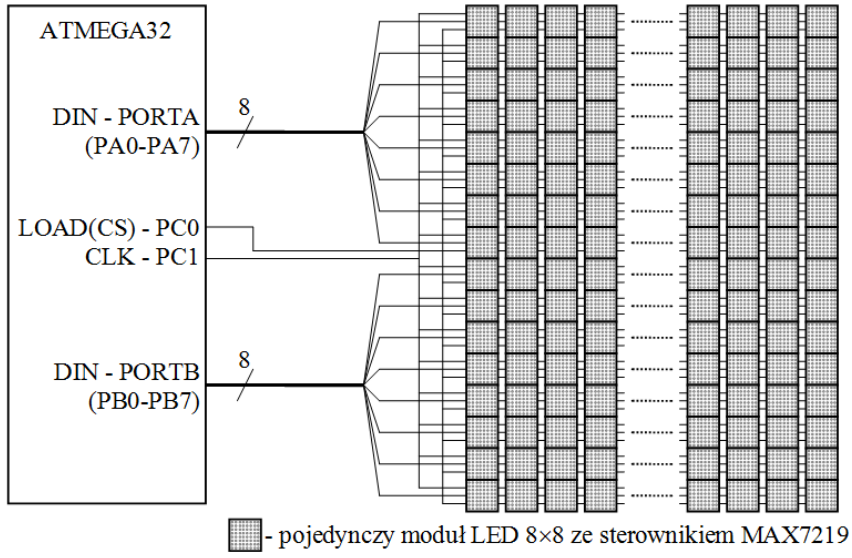
Tabela 2. Podstawowe parametry mikroprocesora Atmega32

Napięcie zasilania	4,5 V - 5,5 V (2,7 V – 5,5 V w wersji L)
Max częstotliwość taktowania	16 MHz (8 MHz w wersji L)
Pamięć programu	32KB (Flash)
Pamięć zmiennych	2048B (SRAM)
Pamięć EEPROM	1024B
Wbudowane magistrale komunikacyjne	UART, SPI, I2C
Ilość programowalnych linii I/O	32 (4 porty po 8 pinów)
Wbudowane peryferia	przetwornik AC, komparator, dwa liczniki 8-bitowe, jeden licznik 16-bitowy,

2.4. Zarządzanie wyświetlaczem

Wyświetlacz podzielony jest na macierze łączone wierszami, więc wszystkie sterowniki w danym wierszu mogą być obsługiwane ze wspólnych linii sterujących. Teoretycznie nie ma ograniczenia, co do ilości szeregowo połączonych układów, jednak płynność działania (odświeżania macierzy) będzie kluczowym elementem wpływającym właśnie na ilość sterowników w szeregu, a tym samym wymiary wyświetlacza. Jednocześnie warto tu pamiętać, że program przesyłający dane również potrzebuje określoną ilość cykli zegara na wykonanie dodatkowych instrukcji, co ma istotny wpływ na wydłużenie się czasu odświeżania. Bardzo istotne jest tu odpowiednie i przemyślane podłączenie do głównego procesora sterującego wyświetlaczem. Ponieważ wszystkie wiersze macierzowe wyświetlacza mogą być odświeżane w tym samym momencie, linie LOAD(CS) oraz CLK mogą być wspólne dla wszystkich sterowników macierzy 8×8. Jedynie linie danych muszą być oddzielne dla każdego wiersza macierzy wyświetlacza. Przykładowo wyświetlacz, o rozdzielczości 512x64 pixele (LED'y), składał by się z 512 macierzy

LED 8×8, ułożonych w 8 wierszach po 64 macierze w każdym. W takiej sytuacji potrzeba jedynie 8 linii danych DIN0-DIN7 oraz po jednej linii LOAD(CS) i CLK. Dodatkowo linie danych mogą być podłączone w procesorze głównym pod jeden port, co znacznie przyśpieszy ładowanie danych (załadowanie po jednym bicie na każdą linię można zrobić w jednym cyklu zegarowym – ładujemy cały bajt na raz, a każdy bit trafia na sterownik w innym wierszu). W ten sam sposób do obsługi np. 16 linii danych można wykorzystać dwa porty 8-bitowe, uzyskując wyświetlacz o wysokości 128 diod LED. Na rysunku 3 przedstawiono schemat połączeń pomiędzy poszczególnymi elementami składowymi oraz podział samego wyświetlacza.



Rys. 3. Schemat połączeń w wyświetlaczu

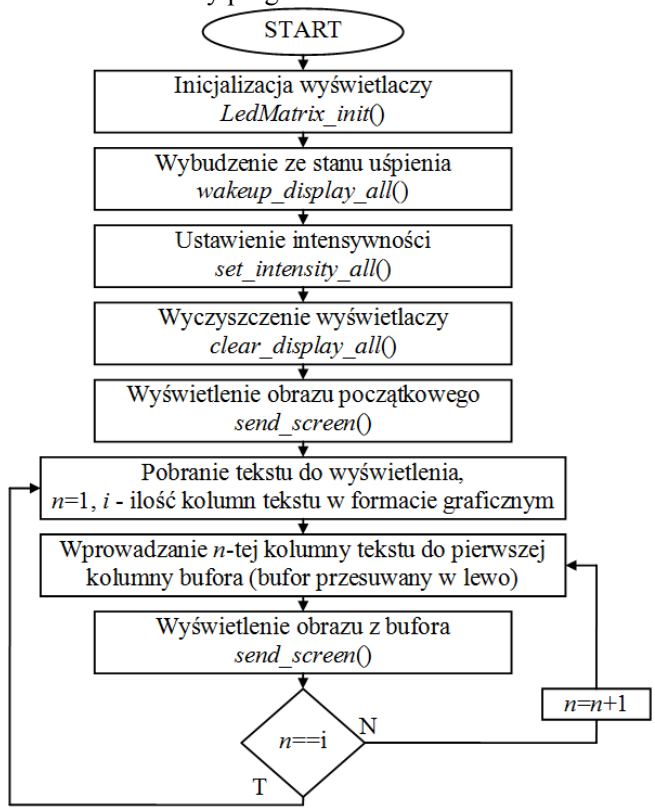
3. Program

Program został przygotowany typowo pod wybrany procesor uwzględniając architekturę 8-bitową oraz taktowanie wynoszące 16 MHz. Przedstawianie tutaj całego kodu nie ma większego sensu, więc poniżej opisano jedynie podstawowe funkcje:

- *LedMatrix_init*() – funkcja inicjalizacji wyświetlaczy,
- *wakeup_display_all*() – funkcja wybudzenia uśpionych wyświetlaczy,
- *set_intensity_all*() – funkcja ustawiająca zadaną jasność wyświetlaczy,
- *clear_display_all*() – funkcja czyszcząca wyświetlacze,

- *send_data_one*() – funkcja wysyłająca określone dane na jeden wskazany wyświetlacz (segment 8×8),
- *send_screen*() – funkcja wysyłająca bufor (obraz) na wyświetlacz,
- *disp_string_shift*() – funkcja do wprowadzania na wyświetlacz zadanego tekstu poprzez przesuwanie bufora (obrazu) w lewo,
- *shift_char16_screen*() – funkcja do wprowadzania na wyświetlacz danego znaku poprzez przesuwanie bufora (obrazu) w lewo,
- *shift_screen_left*() – funkcja przesuwająca bufor (obraz) o jedną kolumnę w lewo ustawiając zadany ciąg bitów.

Poza zadeklarowaniem odpowiednich funkcji i zmiennych (w tym bufor wyświetlanej ramki), konieczne było wprowadzenie w kodzie kilkudziesięciu tablic stałych, które przechowują czcionki dla wyświetlacza. Dzięki temu, przy wyświetlaniu (i przesuwaniu) nowego tekstu wystarczy wprowadzić dany tekst, bez potrzeby programowania całej ramki wyświetlacza (lub wielu ramek). Na rysunku 4 przedstawiono schemat blokowy programu.



Rys. 4. Schemat blokowy programu sterującego wyświetlaczem

4. Podsumowanie

Przedstawiony prototyp został złożony i uruchomiony z wykorzystaniem płytki rozwojowej z wybranym procesorem. Testy odbyły się dla wyświetlacza o wymiarach 32×92 pixele (LED'y). Wszystkie elementy układu działały prawidłowo. Głównymi ograniczeniami okazały się częstotliwość taktowania procesora oraz rozmiar pamięci (zarówno programu jak i zmiennych). Przyszłościowo planowany jest dalszy rozwój projektu, w tym przeniesienie głównego układu sterującego na procesor z rodziny ATxmega lub ARM (np. STM32), co pozwoli osiągnąć dużo lepsze parametry wyświetlacza (wymiary i częstotliwość odświeżania).

Bibliografia

1. RGB Technology [online], http://rgbtechnology.pl/ (data dostępu 12.04.2018r.)
2. GilBT [online], https://www.gilbt.com/ (data dostępu 12.04.2018r.)
3. 8-Bit MCUs Microchip Technology [online], https://www.microchip.com/design-centers/8-bit (data dostępu 12.04.2018r.)
4. Poczekajło P.: *Prototyp uniwersalnego modułu wielopunktowego pomiaru temperatury opartego na cyfrowych termometrach DS18B20*, Zeszyty Naukowe Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej, tom 11 (2017), str. 133-139. ISSN 1897-7421
5. MAX7219 DS [online], https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf (data dostępu 12.04.2018r.)

Streszczenie

W artykule przedstawiono propozycję realizacji jednokolorowego wyświetlacza LED. W projekcie wykorzystano mikrokontroler AVR Atmega32 oraz sterownik macierzy LED MAX7219. Wyświetlacz został podzielony na pojedyncze macierze o wymiarach 8×8. Zaprezentowana konstrukcja stanowi wstępną propozycję układu sterującego wyświetlaczem.

Abstract

The prototype of mono-color LED display is present in the paper. Microcontroller AVR Atmega32 and LED matrix drivers MAX7219 are used. The display consist of 8×8 LED matrixes. The construct is only proposition of the LED display controller.

Keywords: LED display, AVR, MAX7219, 8×8 LED matrix

Bohdan Andriyevsky
Chair of Electronics
Department of Electronics and Computer Sciences
Koszalin University of Technology, Poland

Jacek Piekarski
Chair of Water-and-Sewage Technology and Waste Utilization
Faculty of Civil Engineering, Environmental and Geodetic Sciences
Koszalin University of Technology, Poland

Lyudmyla Andriyevska
Chair of Water-and-Sewage Technology and Waste Utilization
Faculty of Civil Engineering, Environmental and Geodetic Sciences
Koszalin University of Technology, Poland

Fluctuations of kinetic energy at molecular dynamics and the atomic interactions in crystals

Key words: interatomic interactions, molecular dynamics, silicon, phonon relaxation time

1. Introduction

The molecular dynamics (MD) is a powerful method for the computer simulations of many physical and chemical properties of materials depending on temperature [1]. For example, the phenomena of thermal conductivity and structural phase transitions in solids may be studied using MD [2 - 5].

What is obtained from a molecular dynamics simulation, is the configuration or microstate of the system (the positions and momenta of every single atom) at any given time included in the simulation – a quantity that can not be measured experimentally. How to relate the microscopic configuration to macroscopic quantities that can be measured experimentally (observables), such as temperature (1) is the subject of statistical mechanics [6].

$$\left\langle E_{\mathrm{k}} \right\rangle = \frac{1}{2} f k_{\mathrm{B}} T \ , \tag{1}$$

here E_k is the kinetic energy and $f = 3N - 3$ is the degrees of freedoms for the system (crystal) with N atoms.

Statistical mechanics is concerned with statistical *ensembles*, an ensemble being a theoretical construct holding a large number of copies (sometimes infinitely many) of essentially the same system, that is; a collection of systems described by the same set of microscopic interactions, and sharing a common set of macroscopic control variables like internal energy E, volume V and number of atoms (or moles) N.[7]

From the known Maxwell-Boltzmann distribution for kinetic energies one can obtain the relations for the relative variance in single-particle kinetic energy $\varepsilon_k = 1/2 m_k v^2$,

$$\frac{\Delta\varepsilon_k^2}{\langle\varepsilon_k\rangle^2} = \frac{\langle\varepsilon_k^2\rangle - \langle\varepsilon_k\rangle^2}{\langle\varepsilon_k\rangle^2} = \frac{\langle v_k^4\rangle - \langle v_k^2\rangle^2}{\langle v_k^2\rangle^2} = \frac{2}{3}\ , \tag{2}$$

and the relative variance in instantaneous temperature T or N-particle kinetic energy E_k [8],

$$\frac{\Delta T^2}{\langle T\rangle^2} = \frac{\Delta E_k^2}{\langle E_k\rangle^2} = \frac{2}{f} = \frac{2}{3N-3}\ . \tag{3}$$

Here the values ΔT and ΔE_k are correspondingly the standard deviations of T and E_k.

For the case of absence the limitation of the degrees of freedom the value f should be equal to $3N - 3$ [8]. However interactions between atoms in solids may lead to the decrease of the effective value of f and consequently the value $\Delta E_k/\langle E_k\rangle$ (3). Such an interaction may be manifested in the form of correlation the positions and velocities of neighboring atoms, which may be temperature dependent. In the latter case one may expect, for example, the temperature dependent phonon relaxation time, phonon mean free path and as a result the coefficient of thermal conductivity.

The other example of the probable change the mentioned above value of f could be the temperature stimulated structural phase transition in solids. In this case, one may expect more or less large change with temperature of the f value near the temperature of phase transition of a crystal.

In the present study, the original approach based on the study of the temperature fluctuations of the crystal's kinetic energy E_k during MD simulation has been proposed and the results of the corresponding studies are analyzed.

2. Method of calculations

The equilibrium-type *ab initio* MD calculations of crystals have been performed in the framework of the density functional theory (DFT) using the VASP package

[9]. The projector augmented-wave (PAW) method with a cutoff energy of 400 eV for the plane waves was employed [9, 10] together with the corresponding pseudopotentials. For the exchange and correlation terms, the gradient corrected Perdew-Burke-Ernzerhof (PBE) functional was used. The MD calculations of silicon crystals were performed at the macro-canonical NVE ensemble for different temperatures at the optimized crystal structure of the super cell 3×3×3. Most results of MD calculations have been obtained for the simulation time up to 15 ps with the time steps of 1.5 fs. For the post MD analysis the nMoldyn 3.0 program was used [11]. In the case of $(C_3N_2H_5)_2SbF_5$ crystals, the MD calculations have been performed at the NVT ensemble using the smaller time steps of 0.5 fs, what is caused by the light hydrogen atoms.

3. Results and discussion

One of the values being calculated using the results of MD is the velocity autocorrelation functions (VACF) $C_{vv}(t)$,

$$C_{vv}(t) = \frac{1}{3N}\sum_{\alpha=1}^{N} w_\alpha \left\langle v_\alpha(0)\cdot v_\alpha(t)\right\rangle , \tag{4}$$

where, $\langle v_\alpha(0)\cdot v_\alpha(t)\rangle$ denotes the averaged value of the scalar products $v_\alpha(0)\cdot v_\alpha(t)$ for atom velocities for ion of the α-type, w_α is the weight coefficient, t the time and N the number of atoms in a supercell [11]. Due to the relatively strong bonding between atoms in solids the corresponding relatively strong correlation of atomic velocities is expected. But, on the other hand, due to the statistical character of atomic parameters and finite acoustic velocity in solids at finite temperatures the VACF is a decreasing function of a time. The latter property is manifested by the value of the VACF relaxation time τ. The value of τ may be compared with the mean phonon relaxation time τ_{ph}, which is responsible for the value of coefficients of thermal conductivity κ and thermal diffusivity D of materials in the kinetic theory of heat conductivity [12].

$$\kappa = \frac{1}{3}\rho C_V v^2 \tau_{ph} , \tag{5}$$

$$D = \frac{1}{3} v^2 \tau_{ph} . \tag{6}$$

where ρ is density of a material, C_V is the specific heat at constant volume V and v is the mean acoustic phonon velocity.

To verify the validity of our supposition, that due to the interatomic correlations the temperature dependence of the VACF relaxation time $\tau(T)$ and the relative temperature changes of the kinetic energy $\Delta E_k/\langle E_k\rangle(T)$ are dependent one from

another, we have performed the MD calculations for the silicon crystal at different temperatures in the range of 200 K - 1000 K. The MD calculations within the NVE ensemble were performed for the silicon supercell 3×3×3 containing 216 atoms. The fitted linear dependence with the coefficient of determination (COD) $R^2 \approx 0.94$ between the values τ and $\Delta E_k/\langle E_k\rangle$indicates for the relatively high degree of correlation between these values (Fig. 1).

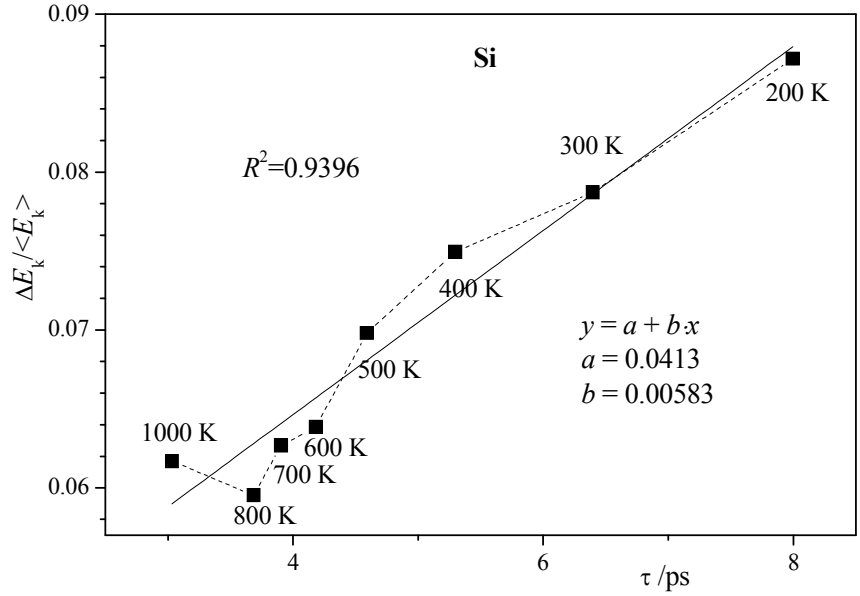


Fig. 1. Correspondence between the VACF relaxation time τ and the relative change of kinetic energy $\Delta E_k/\langle E_k\rangle$ of silicon obtained on the basis of MD calculations at different temperatures in the range of 200 K - 1000 K

The linear fit, $y = a + b \cdot x$, of the dependence between the values $\Delta E_k/\langle E_k\rangle$ and τ is characterized by the relatively high coefficient of determination $R^2 = 0.9296$ (Fig. 1), that is a clear proof for the validity of this relation. Here ΔE_k is the standard deviation of the kinetic energy E_k and $\langle E_k\rangle$ is the averaged value of kinetic energy. This relation could be useful for estimation the temperature dependent mean phonon relaxation time τ_{ph} and the coefficient of phonon thermal diffusivity D of silicon by means of calculation the value of $\Delta E_k/\langle E_k\rangle$. Of course, the fitting parameters a and b found (Fig. 1) are characteristic for silicon. Due to the straightforward calculation of the value $\Delta E_k/\langle E_k\rangle$ on the basis of the molecular dynamics run the proposed approach may be applied also for study of the coefficient of phonon thermal diffusivity D and thermal conductivity κ_{ph} as functions of different factors: chemical composition, temperature, pressure, etc.

Conclusions

1. The relative fluctuations of the silicon crystal kinetic energy $\Delta E_k/\langle E_k \rangle$ have been found to be in the inverse dependence with the temperature.
2. Clear correlation with the relatively high coefficient of determination $R^2 = 0.9296$ has been revealed between the fluctuations of the silicon crystal kinetic energy $\Delta E_k/\langle E_k \rangle$ and the corresponding relaxation time τ of the velocity autocorrelation function and the phonon relaxation time. This substantiate the use of the proposed approach for the calculation of values related to the heat conductivity in the silicon based semiconductors.

Acknowledgments

The calculations were performed in the computer centers ICM of Warsaw University (the project No. G26-3) and WCSS of Wrocław University of Technology (the project No. 053).

References

1. Marx D., Hutter J. 2009 *Ab initio molecular dynamics: basic theory and advanced methods*, Cambridge University Press
2. Stackhouse S., Stixrude L. 2010 *Reviews in Mineralogy & Geochemistry* **71** 253
3. Green M. S. 1954 *J. Chem. Phys.* **22** 398
4. Kubo R. 1957 *J. Phys. Soc. Jpn* **12** 570
5. Andriyevsky B., Czapla Z., Podsiadła D. 2018 *Mater. Chem. Phys.* **205** 452
6. Olsen Heggø D. M., *Ab initio molecular dynamics simulation of phosphorus diffusion in silicon*, MSc thesis, University of Oslo, 2012
7. Tuckerman M. E. 2010 *Statistical Mechanics: Theory and Molecular Simulations*, Oxford University Press
8. Frenkel D. 1996 *Understanding Molecular Simulation : from algorithms to applications*, Academic Press
9. Kresse G., Joubert D. 1999 *Phys. Rev. B* **59** 1758; Kresse G, Marsman M, Furthmüller J 2015 http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html, Vienna
10. Blöchl P. E. 1994 *Phys. Rev. B* **50** 17953
11. Rog T., Murzyn K., Hinsen K., Kneller G. R. 2003 *J. Comput. Chem.* **24** 657
12. Ziman J. M. 2001 *Electrons and Phonons*. Oxford University Press, Oxford

Abstract

The calculation method of the molecular dynamics has been applied to study the correlation of the kinetic energy fluctuations and the relaxation time of the velocity autocorrelation function and the phonon relaxation time in a crystal. On the basis of the molecular dynamics data for silicon crystal obtained at different temperatures in the range 200 K – 1000 K the correlation between the kinetic energy fluctuations and the relaxation time of the velocity autocorrelation function has been calculated with the relatively high coefficient of determination $R^2 = 0.9396$. The correlation obtained and the corresponding approach substantiate a use of the kinetic energy fluctuations for the calculation of values related to the heat conductivity in the silicon based semiconductors (coefficients of thermal conductivity and diffusivity).

Streszczenie

Obliczeniowa metoda dynamiki molekularnej została zastosowana do badania korelacji fluktuacji energii kinetycznej i czasu relaksacji autokorelacyjnej funkcji prędkości i czasu relaksacji fononów w krysztale. Na bazie danych dynamiki molekularnej kryształu krzemu otrzymanych w różnych temperaturach w zakresie 200 K – 1000 K została obliczona korelacja fluktuacji energii kinetycznej i czasu relaksacji autokorelacyjnej funkcji prędkości, która cechuje się stosunkowo wysokim współczynnikiem determinacji $R^2 = 0.9396$. Otrzymana korelacja uzasadnia zastosowanie fluktuacji energii kinetycznej do badań obliczeniowych wielkości powiązanych z przewodnością cieplną półprzewodników na bazie krzemu (współczynniki przewodności i dyfuzyjności cieplnej).

Słowa kluczowe: oddziaływania międzyatomowe, dynamika molekularna, krzem, czas relaksacji fononów

Grzegorz Górski
Mateusz Wojsa
Zakład Systemów Multimedialnych i Sztucznej Inteligencji
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. J.J. Śniadeckich 2
75-453 Koszalin

Blokowanie usług operatora sieciowego – przegląd wybranych ataków i metod ochrony

Słowa kluczowe: usługi internetowe, ataki sieciowe, blokowanie usług

1. Wprowadzenie

Większość powszechnie stosowanych protokołów sieciowych warstwy aplikacji powstała w czasach, kiedy kwestie związane z bezpieczeństwem transmisji danych nie miały dużego znaczenia. Nikt tak naprawdę nie spodziewał się tak szybkiego wykorzystania tych narzędzi przez cyberprzestępców. Dziś oczywiście nie ma możliwości zmiany fundamentów owych protokołów, przynajmniej jeżeli patrzymy na to krótkoterminowo.
Jednakże prócz problemów związanych z kompromitacją danych poufnych, protokoły te są praktycznie bezsilne na ataki, na jeden z fundamentów bezpieczeństwa informacji, czyli dostępność.

2. Rodzaje ataków

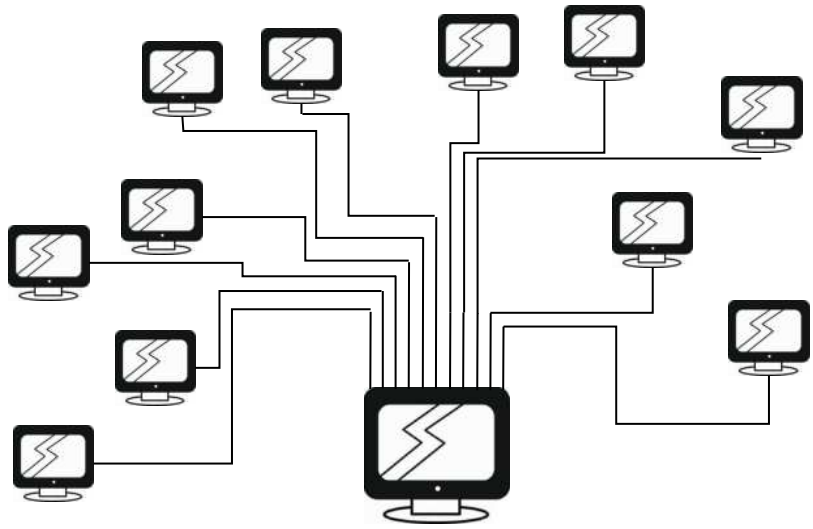
Atak a raczej cyberatak, jest to ciąg działań podjętych w celu zakłócenia, przejęcia danych bądź uzyskania kontroli nad danym systemem. Rodzajów ataków oraz ich wariacji jest wiele. Autorzy przedstawiają kilka podstawowych ataków oraz metody, które znacznie przeciwdziałają ich skutecznemu przeprowadzeniu.

2.1. Ataki których celem jest zablokowanie usług operatora

2.1.1. Botnet

Botnet jest to sieć zainfekowanych komputerów, która udostępnia hakerowi zdalną kontrolę nad zainfekowanymi maszynami. Dzięki kontroli nad setkami czy

nawet tysiącami maszyn, haker może rozsyłać różnego rodzaju kontent począwszy od spamu, a kończąc na wirusach, kraść dane osobowe, czy nawet przeprowadzać ataki DDoS. Botnety to jedno z największych współczesnych zagrożeń IT. Ich rosnąca popularność wśród cyberprzestępców wynika z ich zdolności do infiltracji niemal każdego urządzenia podłączonego do Internetu. Wykorzystuje się je nawet do „kopania" kryptowalut, które w ostatnich latach zyskały na popularności a cena niektórych wzrosła tysiące razy.



Aby lepiej zrozumieć działanie botnetów, należy wziąć pod uwagę, że sama nazwa to połączenie dwóch członów "robot" oraz "sieć". W szerokim tego słowa znaczeniu botnet jest to sieć robotów wykorzystywanych do popełniania cyberprzestępczości. Cyberprzestępcy je kontrolujący nazywani są pasterzami lub botmasterami.

Aby zbudować botnet, botmasterzy potrzebują jak najwięcej zainfekowanych urządzeń pod ich komendą. Im więcej botów jest podłączonych, tym większy botnet. Im większy botnet, tym większy wpływ i możliwa większa skala ataku. Rozmiar w tym przypadku ma znaczenie i to kolosalne. Ostatecznym celem przestępcy najczęściej jest zysk finansowy, propagowanie szkodliwego oprogramowania lub po prostu wprowadzenie ogólnego chaosu w Internecie.

Botnety zazwyczaj są tworzone nie tylko w celu złamania pojedynczego komputera. Charakteryzują się tym, iż zostały zaprojektowane do infekowania wielu urządzeń, liczonych nawet w milionach. Do zainfekowania komputera dochodzi najczęściej za pośrednictwem koni trojańskich. Strategia dostarczenia konia trojańskiego na komputer ofiary wymaga od użytkownika zainfekowania własnego systemu np. poprzez otwieranie załączników wiadomości e-mail ze złośliwym

oprogramowaniem, klikanie złośliwych wyskakujących reklam lub pobieranie niebezpiecznego oprogramowania. Gdy urządzenia są już zainfekowane, botnety mogą uzyskiwać dostęp i modyfikować dane, atakować inne komputery, czy też popełniać inne przestępstwa. Bardziej złożone botnety mogą nawet automatycznie się rozprzestrzeniać, wyszukiwać oraz infekować urządzenia.

Botnety są trudne do wykrycia. Ograniczają swoją pracę używając tylko niewielkiej ilości mocy obliczeniowej, by uniknąć zakłóceń normalnych funkcji urządzenia, co mogłoby do prowadzić do ostrzeżenia użytkownika. Bardziej zaawansowane botnety są zaprojektowane tak, aby aktualizować swoje zachowanie. Zachowanie to znacznie utrudnia wykrywanie ich przez oprogramowanie cyberbezpieczeństwa. Użytkownicy nie zdają sobie sprawy, że ich urządzenie jest w tym samym czasie kontrolowane przez botmastera. Usunięcie komputera z botnetu, wiąże się z usunięciem złośliwego oprogramowania, które kontroluje maszynę.

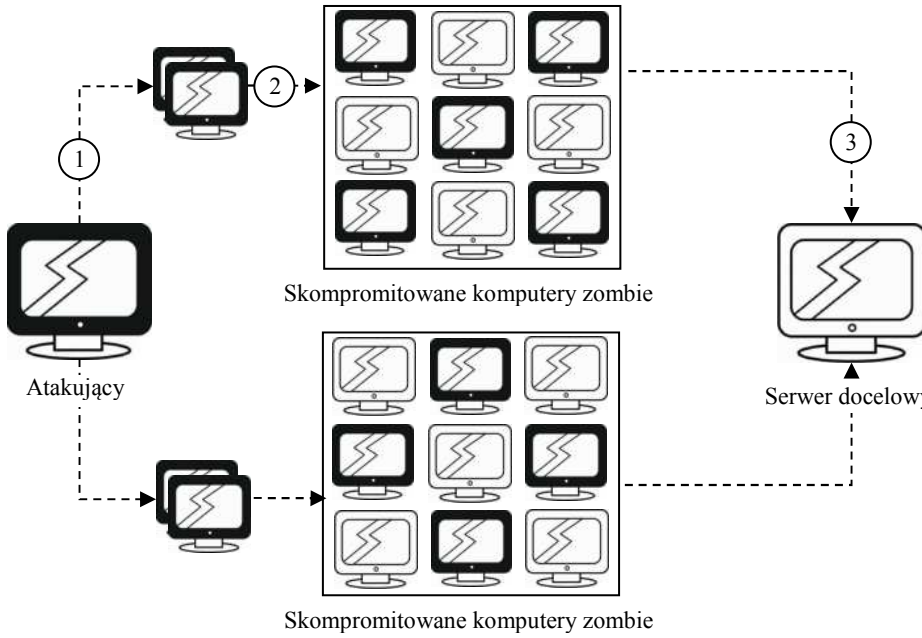
Przeprowadzenie ataku odmowy dostępu do usługi jest dość proste z wykorzystaniem Botnetu. Duża masa pozwala na generowanie dużej ilości żądań w sieci, a co za tym idzie zalanie nimi ofiary, przez co usługa usługodawcy zostaje zdestabilizowana.

2.1.2. DDoS

Jest atakiem na komputer lub sieć, którego celem jest zmniejszenie, ograniczenie bądź zablokowanie dostępu do zasobów systemowych użytkownikom. Atak ten bazuje na ataku DoS, którego celem jest destabilizacja pracy systemu, przeładowując jego zasoby za pomocą nieuzasadnionych żądań.

Istnieje wiele kategorii ataków:

- Volumetric Attack
- Fragmentation Attack
- TCP State-Exhaustion Attack
- Application Laer Attack



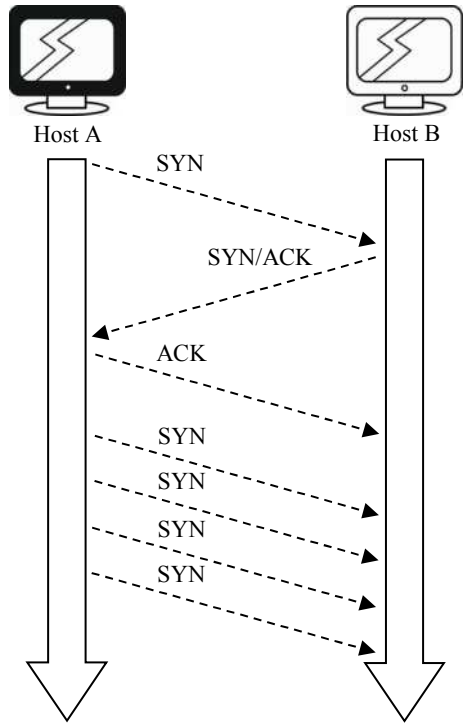
Przebieg ataku:

1. Atakujący instruuje kontrolery.
2. Kontrolery infekują dużą ilość komputerów.
3. Poinstruowane komputery zombie zalewają żądaniami serwer docelowy.

Koszt przeprowadzenia ataku DDoS jest stosunkowo niewielki, a co za tym idzie jest łatwy do zorganizowania. Stosowany jest najczęściej jako dywersja. Napływające żądania skutecznie potrafią odwrócić uwagę personelu IT, a tym samym znacznie podnoszą prawdopodobieństwo skutecznego przeprowadzenia właściwego ataku by mógł odnieść oczekiwany skutek.

2.1.3. SYN Flooding

Podczas gdy nowoczesne systemy operacyjne są lepiej przygotowane do zarządzania zasobami, co utrudnia przepełnienie tabel połączeń, serwery nadal są narażone na ataki typu SYN Flooding.



SYN Flooding bazuje na wadzie implementacji three-way handshake przez większość hostów. Kiedy host B odbiera żądanie SYN od hostu A, musi utrzymywać częściowo otwarte połączenie w liście „kolejki odsłuchowej" przez określony czas. Zasadniczo, sprawca wysyła żądania połączenia TCP szybciej niż maszyna docelowa może je przetworzyć. Kolejka odsłuchowa ofiary szybko się zapełnia, co powoduje nasycenie sieci. Zdolność zatrzymywania każdego niecałkowitego połączenia, może być wykorzystana do wykonania ataku DoS.

2.1.3.1. Przebieg ataku

Kiedy klient i serwer ustanawiają normalny "trójdrożny uścisk dłoni" TCP, wymiana wygląda następująco:

1. Klient żąda połączenia, wysyłając komunikat SYN (synchronizacja) do serwera. Najczęściej atakujący w celu anonimizacji swojego ruchu wykorzystuje fałszywy adres IP.
2. Serwer potwierdza, wysyłając wiadomość SYN-ACK (potwierdzenie synchronizacji) do klienta.
3. Klient odpowiada komunikatem ACK (potwierdzenie), a połączenie zostaje ustanowione.

W ataku powodzi SYN, atakujący wysyła powtarzające się pakiety SYN do każdego portu na docelowym serwerze. W tym czasie serwer nie może zamknąć połączenia, wysyłając pakiet RST, a połączenie pozostaje otwarte.

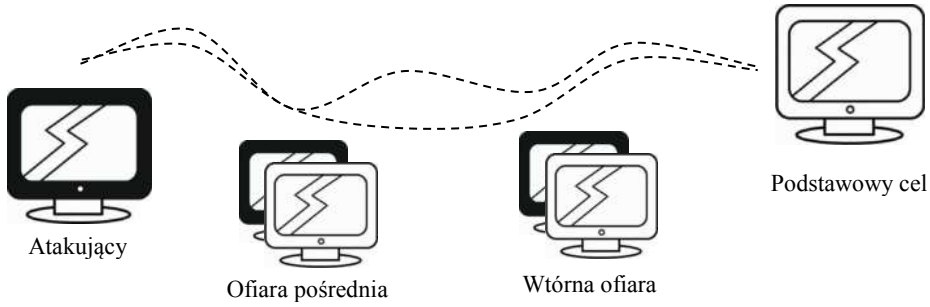
Zanim upłynie limit czasu połączenia Serwer, nieświadomy ataku, otrzymuje wiele pozornie uzasadnionych próśb o nawiązanie komunikacji. Odpowiada na każdą próbę pakietem SYN-ACK z każdego otwartego portu. Pozostawia to coraz więcej połączeń na wpół otwartych. Złośliwy klient albo nie wysyła oczekiwanego ACK, a atakowany serwer będzie czekał na potwierdzenie swojego pakietu SYN-ACK. W końcu, gdy wypełnią się tablice przepełnienia połączenia serwera, usługa dla legalnych klientów zostanie odrzucona, a serwer może nawet działać nieprawidłowo lub ulec awarii.

Pakiety SYN są często używane, ponieważ najmniej prawdopodobne jest, że zostaną odrzucone domyślnie.

2.1.4. DRDoS połączenie dwóch powyższych

DRDoS jest to odmiana ataku odmowy dostępu DoS. Powstała w wyniku połączenia metody zalewania żądaniami synchronizacji oraz metod używanych przy rozproszonych atakach odmowy dostępu DDoS.

Atak ten polega na generowaniu specjalnych pakietów SYN. Ich adres źródłowy jest oczywiście fałszywy ponieważ jest nim adres ofiary. Następnym krokiem jest wysłanie dużej ilości takich pakietów do sieci. Komputery, do których zostały zaadresowane, odpowiadają pakietami SYN/ACK. Pakiety SYN/ACK, są kierowane na adres pochodzący z fałszywego nagłówka. W wyniku czego ofiara jest zalewana olbrzymią liczbą pakietów, pochodzących z wielu hostów. W porównaniu do tradycyjnego ataku typu DDoS, utrudnia to wykrycie rzeczywistego źródła ataku.



Przebieg ataku:

1. Atakujący rozpoczyna atak wysyłając żądanie do ofiar pośrednich
2. Żądanie zostaje przekierowane do ofiar wtórnych
3. Cel zostaje zaatakowany przez ofiary wtórne

Protokół TCP został zaprojektowany w taki sposób, że przed transmisją danych zapewnia:

- że przed transmisją danych cel jest gotowy do odbioru informacji ("uzgadnianie")
- że po transmisji wszystkie dane zostały odebrane ("ACK").

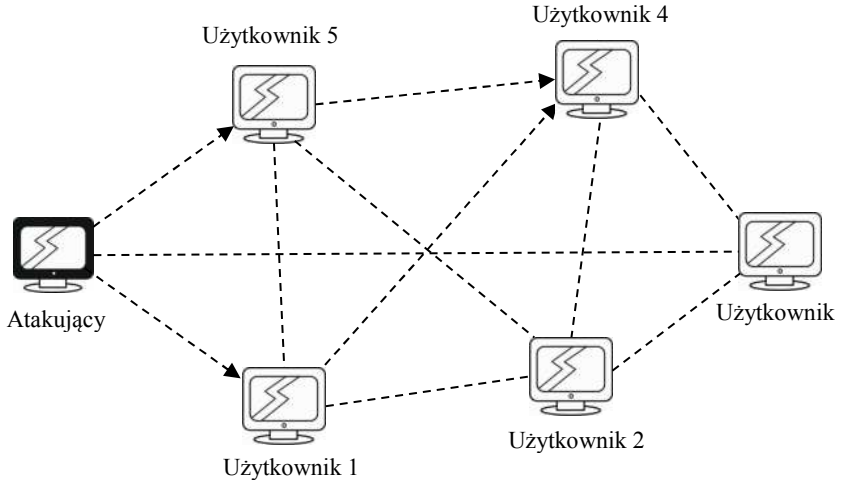
"Uścisk dłoni" można wypróbować kilka razy z rzędu. Dla żądania otwarcia sesji TCP ("SYN"), sfałszowany odbiorca pakietów otrzyma kilka prób otwarcia sesji. Im wyższy jest współczynnik wzmocnienia między rozmiarem minimalnego żądania a rozmiarem odpowiedzi, tym bardziej efektywny jest DRDoS.

Możliwe jest również wykonanie tego rodzaju wzmocnienia za pomocą protokołu UDP. Jego rolą jest umożliwienie transmisji danych między dwiema jednostkami. Jest jednym z głównych protokołów używanych w sieci Internet. Dzięki UDP atakujący może ukryć źródło ataku, poprzez użycie adresu IP osoby trzeciej w celu odbicia pakietów (fragmentacja przesyłanych danych).

Atakujący wysyła żądania odpowiedzi na różne serwery. Żądanie posiada zmodyfikowany adres IP celu, tzn. nie jest to adres własny atakującego, a adres atakowanego. Cel otrzymuje ogromne fale pakietów ze wszystkich serwerów, co skutkuje zdestabilizowaniem środowiska.

2.1.5. Peer to Peer atack

Sieci peer-to-peer różnią się do tradycyjnych sieci klient-serwer pod wieloma aspektami. Jednym z najważniejszych jest to, że każdy peer działa jako serwer i klient sieci. Innymi słowy w owych sieciach nie ma centralnego serwera służącego do przechowywania i udostępniania plików. Sieci peer-to-peer zawierają zdecentralizowane struktury ad hoc. Topologia sieci zmienia się co chwilę na wskutek możliwości losowego opuszczenia oraz dołączania do sieci przez uczestnika. Cechy te sprawiają, że jest ona podatna na ataki m.in. takie jak DoS.



Sieci P2P składają się z dużej liczby jednocześnie działających hostów. Tak więc jeden lub więcej złośliwych węzłów sieci, może z łatwością wykonywać DoS lub DDoS, czyli próbę zalania sieci fałszywymi pakietami, uniemożliwiając w ten sposób legalny ruch sieciowy. Inną metodą jest zalanie ofiary żądaniami kalkulacji w taki sposób, aby był na tyle zajęty by nie mógł odpowiedzieć na inne żądania. Ataki DoS są znacznie bardziej efektywne, jeśli w atak zaangażowanych jest wiele hostów (rozproszona odmowa usługi). W DDoS atakującymi komputerami są często komputery osobiste z dostępem do połączenia Internet, które zostały zainfekowane przez wirusa lub trojana. Sprawca może zdalnie nimi sterować oraz kierować atakiem na dowolny host. Atak DDoS można jeszcze bardziej wzmocnić stosując nieskompromitowane hosty jako wzmacniacze. Zombie wysyłają żądania do nieskompromitowanych hostów i podszywają swój adres IP adresem IP ofiary. Kiedy nieskompromitowani gospodarze odpowiadają, wyślą pakiety odpowiedzi do ofiary. Jest to tak zwany atak refleksyjny.

Sieci P2P do udostępniania plików nie są nowe. Ich wykorzystanie do dzielenia się wszystkimi mediami przez Internet, umożliwia „wybuch” w zapisach na stacjach roboczych. To była tylko kwestia czasu, kiedy cyberprzestępcy zaczęli wykorzystywać te "publiczne" sieci.

Wykrywanie ataku DoS P2P jest łatwe, lecz obrona przed nim jest trudna. Obrony obwodowe organizacji byłyby przytłoczone tak dużym atakiem. Blokowanie dużej liczby źródłowych adresów IP jest czasochłonne i spowolniłoby przetwarzanie pakietów do indeksowania. Jednym z rozwiązań jest zapobieganie w pierwszej kolejności docieraniu pakietów do sieci biznesowej.

2.1.6. Metody zapobiegawcze

Głównymi metodami zapobiegawczymi, są systemy detekcji bazujące na identyfikacji oraz dyskryminacji nielegalnego wzrostu ruchu z legalnego ruchu pakietowego. Systemy te kategoryzują atak jako zjawisko anormalne poprzez zbieranie statystyk ruchu sieciowego oraz badanie jego trendu.

Dobrą praktyką jednakże wymagającą wcześniejszego planowania jest budowa infrastruktury, w taki sposób by można było zarezerwować dodatkową nadmiarową pojemność, która byłaby w stanie wchłonąć atak poprzez przekierowanie części ruchu.

Należy również zidentyfikować krytyczne usługi i zatrzymać te, które nie są wykorzystywane. Należy mieć zawsze zainstalowane oraz zawsze aktualne aplikacje antywirusowe oraz antymalware'owe. Świadomość zespołu IT jest również jednym z ważniejszych elementów, dlatego należy dbać o szkolenia związane z bezpieczeństwem. Jeżeli atak został przeprowadzony skutecznie to ostatecznym krokiem jest zamknięcie wszystkich usług, w których wystąpił atak.

3. Podsumowanie

Niezależnie od rodzaju i wielkości przemysłu, firmy na całym świecie coraz częściej stają się celem ataków DDoS. Wyrafinowanie i intensywność tych ataków rośnie wykładniczo ze względu na wzrost w liczbie zaatakowanych systemów oraz niezałatanych luk w zabezpieczeniach.

Rozpoczęcie ataku DDoS jest trywialne w porównaniu do ilości czasu i zasobów poświęconych na stworzenie skutecznego środka zaradczego. Nowe techniki wykrywania i zwalczania tych ataków, są nieustannie tworzone, jednak tworzone są również nowe formy ataków, które powodują, że środki zaradcze stają się przestarzałe. Jest to problem, który aktualnie nie ma trwałego rozwiązania.

Bibliografia

1. Lei Xue, Xiaobo Ma , Xiapu Luo, Edmond W. W. Chan , Tony T. N. Miu, Guofei Gu, „Toward Detecting Target Link Flooding Attacks”, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, Volume 13, Issue 10, Pages 2423-2438, OCT 2018
2. Murat Semerci, Ali Taylan Cemgil, Bulent Sankur, „An intelligent cyber security system against DDoS attacks in SIP networks”, COMPUTER NETWORKS, Volume 136, Pages 137-154, MAY 8 2018

3. Ivica Dodig, Vlado Sruk, Davor Cafuta, „Reducing false rate packet recognition using Dual Counting Bloom Filter", TELECOMMUNICATION SYSTEMS, Volume 68, Issue 1, Pages 67-78, MAY 2018
4. Mutaz H. H. Khairi, Sharifah H. S. Ariffin, N. M. Abdul Latiff, A. S. Abdullah, M. K. Hassan, „A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN)", ENGINEERING TECHNOLOGY & APPLIED SCIENCE RESEARCH, Volume 8, Issue 2, Pages 2724-2730, APR 2018
5. Ademola P. Abidoye, Ibidun C. Obagbuwa, "DDoS attacks in WSNs: detection and countermeasures", ET WIRELESS SENSOR SYSTEMS, Volume 8, Issue 2, Pages 52-59, APR 2018
6. Yong-Joon Lee, Nam-Kyun Baik, Cheonshik Kim, Ching-Nung Yang, „Study of detection method for spoofed IP against DDoS attacks", PERSONAL AND UBIQUITOUS COMPUTING, Volume 22, Issue 1, Pages 35-44, Special Issue SI, FEB 2018
7. Silva, S., Silva, R., Pinto, R., Salles, R. M., „Botnets: A survey", Computer Networks, Volume 57, Issue 2, Pages 378-403, 4 February 2013
8. Materiały szkoleniowe EC-Council do egzaminu CEH v9 (Certified Ethical Hacker)

Streszczenie

Udany atak na usługę internetową zwykle kojarzy się z kompromitacją danych poufnych, jednakże nie musi tak być zawsze. Celem ataków z rodziny DoS nie jest uszkodzenie czy przechwycenie danych, a utrudnienie, bądź nawet uniemożliwienie dostępu do nich. Okazało się, że u usługodawców, mogą spowodować równie kosztowne straty.

Autorzy przedstawili przegląd wybranych ataków, których celem jest zablokowanie usług udostępnianych przez operatora oraz wybrane metody minimalizujące ich skutki.

Abstract

A successful attack on an internet service is usually associated with the embarrassment of confidential data, but it does not always have to be that way. The purpose of DoS attacks is not to damage or intercept data, but to hinder or even prevent access to them. It turned out that the service providers can also cause costly losses. The authors presented an overview of selected attacks that aim to block the services provided by the operator and selected methods minimizing their effects.

Keywords: internet services, network attacks, services blocking

Grzegorz Górski
Mateusz Wojsa
Zakład Systemów Multimedialnych i Sztucznej Inteligencji
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
ul. J.J. Śniadeckich 2
75-453 Koszalin

Wybrane ataki mające na celu kompromitację danych poufnych oraz metody zapewnienia bezpieczeństwa aplikacji i usług internetowych

Słowa kluczowe: usługi internetowe, ataki sieciowe, kompromitacja danych poufnych

1. Wprowadzenie

Zagrożenia związane z szeroko rozumianym IT dotyczą wszystkich. Ich skala oraz mnogość kierunków z których przychodzą jest duża, dlatego budowa systemu bezpieczeństwa organizacji musi przebiegać na wielu płaszczyznach. Przykładem takich zagrożeń może być m.in. dostęp osób nieuprawnionych, awaria lub utrata sprzętu, modyfikacja danych przez osoby nieuprawnione, czy nawet przypadkowe ich usunięcie bądź udostępnianie.

Wycieki danych zdarzają się niestety coraz częściej i są dużym problemem dla organizacji, zwłaszcza gdy ich skala jest ogromna. Niestety nigdy nie jesteśmy w stanie stwierdzić na ile i czy nasz system bezpieczeństwa jest szczelny, jednakże możemy skutecznie je ograniczać. Skuteczna ochrona danych poufnych organizacji wiąże się z zapewnieniem bezpieczeństwa na trzech poziomach:

- prawnym – zgodność z przepisami prawa, poziomu świadczenia usług (SLA), precyzyjnie określony zakres odpowiedzialności;
- ekonomicznym – stabilność, dostęp do zasobów, pewność oraz bezpieczeństwo transakcji;
- organizacyjnym: szkolenia, audyty, procedury, zabezpieczenia fizyczne oraz sieciowe.

2. Ataki mające na celu kompromitacje danych poufnych

2.1. SQL injection

SQL injection polega na tym, że atakujący próbuje wstrzyknąć swój kod SQL podczas wykonywania się w aplikacji innego zapytania, które nie ma odpowiedniego formatowania.

Przykładowe zapytanie SQL sprawdzające poświadczenia użytkownika:

SELECT 1 FROM USERS u WHERE u.login = ‘userLogin’ and u.password =’userPasword’

Gdzie userLogin, userPasword to place holdery, które podmieniane są na wartość z formularza.

W loginie użytkownika można wpisać np. jeżeli posiadamy użytkownika admin

admin’;--

W aplikacji nie posiadającej odpowiednich zabezpieczeń na tego typu atak, spowoduje to wygenerowanie zapytania:

SELECT 1 FROM USERS u WHERE u.login = ‘admin’;--' and u.password =’’;

Podobny efekt możemy uzyskać wpisując ‘ or 1=1;-- w pole hasło gdyby pole login było zabezpieczone ale znamy login użytkownika:

SELECT 1 FROM USERS u WHERE u.login = ‘admin’ and u.password =’’ or 1=1;--';

Takie zapytania zwrócą 1 więc framework zostanie poinformowany o tym, że w bazie zapisane są poprawne poświadczenia i dany użytkownik zostanie zalogowany bez hasła.

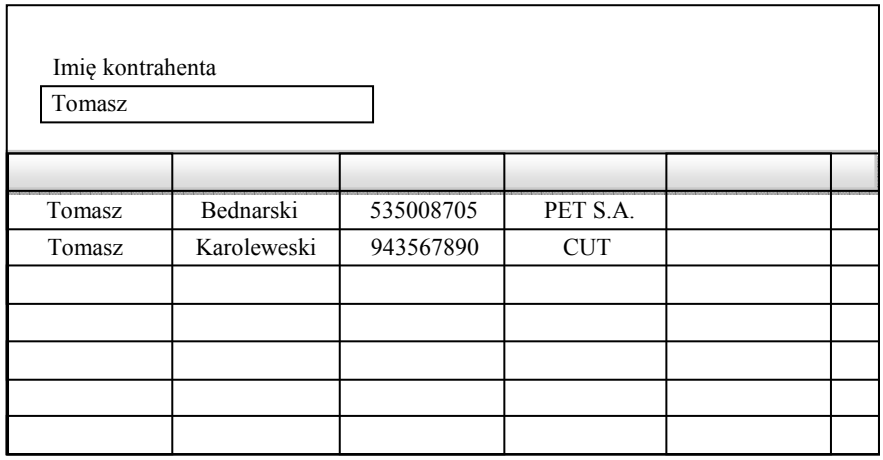
Rozróżniamy trzy główne podatności:

- zwykłe
- ślepe
- poprzez komunikat błędu.

2.1.1. Zwykłe

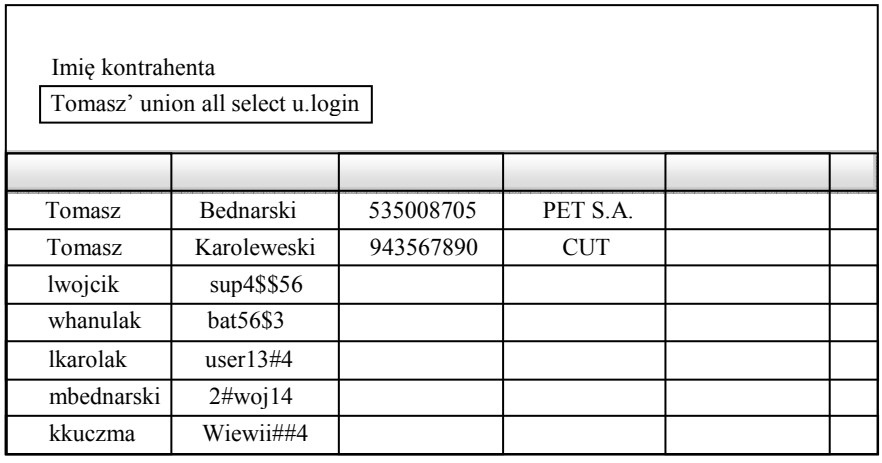
Zwykłe SQL injection to podatność formularzy wyświetlających dane na stronie. Wykorzystując przykładowo okno filtrowania możemy wstrzyknąć poprzez zapytanie dane z innej tabeli np. poprzez zastosowanie złączenia („UNION”). Należy jednak pamiętać, że wstrzyknięta kwerenda powinna posiadać tyle samo kolumn ile wyświetlanych jest w formularzu. Ponadto dane powinny być tego samego typu by silnik bazodanowy poprawnie utworzył tabelę tymczasową z wyłuskanych danych. Najczęściej robi się to drogą dedukcji.

Jeśli aplikacja umożliwia wykonywanie wielokrotnych zapytań albo gdy mamy hasło do bazy danych atakujący może dodać do bazy dowolne dane np. jakiś złośliwy kod, gdy aplikacja jest podatna na XSS (nie sprawdza danych pochodzących z bazy).



Tomasz	Bednarski	535008705	PET S.A.		
Tomasz	Karoleweski	943567890	CUT		

Wpisując np. ‘ union all select u.login, u.password, null, null, null;--



Tomasz	Bednarski	535008705	PET S.A.		
Tomasz	Karoleweski	943567890	CUT		
lwojcik	sup4$$56				
whanulak	bat56$3				
lkarolak	user13#4				
mbednarski	2#woj14				
kkuczma	Wiewii##4				

2.1.2. Ślepe

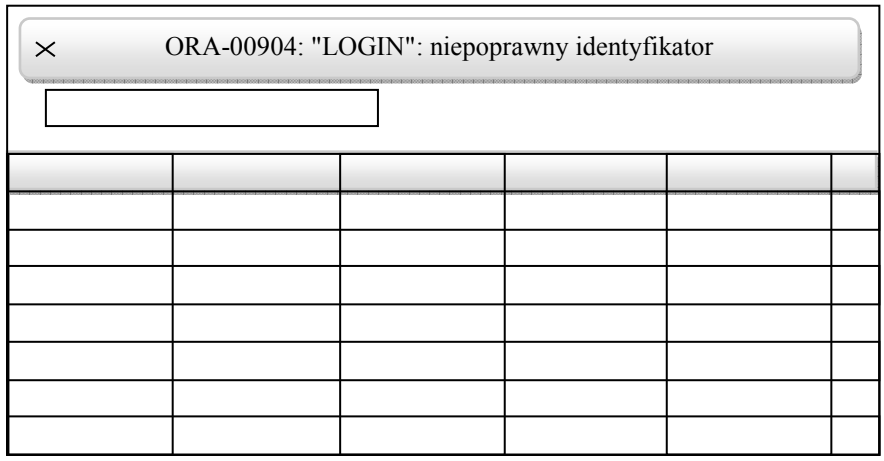
Najczęściej o blind injection mówi się wtedy, kiedy nie zwracane są żadne dane. Pomimo to aplikacja nadal jest narażona na tego typu ataki. Do zapytania można dodać np. próbę zgadnięcia liter w haśle np. przy użyciu metody substring oraz instrukcji case. Jeżeli warunek jest spełniony dodaje się opóźnienie. Natomiast

w przypadku gdy litera nie została odnaleziona można pominąć opóźnienie albo dodać inną jego długość celem upewnienia się, że zapytanie wykonało się poprawnie.

```
UNION SELECT IF(SUBSTRING(user_password,1,1) =
CHAR(50),BENCHMARK(5000000,ENCODE('MSG','by 5 seconds')),null) FROM
users WHERE user_id = 1;
```

2.1.3. Poprzez komunikat błędu

W przypadku gdy aplikacja zwraca błędy bazy danych, można wykorzystać je przy atakach typy SQL Injection. Przykładem tego może być baza danych ORACLE, z której można odczytać nazwę kolumny, co może znacznie przyspieszyć atak.



2.2. Jak się zabezpieczyć

Aby zabezpieczyć się przed tego typu błędami, jeśli dane wejściowe od użytkownika muszą być użyte w zapytaniu SQL, trzeba tak jak w przypadku XSS odpowiednio oznaczyć aby nie były traktowane jako tzn. znaki specjalne. Najlepiej skorzystać z zapytań preparowanych. Polega to na tym, że zanim wykonamy zapytanie, przekazujemy do niego kod SQL z miejscami na wstawienie właściwych parametrów. Następnie przy wywołaniu kwerendy przekazujemy, do tak przygotowanego zapytania dane, które zostaną odpowiednio sformatowane przez bibliotekę bazy danych, w zależności od typu.

```
Select u.* from users u where u.login = :userLogin;
```

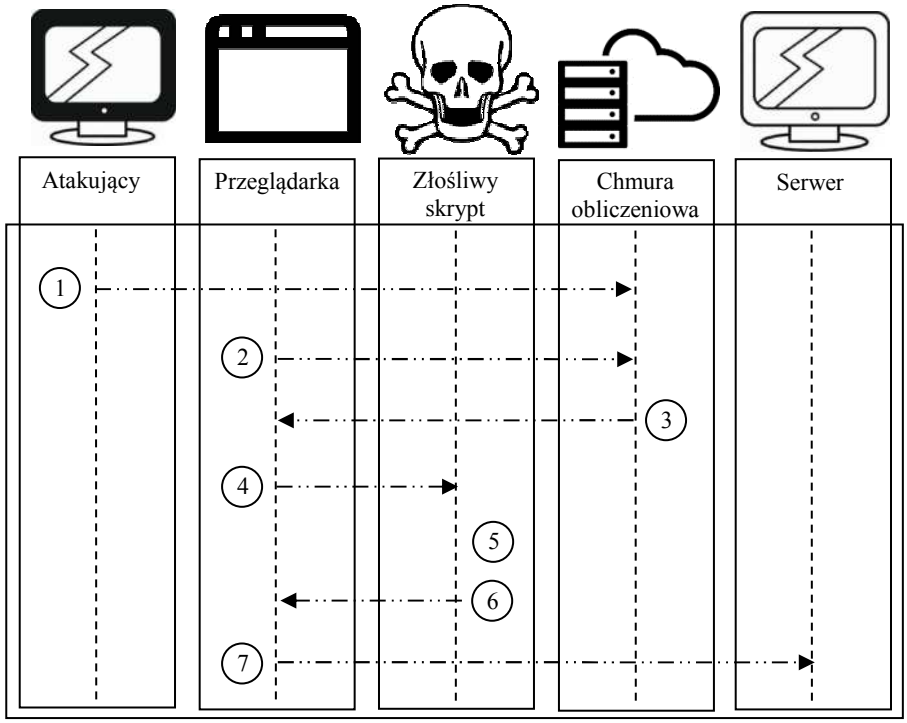
Jeśli biblioteka do bazy danych zwraca wyjątki należy je przechwytywać oraz wyświetlać błąd przyjazny dla użytkownika. Prawdziwy błąd SQL nie powinien być nigdy zwracany do interfejsu użytkownika, zaś pliki logów powinny być zabezpieczone w należyty sposób przed niepowołanym dostępem.

2.3. Session Hijacking

Podatność ta polega na uzyskaniu dostępu do sieci za pomocą luk bezpieczeństwa w systemie. Atakujący próbuje uzyskać dostęp do istniejącej sesji użytkownika, czyli do takiej, której identyfikator został przydzielony już wcześniej. Po uzyskaniu dostępu do sesji użytkownika, atakujący może przejąć część komunikacji klient-serwer symulując jej uczestnika.

Przejęcie sesji ma miejsce, gdy po pomyślnym uwierzytelnieniu logowania, token sesji zostaje wysyłany do przeglądarki klienta z serwera WWW. Atak typu "przejęcie sesji" działa, gdy narusza token, konfiskując go lub zgadując, jaka będzie autentyczna sesja tokenu, uzyskując w ten sposób nieautoryzowany dostęp do serwera sieci Web. Może to doprowadzić do sniffingu sesji, ataków man in the middle czy man in the browser, trojanów, a nawet implementacji złośliwych kodów JavaScript.

Deweloperzy stron internetowych są szczególnie ostrożni podczas przechwytywania sesji, ponieważ pliki cookie HTTP, które służą do podtrzymywania sesji witryny, mogą zostać zaatakowane przez atakującego.



Przebieg ataku:

1. Atakujący hostuje stronę zawierającą złośliwy skrypt
2. Ofiara odwiedza stronę atakującego
3. Przeglądarka pobiera kod HTML wraz ze złośliwym skryptem
4. Następuje uruchomienie skryptu
5. Złośliwy skrypt przechwytuje pliki cookie
6. Nastepuje przekierowanie na serwer atakującego
7. Pliki cookies przekazane zostają do serwera, którego ma nastąpić kompromitacja.

2.3.1. XSS

Cross Site Scripting polega na wstrzykiwaniu złośliwego kodu w oryginalną treść strony, przeważnie poprzez wszelkiego rodzaju formularze internetowe, gdzie zatwierdzona treść jest potem wyświetlana.

Istota podatności XSS to przede wszystkim atak na klienta korzystającego z podatnej web-aplikacji. Przykładem wstrzyknięcia do przeglądarki ofiary fragmentu języka skryptowego, który może być uruchomiony w przeglądarce, jest poniższy kod w języku JS.

```
<script>alert(document.cookie());</script>
```

Skrypt wyciąga plik cookie i wyświetla go w oknie alertu. W efekcie, atakujący ma możliwość wykonania dowolnego kodu skryptowego w przeglądarce.

Warto tu również wspomnieć o tym czym może skutkować wykonanie javascriptowego kodu w przeglądarce ofiary:

- wykradanie cookies sesyjnych czyli de facto przejęcie zalogowanej sesji ofiary,
- dynamiczna podmiana zawartości strony www,
- uruchomienie keyloggera w przeglądarce,
- hostowanie malware-u z wykorzystaniem zaatakowanej aplikacji.

Istnieje wiele narzędzi, które w praktyce pokazują różne możliwe negatywne efekty wykorzystania XSS.

Błędy XSS dzielimy na trzy kategorie:

- Persistent/stored XSS – najbardziej złowroga odmiana, polegająca na umieszczeniu kodu JavaScript po stronie serwerowej.
- Reflected XSS – w tym przypadku kod JavaScript zaszyty jest w linku, który atakujący przesyła do ofiary. Ofiara po kliknięciu na linka łączy się z aplikacją, przekazując jej nieświadomie fragment HTML zawierający kod wykonujący JavaScript. Aplikacja zwraca ofierze HTML zawierający wcześniej podany JavaScript, powodując wykonanie kodu w przeglądarce.
- DOM Based XSS – jest atakiem, w którym atak jest wykonywany poprzez modyfikację drzewa DOM w przeglądarce ofiary, w taki sposób, aby kod po stronie klienta działał w sposób "nieoczekiwany". Oznacza to, że sama odpowiedź HTTP się nie zmienia, ale kod strony klienta jest wykonywany inaczej, ze względu na złośliwe modyfikacje, które wystąpiły w środowisku DOM.

XSS jest jedną z najbardziej rozpowszechnionych luk w aplikacjach internetowych. Występuje najczęściej w aplikacjach internetowych, które korzystają z niezatwierdzonych lub niekodowanych danych wejściowych użytkownika w generowanym przez siebie wyjściu.

Wykorzystując XSS, atakujący nie atakuje bezpośrednio ofiary. Zamiast tego osoba atakująca wykorzysta lukę w witrynie lub aplikacji internetowej odwiedzanej przez ofiarę, wykorzystując w naturalny sposób podatną witrynę internetową do dostarczania szkodliwego skryptu do przeglądarki ofiary.

Przykład działania może zostać opisany na popularnym języku JavaScript. Złośliwy JavaScript ma dostęp do wszystkich tych samych obiektów, co reszta strony, w tym do plików cookie. Pliki cookie są często używane do przechowywania

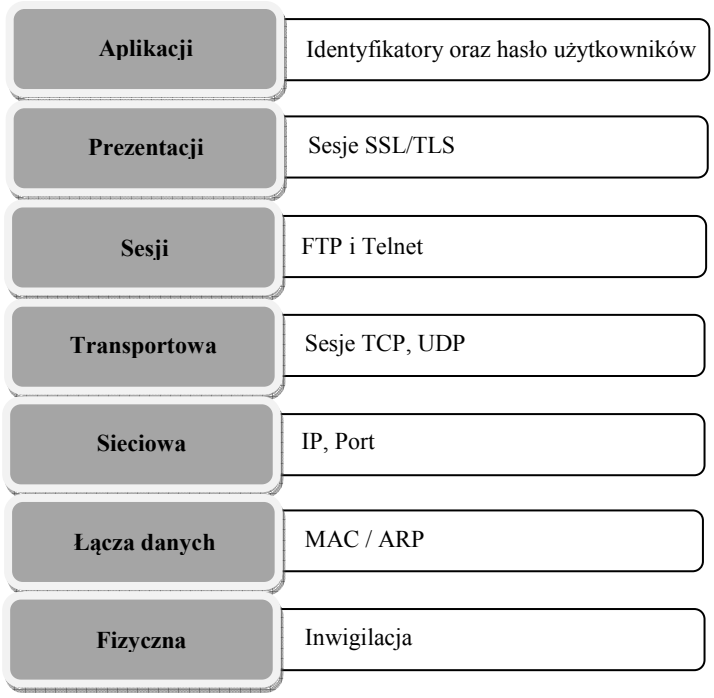
tokenów sesji, jeśli atakujący może uzyskać plik cookie sesji użytkownika, może podszyć się pod tego użytkownika. JavaScript ma dostęp do wielu funkcji przeglądarek takich jak czytanie i dowolna modyfikacja DOM, wykorzystywanie nagłówka XMLHttpRequest do wysyłania żądań HTTP z dowolną treścią do dowolnych miejsc docelowych, wykorzystanie interfejsów API HTML5, umożliwiających dostęp do geolokacji, kamery internetowej, mikrofonu, a nawet określonych plików z systemu plików użytkownika. Podczas gdy większość tych interfejsów API wymaga zgody użytkownika, XSS w połączeniu z pewną sprytną inżynierią społeczną może skutecznie umożliwić atakującemu ich wykorzystanie.
Powyższe, w połączeniu z inżynierią społeczną, pozwala atakującym na podejmowanie zaawansowanych ataków, takich jak kradzież plików cookie, keylogging, phishing (kradzież tożsamości).

2.3.2. Network Sniffing

Network Sniffing koncentruje się na znacznie szerszym podejściu do hakowania, czyli podsłuchiwaniem sieci. Technika ta jest często wykorzystywana przez hakerów. Istnieje wiele darmowych snifferów, które między innymi poprzez nasłuchiwanie pakietów w sieci potrafią w dość szybki sposób złamać klucz WPA. Atak ten obejmuje przechwytywanie, dekodowanie, sprawdzanie oraz interpretowanie informacji zawartych w pakietach sieciowych. Celem ataku jest zwykle kradzież identyfikatorów użytkowników, haseł czy nawet numerów kart kredytowych. Sniffing ze względu na przebieg ataku określa się go jako typ pasywny gdzie atakujący jest niewidzialny w sieci. Utrudnia to znacznie wykrycie atakującego w sieci, dlatego jest to jeden z najniebezpieczniejszych rodzajów ataków.

Narzędzia do nasłuchiwania sieci zostały stworzone z bardziej etycznych pobudek i były wykorzystywane wyłącznie przez profesjonalnych inżynierów sieciowych. Etyczne ich wykorzystanie m.in. przez ludzi potocznie zwanych white hat, którzy mogą dostarczyć organizacji informacji z analizy przechwyconych pakietów, analizy ruchu sieciowego, ewentualnych problemów sieciowych, jednakże cyberprzestępcy wykorzystują je w celach nieetycznych takich jak, wykradanie identyfikatorów i haseł użytkowników sieci, kradzież danych wiadomości błyskawicznych czy e-mail, celem kompromitacji organizacji.

Aby zrozumieć, dlaczego hackerzy „węszą”, musimy wiedzieć, co mogą uzyskać z sieci. Poniższy rysunek pokazuje warstwy OSI oraz informacje, które haker może wyłuskać w każdej warstwie.



Jak można zauważyć haker może zaatakować aż na 7 warstwach modelu OSI, dlatego jasne i szczelne procedury bezpieczeństwa organizacji są szczególnie ważne by skutecznie się przed nim chronić.

2.3.3. Social Engineering

Inżynieria społeczna jest w istocie sztuką uzyskiwania dostępu do budynków, systemów lub danych poprzez wykorzystanie ludzkiej psychiki, a nie poprzez włamywanie się lub używanie technik hakerskich. Na przykład, zamiast próbować znaleźć lukę w oprogramowaniu, inżynier socjalny może zadzwonić do pracownika i przedstawić się jako osoba wspierająca IT, próbując oszukać pracownika w celu ujawnienia jego hasła.

Przeprowadzenie skutecznej inżynierii społecznej zależy od tego, czy ludzie są świadomi swoich cennych informacji i czy zwracają szczególną uwagę na ich ochronę.

Ludzka natura, czyli ufność jest to podstawa każdego ataku inżynierii społecznej.

Ignorancja na temat inżynierii społecznej i jej wpływ na siłę roboczą sprawia, że organizacja jest łatwym celem. Inżynierowie społeczni by ujawnić informacje opierają się na ludzkiej chciwości np. obiecując coś za nic, bądź poczuciu

moralności np. prosząc o pomoc, a atakowani zgodnie z poczuciem moralnego obowiązku ujawniają informacje w ramach wyższego celu.

2.4. Spoofing

Jest to atak, którego ideą jest przyjmowanie fałszywej tożsamości celem uzyskania nieuprawnionego dostępu do systemu i usług. Ponieważ podszywanie dotyczy wszystkich warstw systemu OSI, bardzo trudno się przed nim bronić. Wyróżnia się dwie główne odmiany spoofingu:

- Blind spoofing – polega na przesyłaniu danych uwierzytelniających, przy braku dostępu do informacji, aby określić parametry sieci, śledzić zmiany ustawień lub, w szczególnym przypadku, uzyskać połączenie.
- Active spoofing – monitorowanie, generowanie, uszkadzanie bądź usuwanie pakietów wysyłanych w trakcie komunikacji, między serwerem a uwierzytelnionym użytkownikiem, w celu uzyskania uprawnień.

Jedyną obroną przed spoofingiem są ścisłe reguły filtrowania pakietów, stosowanie zabezpieczeń kryptograficznych np. SSH, czy PGP oraz walka z lekkomyślnością użytkownika, który przez nieuwagę, czy też niewiedzę zezwala na połączenie. Często użytkownicy sami pobierają programy umożliwiające wykradzenie klucza, zwłaszcza gdy łatwowiernie korzysta się z automatycznych aktualizacji programów, ponieważ nawet zaufane witryny często padają ofiarami podszywania. Pobieranie plików i programów z nieznanych źródeł bez odpowiednich poświadczeń to następna otwarta furtka dla cyberprzestępców.

Większość programów malware, działa w tle na poziomie ukrytym, wysyłając informacje przez Internet, lub nasłuchując na określonych portach poleceń od

hackera. Instalowanie poprawek, nie tylko systemowych, sprzyja bezpieczeństwu i wydajności systemu. Programy takie jak konie trojańskie rezydują w systemie, uruchamiając się dzięki wpisom do kluczy rejestru. Często też tworzą zapisy w plikach uruchomieniowych, czyli zarządzających każdym startem systemu. Należy regularnie przeglądać listy działających programów, pliki rozruchowe, ustawienia kluczy autostartu oraz dzienniki zdarzeń. Badanie wydajności systemu, pozwala na wykrycie nieprawidłowości w wykorzystaniu zasobów komputera i odnalezienie programów, użytkowników i usług mogących negatywnie wpływać na system. Uwagę powinny zwracać zwłaszcza nietypowe parametry ruchu sieciowego, takie jak błędy w transmisji, czy liczba odrzuconych pakietów. Dzięki analizie wykorzystania połączenia sieciowego komputera, można odnaleźć szkodliwe programy łączące się z siecią, komunikujące się z nieznanymi lokacjami, lub powtarzające się w określonym czasie, żądania nawiązania połączenia z różnymi usługami w sieci.

2.5. Metody zapobiegawcze

By skutecznie chronić się przed wyżej wymienionymi atakami należy przede wszystkim używać generatora liczb losowych, które generują silnie kryptograficzne liczby losowe, zapewniające odpowiednio mocne klucze sesyjne. Nie należy używać skompromitowanych algorytmów szyfrujących. Dobrą praktyką jest wyłączanie wszystkich niepotrzebnych usług systemowych. W swoich aplikacjach blokuj wczytywania stron w ramkach (iFrame). Użytkownicy powinni dysponować wydajnym programem antywirusowym, zabezpieczającym przed złośliwym oprogramowaniem oraz powinni je jak najczęściej aktualizować.

Aby zapobiec atakom typu XSS, należy filtrować dane od użytkownika oraz bazy danych lub je odpowiednio formatować, jeśli są wstawiane w kontent naszej strony.

Jeśli przechowujemy wrażliwe dane w aplikacji, powinniśmy udostępniać naszą aplikację poprzez protokół https. W przeciwnym wypadku będą możliwe ataki typu Man in the middle na naszą aplikację.

Identyfikator sesji możemy przekazywać zarówno w adresie URL, jak i w ciasteczku. Pierwszy z wymienionych sposobów zdecydowanie nie jest dobrym pomysłem, ponieważ:

- Każda osoba używająca tego samego komputera będzie mogła poznać identyfikator sesji z historii stron przeglądarki.
- Adresy URL często zapisywane są przez serwery proxy, dzienniki zdarzeń, itp. W wyniku ich analizy można przeczytać identyfikatory sesji.
- Kiedy użytkownik przechodzi na inną stronę, adres URL wraz z identyfikatorem sesji dostępny jest w nagłówku HTTP Referer.

- Użytkownik, przekazując adres URL innym osobom, nieumyślnie udostępnia także identyfikator sesji.
- Naturalnym wyborem do przekazywania identyfikatora sesji są więc ciasteczka.
- W konfiguracji powinniśmy też ustawić odpowiednie parametry dotyczące samego ciasteczka sesji (lifetime, path, secure, httponly).
- Identyfikator sesji powinien być odpowiednio długi i losowy, trudny do odgadnięcia oraz trudny do odtworzenia.

Sesje, które nie wygasają w odpowiednio krótkim odstępie czasu, dają atakującemu o wiele więcej możliwości na atak. Mechanizm obsługi sesji musi w pełni kontrolować czas, po jakim sesja traci ważność albo jest usuwana z magazynu przechowującego. Sesja powinna być przerwana, gdy nie stwierdzono żadnej aktywności przez pewien okres czasu, np. 30 minut, gdy nastąpi błąd bezpieczeństwa, lub gdy użytkownik sam przerwie sesję poprzez wylogowanie z systemu.

3. Podsumowanie

Pomimo tego. że wiele organizacji boryka się z cyberprzestępczością, wprowadzanie oraz przestrzeganie odpowiednich procedur bezpieczeństwa jest rzadkością.

Firmy powinny korzystać zawsze z najnowszego oprogramowania, a zespół IT powinien pilnować by były zawsze aktualne. Organizacje powinny korzystać z technologii pozwalających na wykrywanie szkodliwego oprogramowania, a również uniemożliwić pracownikom instalację oprogramowania z niezaufanych źródeł. Kopie bezpieczeństwa powinny być tworzone stosunkowo często, ponieważ w przypadku utraty danych bądź dostępu skutkiem ataku będzie można w stosunkowo prosty i bezbolesny sposób przywrócić utracony dostęp.

Często niedoceniane są wszelkie audyty bezpieczeństwa, które dają jasną ocenę aktualnego bezpieczeństwa sieci oraz pozwalają na identyfikację błędów w zabezpieczeniach.

Bibliografia

1. Francois Mouton, Alastair Nottingham, Louise Leenen, H. S. Venter, „FINITE STATE MACHINE FOR THE SOCIAL ENGINEERING ATTACK DETECTION MODEL: SEADM", SAIEE AFRICA RESEARCH JOURNAL, Volume 109, Issue 2, Pages 133-147, JUN 2018
2. Zhenjun Zhang, Xingqun Zhan, „Statistical Analysis of Spoofing Detection based on TDOA", IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING,nVolume 13, Issue 6, Pages 840-850, JUN 2018
3. Hossen Mustafa, Wenyuan Xu, Ahmad-Reza Sadeghi, Steffen Schulz, „End-to-End Detection of Caller ID Spoofing Attacks", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, Volume 15, Issue 3, Pages 423-436, MAY-JUN 2018
4. Karis D'silva, J. Vanajakshi, K. N. Manjunath, Srikanth Prabhu, „An Effective Method for Preventing SQL Injection Attack and Session Hijacking", 2017 2ND IEEE INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ELECTRONICS, INFORMATION & COMMUNICATION TECHNOLOGY (RTEICT), Pages 697-701, 2017
5. Bharti Nagpal, Naresh Chauhan, Nanhay Singh, „A Survey on the Detection of SQL Injection Attacks and Their Countermeasures", JOURNAL OF INFORMATION PROCESSING SYSTEMS, Volume 13, Issue 4, Pages 689-702, AUG 2017
6. Shashank Gupta, B. B. Gupta, Pooja Chaudhary, „Hunting for DOM-Based XSS vulnerabilities in mobile cloud-based online social network", FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE, Volume 79, Pages 319-336, FEB 2018
7. Materiały szkoleniowe EC-Council do egzaminu CEH v9 (Certified Ethical Hacker)

Streszczenie

Każda organizacja niezależnie od jej wielkości musi posiadać system bezpieczeństwa. Zrozumienie wartości danych wrażliwych oraz ochronę tychże informacji ułatwiają nam dziś między innymi regulacje branżowe i prawne. W praktyce stosuje się dedykowane narzędzia oraz rozwiązania, które najczęściej są zaszyte w komercyjnych usługach IT. W artykule autorzy przedstawili krótki przegląd ataków, których celem jest kompromitacja danych poufnych bądź zaprzestanie świadczenia usług, oraz metody zapobiegania im.

Abstract

Every organization, regardless of its size, must have a security system. Understanding the value of sensitive data and the protection of this information are facilitated today by, inter alia, industry and legal regulations. In practice, dedicated tools and solutions are most often stitched in commercial IT services. In the article, the authors presented a brief overview of attacks aimed at compromising confidential data or ceasing to provide services, as well as methods of preventing them.

Keywords: internet services, network attacks, compromise of confidential data

Bohdan Pustovyi
Computer Engineering Department
Odessa National Academy of Food Technologies
E-mail: b.pustoviy@gmail.com

Automation of analytical model construction for intellectual superstructure in next generation networks

Keywords: intellectual services, Markov model, analytical model, ngn, intellectual superstructure, centralized control principle, decentralized control principle.

Introduction

Currently, the number of services rendered by telecommunications networks has considerably grown. Intellectual services (IS) are specially distinguished in this list.

This is due to popularity of these services among the users, and, consequently, their profitability for operators; thus, the issues of IS efficiency control assessment are getting more urgent. The issues related to study of assessment methods of telecommunications services efficiency control are dealt with in works of [2, 4] and other scientists.

It should be noted that IS efficiency control issues are still not adequately investigated. According to ITU [3] recommendations, to determine efficiency of IS granting control, technical indices of the network operation should be calculated – the time of requisition for IS stay in the network, the probability of requisition deadlocking, the number of requisitions waiting for serving. To calculate these indices, analytical models of IS control systems are used.

Nearly all models of discrete systems with stochastic functioning are developed based on **Queuing Systems** (QS) the processes in which are accidental ones, in many cases the Markov or somehow associated with the Markov processes. That is why mathematical apparatus of the Markov processes theory can be used while solving tasks of queuing theory.

The use of the Markov processes is especially efficient and successful at QS studies and queuing networks (QN) with storing devices of limited capacity [1].

The use of Markov process for ISCCP representation

At the current stage of NGN development intellectual superstructure with centralized principle (ISCCP) of intellectual services control is used.

ISCCP can be represented as QS of type $M/M/1/r$ [2] based on the use of the Markov processes theory framework. To provide QS example (fig. 1) we will give the system description:

1. The system contains one serving device (SD) and is a *one-channel*.
2. The flow of requisitions entering the system, is homogenious.
3. Though, requisitions of some classes exist, nevertheless, we will still assume that λ and μ are similar for them.
4. The duration of requisitions serving in the device is an accidental value.
5. A storage device for requisitions has *limited capacity*.

Assumption [2]: The duration of requisitions serving in the device is divided exponentially with *intensity* $\mu=1/b$, where b – an average duration of requisitions serving in the device.

1. Buffering discipline – *with losses*: a requisition that entered the system and found the storage device full, is lost.
2. Nonpreemptive service priority order – due to entry order based on 'First In First Out' rule (FIFO).

QS with limited capacity storage device always has steady mode, since the length of a waiting line would not grow infinitely, even at big load values.

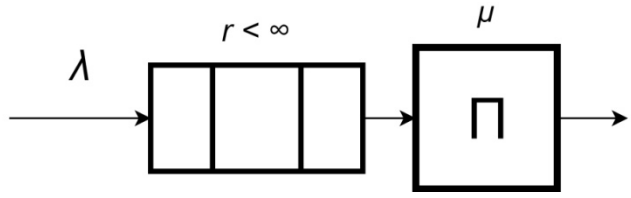


Fig. 1. QS with a limited capacity storage device

The use of ISCCP can cause a number of problems in highly loaded systems. Intellectual superstructure with decentralized control principle (ISDCP) can be a solution to all these problems. ISDCP can be viewed as several Queuing Systems connected with each other, that is, as a queuing network (QN).

The use of the Markov process for ISDCP representation

A combination of final numeral D of serving nodes in which requisitions circulate, thus transferring from one node to another, is a queuing network. ISDCP is an open-loop network. An open-loop network is such an open network to which requisitions come from the environment and are directed to the environment after being served in the network. Requisitions from one server output can come to inputs of others.

We will call requisitions ingress flow a requisitions flow that come to input of a specific server from the environment (from the program commutator), that is, not from output of any other server. In general case the number of ingress flows in the control system corresponds to a number of the servers used.

Description of ISDCP as QN [1]:

ISDCP is an open-loop exponential QN with D nodes that correspond to ISDCP servers.

1. QN nodes are *one-channel.*
2. Storage devices in nodes have a limited capacity r_i. Let's define $r_i = M$, where $i = \overline{1, D}$.
3. The ingress flow of requisitions is not *homogenious:* N classes of requisitions come to the system with intensities λ_{ij}, where $i = \overline{1, D}$ – servers number, $j = \overline{1, N}$ – requisitions class.
4. Buffering discipline in the nodes – with requisitions losses, if storing devices are full.
5. Nonpreemptive service priority order – *with relative priorities*: the lesser the N value, the higher the requisitions priority. A top priority requisition is selected each time from the storage device for serving. In this case, during arrival of top priority requisition to the system the serving is not interrupted.
6. The set matrixes $Q^j = \left\| q_{ik}^j \right\|$ of probabilities of a requisition transmission from the current server i to other servers k, or serving by the current server, where $i, k = \overline{1, D}$ – the servers number for requisitions classes $j = \overline{1, N}$. When $i \neq k$ q^j corresponds to probability of transmission, when $i = k$ q^j corresponds to the probability of serving by the current server.

Assumptions and suppositions.

The duration of requisitions serving in QN nodes is divided exponentially with serving intensities: $\mu = 1/b_{ij}$, where $i = \overline{1, D}$ – servers number, $j = \overline{1, N}$ – the requisitions class. b_{ij} – average duration of the requisitions serving of j class on i – server. In open-loop QN stationary mode exists at any mode, since there cannot be infinite queues in the network nodes.

For the purpose of a more detailed analysis, we will use the Markov processes theory framework. This work suggests that ISDCP should be represented in the following way:

1. ISDCP – open-loop exponential queuing network (QN). In fig. 2 ISDCP is represented with two one-channel servers (fig. 2).
2. Storage devices in both servers have a limited capacity. In example (fig. 2) the storage devices capacity is assumed as: $r_1=r_2=1$.
3. Nonpreemptive service priority order – with relative priority. First class requisitions have a higher priority than another class priority.
4. Buffering discipline - nonpreemptive. A requisition that entered the system and found the storage device full, is lost.

Assumptions:

Requisitions of two classes with intensities λ_{11}, λ_{12}, λ_{21}, λ_{22} enter from the environment.

1. The duration of requisitions serving in QN nodes is divided *exponentially* with serving intensities: $\mu_{11}=1/b_{11}$, $\mu_{12}=1/b_{12}$, $\mu_{21}=1/b_{21}$, $\mu_{22}=1/b_{22}$, where b_{11}, b_{12}, b_{21}, b_{22} – average durations of serving.
2. The probability of serving a requisition of j-class by i-server will be defined as q_{ij}. Then requisitions after serving in server 1 with probabilities q_{11}, q_{12} are directed to server 2 with probabilities $(1- q_{11})$, $(1- q_{12})$ leaving QN. Requisitions after serving in server 2 with probabilities q_{21}, q_{22} are directed to server 1 with probabilities $(1- q_{21})$, $(1- q_{22})$ leaving QN.
3. Since requisitions can be lost in the network, open-loop QN is non-linear, that is, intensities of requisitions flows that come to QN nodes, are not associated with each other by linear dependence, and cannot be calculated by solving a system of linear algebraic equations.

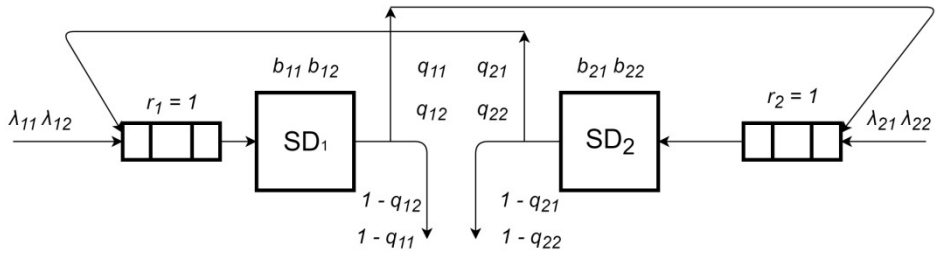


Fig. 2. The ISDCP with two servers is represented by QN

The Formation of Analytical Model

Markov processes states should be coded in order to define the stationary probability of states. This can achieved in the following way: (П,Ч).

For example, (fig. 2) Markov processes coding is performed in the following way: П = {0, 1, 2} – the state of a serving device, which is set by a requisition class, which is being served ('0' – the device is free; '1' or '2' – a requisition of class 1 or 2 respectively, is being served in the device).

The state of the storage device can be represented in the following way: Ч= {0, 1, 2, 11, 12, 22}, where '0' – means the absence of requisitions in the storage device; '1' – the presence of only one requisition of class 1 in the storage device; '2' – the presence of requisition of class 2 in the storage device; '11' – the presence of two class 1 requisitions in the storage device1; '22' – the presence of two class 2 requisitions in the storage device and '12' – the presence of one class 1 requisition and and one class 2 requisition in the storage device.

State '12' does not differentiate in which order these requisitions came to the system which is determined by availability of the relative priority between them – irrespective of the moment the requisitions came for serving, class 1 requisition will always be chosen.

In case of serving without priority, when requisitions of different classes are chosen for serving based on the order of their arrival, one more storage device state should be introduced – '21', which means that class 2 requisition came to the system earlier than class 1 requisition, whereas state '12' means that class 1 requisition came earlier to the system.

Then, the Markov process can occur in one of the following states at any moment:

E_0 : $k = 0$ – the system contains no requisition;

E_1 : $k = 1$ – the system contains 1 requisition being served in the device;

E_2 : $k = 2$ – the system contains 2 requisitions: one – being served in the device and the other one – waiting for in the storage device;

E_{r+1} : $k = r + 1$ – the system contains $(r + 1)$ requisitions: one – being served in the device and r – in the storage device.

The marked graph of random probability transitions is represented in fig.3 [2].

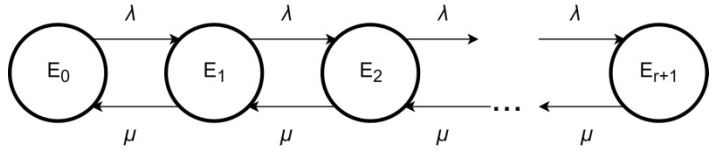


Fig. 1. The graph of Markov process transition

One and the same occurrence can take place in the system at the same time only once:

- the arrival of a requisition with intensity λ, that corresponds to increase in the requisitions by one in the system and the random probability transition to a state with number exceeding by one;
- completion of serving a requisition in a device with intensity μ that corresponds to lessening in the number of requisitions in the system and random probability transition to a state with number which is less by one.

Based on the defined states of the Markov process, the system of balance equations is generated (whose number equals to the number of states), and whose solving provides possibility for the Markov process probabilities defining and further – for obtaining technical indices of the network operation.

Assessment of the balance equations system

It should be noted that labor intensity of solving the balance equations system grows significantly at growing the number of classes of requisitions for IS and at growing the waiting line. Figures 4 and 5 show the change of the system states (that is, the size of the equations system) subject to the number of classes of requisitions for IS, and to the waiting line. As we can see, the abrupt increase in the number of states of the equations system disables the use of manual calculations.

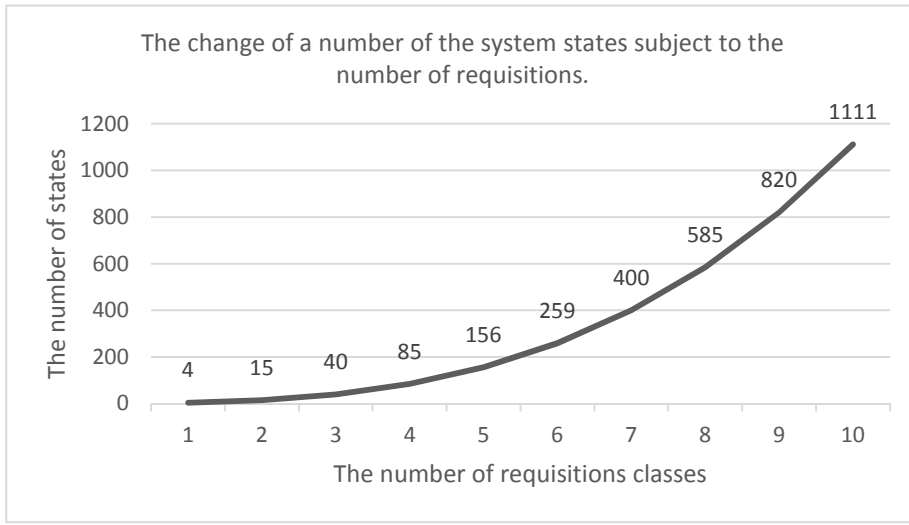


Fig. 2. The diagram of the change in the number of possible states of the system at the change of requisitions classes number in the system with 1 SD and the waiting line for serving of 2 requisitions

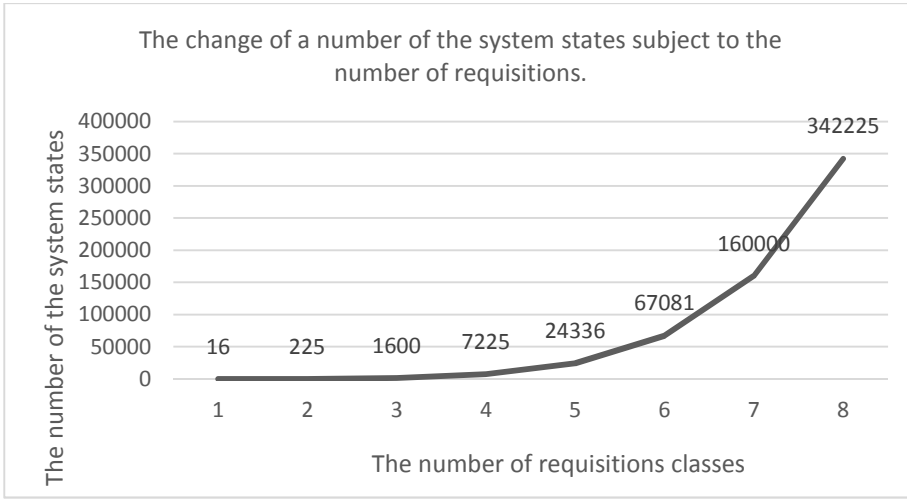


Fig. 3. The diagram of the change in the number of possible states of the system at the change of requisitions classes number in the system with 2 SD and the waiting line for serving of 2 requisitions

Software description

In this work the software is proposed which is designed for automation of the system probable states calculation, the construction of the Markov process transition graph and generation of equations system to determine stationary probabilities. SW is designed for the use both in ISCCP (QS), and ISDCP (QN).

The SW written in the JavaScript scripting language with use of the open library React JS which allows its launching in any browser. The maximal number of states for calculation of QS (QN) depends on a computer technical characteristics.

The following assumptions were accepted when developing this SW:

- the SD can serve only one class of requisitions at a time;
- the queue for serving is built as per FIFO discipline;
- the system functions without requisitions losses.

The input data of the developed SW is:

- SD number;
- the maximal length of serving waiting line;
- the number of requisitions classes and their priority.

In fig. 6,7,8 SW interface is shown. The SW performs the following functions:

- ensures data input for calculation;
- performs coding of all possible states of the Markov process for the system chosen;
- constructs and models the Markov process transition graph;
- the SW generates the equations system for determining stationary probabilities based on the generated transition graph;
- generates expressions for technical indices calculation based on found values of stationary probabilities of an accidental process states:
  - average number of requisitions in the queue;
  - average number of requisitions in the system;
  - probability of requisitions loss;
  - allows obtained results download to a text file.

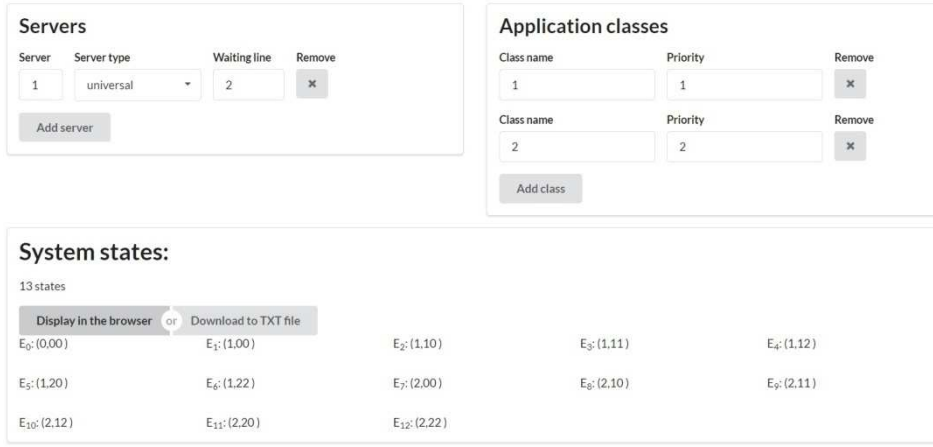


Fig. 4. Form for input data introduction in SW

Based on the introduced input data, the SW codes all possible states of the Markov process for a set system. The coding occurs in the following way:

$E_n = (\Pi_1 Ч_1 \,/\, \Pi_2 Ч_2 \,/\, \ldots \,/\, \Pi_i Ч_i / .. / \Pi_N Ч_N)$, where: (1)

- E_n – the Markov process state, n – the state number;
- $\Pi_i = \overline{0,\ m}$ – the state of a serving device, in which only one requisition of a specific class can be served at a time ('0' - the device is free, 'm' – 'm' class requisition is served), 'i' – the device number, 'N' – SD number;

- $Ч_i = \underbrace{\overline{0, m \ldots m}}_{k}$ – the state of the serving queue ('0' – the queue is empty, 'm' – 'm' class requisition in queue for being served), 'k' - the number of places in queue for being served.

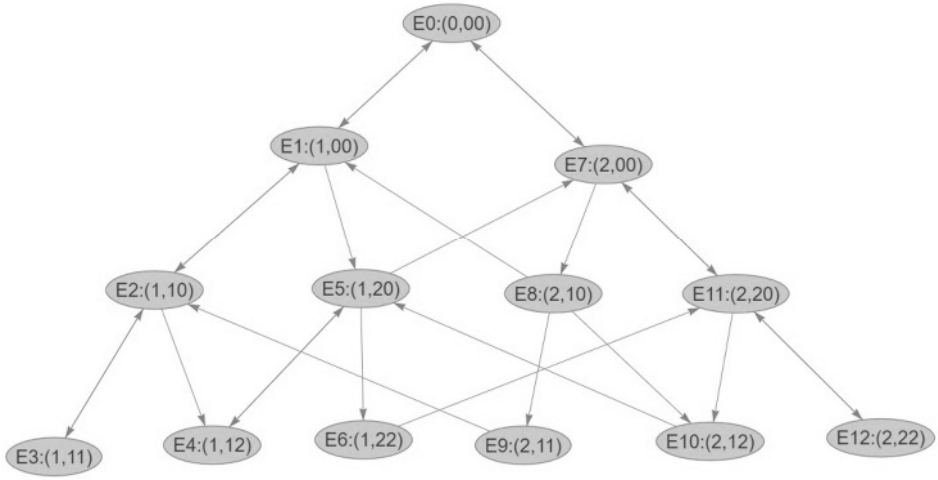


Fig. 5. The marked Markov process transition graph

Using encoded states, the SW constructs the Markov processes marked graph. In fig.7 the given example of the Markov process transitions graph in the system with one SD with maximal waiting line of two requisitions with different priority.

Subject to transitions graph we'll generate the equations system to define the values of stationary probabilities:

Display in the browser or Download to TXT file

$p_0 * \lambda_1 + p_0 * \lambda_2 = p_1 * \mu_1 + p_7 * \mu_2$

$p_1 * \mu_1 + p_1 * \lambda_1 + p_1 * \lambda_2 = p_0 * \lambda_1 + p_2 * \mu_1 + p_5 * \mu_1$

$p_2 * \mu_1 + p_2 * \lambda_1 + p_2 * \lambda_2 = p_1 * \lambda_1 + p_3 * \mu_1 + p_4 * \mu_1$

$p_3 * \mu_1 = p_2 * \lambda_1$

$p_4 * \mu_1 + p_4 * \mu_1 = p_5 * \lambda_1$

$p_5 * \mu_1 + p_5 * \lambda_1 + p_5 * \lambda_2 = p_4 * \mu_1 + p_6 * \mu_1 + p_7 * \lambda_1$

$p_6 * \mu_1 = p_{11} * \lambda_1$

$p_7 * \mu_2 + p_7 * \lambda_1 + p_7 * \lambda_2 = p_0 * \lambda_2 + p_8 * \mu_2 + p_{11} * \mu_2$

$p_8 * \mu_2 = p_1 * \lambda_2 + p_9 * \mu_2 + p_{10} * \mu_2$

$p_9 * \mu_2 = p_2 * \lambda_2$

$p_{10} * \mu_2 + p_{10} * \mu_2 = p_5 * \lambda_2$

$p_{11} * \lambda_1 + p_{11} * \mu_2 + p_{11} * \lambda_2 = p_7 * \lambda_2 + p_{10} * \mu_2 + p_{12} * \mu_2$

$p_{12} * \mu_2 = p_{11} * \lambda_2$

Fig. 6. Equations system of the Markov process stationary probabilities

The SW generates the equations system (2) for determining stationary probabilities based on the transitions graph:

$$\begin{cases} p0 \cdot \lambda1 + p0 \cdot \lambda2 = p1 \cdot \mu1 + p7 \cdot \mu2 \\ p1 \cdot \mu1 + p1 \cdot \lambda1 + p1 \cdot \lambda2 = p0 \cdot \lambda1 + p2 \cdot \mu1 + p5 \cdot \mu1 \\ p2 \cdot \mu1 + p2 \cdot \lambda1 + p2 \cdot \lambda2 = p1 \cdot \lambda1 + p3 \cdot \mu1 + p4 \cdot \mu1 \\ p3 \cdot \mu1 = p2 \cdot \lambda1 \\ p4 \cdot \mu1 + p4 \cdot \mu1 = p5 \cdot \lambda1 \\ p5 \cdot \mu1 + p5 \cdot \lambda1 + p5 \cdot \lambda2 = p4 \cdot \mu1 + p6 \cdot \mu1 + p7 \cdot \lambda1 \\ p6 \cdot \mu1 = p11 \cdot \lambda1 \\ p7 \cdot \mu2 + p7 \cdot \lambda1 + p7 \cdot \lambda2 = p0 \cdot \lambda2 + p8 \cdot \mu2 + p11 \cdot \mu2 \\ p8 \cdot \mu2 = p1 \cdot \lambda2 + p9 \cdot \mu2 + p10 \cdot \mu2 \\ p9 \cdot \mu2 = p2 \cdot \lambda2 \\ p10 \cdot \mu2 + p10 \cdot \mu2 = p5 \cdot \lambda2 \\ p11 \cdot \lambda1 + p11 \cdot \mu2 + p11 \cdot \lambda2 = p7 \cdot \lambda2 + p10 \cdot \mu2 + p12 \cdot \mu2 \\ p12 \cdot \mu2 = p11 \cdot \lambda2 \\ \sum_{k=0}^{12} p_k = 1 \end{cases} \quad (2)$$

where λ – the intensity of the requisitions arrival, μ – the intensity of the requisitions serving.

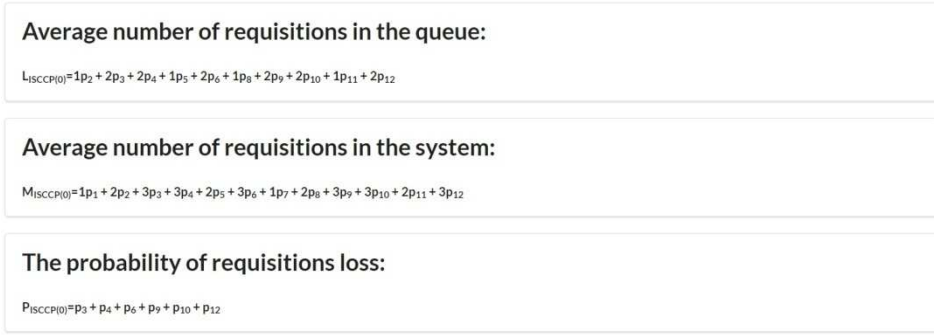


Fig. 9. The display of the Markov process transitions graph construction results

For the proposed QS (fig. 7) the SW generates expressions for technical indices calculation [1] abased on calculated values of stationary probabilities of the accidental process.

Average number of requisitions in the queue:

$$L=1p_2+2p_3+2p_4+1p_5+2p_6+1p_8+2p_9+2p_{10}+1p_{11}+2p_{12} \quad (3)$$

Average number of requisitions in the system:

$$M=1p_1 + 2p_2 + 3p_3 + 3p_4 + 2p_5 + 3p_6 + 1p_7 + 2p_8 + 3p_9 + 3p_{10} + 2p_{11} + 3p_{12} \quad (4)$$

The probability of requisitions loss:

$$P= p_3 + p_4 + p_6 + p_9 + p_{10} + p_{12} \quad (5)$$

This example demonstrates expressions for calculation of technical indices y for QS where there is only one control device, though, the SW allows calculation of indices i for QN with any number of servers.

Conclusions

Automation of the Markov processes balance equation generation reduces labor intensity of calculations significantly which allows the calculation of IC control systems technical indices with any number of servers. Besides, the use of the developed automation system will allow selection of control principle and the structure of NGN intellectual superstructure as early as the network design stage.

Literature

1. Aliev T.I. The basics of discrete systems modelling.– St-Pb:SPbGU ITMO, 2009. – 363 p.
2. Knyazeva N.O. Intellectual services control in the next generation networks:monograph /N.O.Knyazeva , S.V. Shestoopalov – Odesa: Bondarenko M.O., 2017. – 268 p.
3. 'International Telecommunication Union (ITU)', official Internet-agency. — Access mode: http://www.itu.int (accessed date March 20, 2015).
4. Berkman L.N. Invariance of converged networks control systems in emergency modes / Berkman L.N., L.O. Komarova, I. A. Boyko// Scientific notes of the Ukrainan Scientific-Research Communications Institute.. - 2014. - No. 1. - P. 11-15. - Access mode: http://nbuv.gov.ua/UJRN/Nzundiz_2014_1_4.

Abstract

The software is suggested for automation of analytical model construction for intellectual superstructure in next generation networks. The software is designed for automation of the system probable states calculation, the construction of the Markov process transition graph and generation of equations system to determine stationary probabilities. An example of analytical model construction is represented for intellectual superstructure with centralized control principle.

Streszczenie

Zaproponowano oprogramowanie do zautomatyzowania analitycznego modelu budowy intelektualnej nadbudowy w sieciach następnej generacji. Oprogramowanie jest przeznaczone do automatyzacji obliczania stanów prawdopodobieństwa systemu, konstrukcji wykresu przejścia procesu Markowa i generowania układu równań w celu ustalenia stacjonarnych prawdopodobieństw. Przykład konstrukcji modelu analitycznego jest reprezentowany dla nadbudowy intelektualnej z zasadą scentralizowanej kontroli.

Słowa kluczowe: usługi intelektualne, model Markowa, model analityczny, nadbudowa intelektualna, zasada scentralizowanej kontroli, zasada zdecentralizowanej kontroli.