Romuald Hoffmann[1]

# Modele wykrywania podatności oprogramowania oparte na stochastycznych równaniach różniczkowych

## 1. Wstęp

Modele wykrywania podatności oprogramowania VDM (ang. *Vulnerability Discovery Models*) ilustrują zmiany skuteczności detekcji podatności w całym cyklu życia oprogramowania systemu. Modele te mogą zostać użyte do opracowania metod i zaplanowania minimalizacji ryzyka wykorzystania luk w oprogramowaniu przez atakującego. Pozwalają zarówno menadżerom, jak i projektantom systemów informatycznych na optymalną alokację zasobów do zaprojektowania lub przygotowania i wykonania poprawek (łatek) w oprogramowaniu. Tego typu modele pozwalają również końcowemu użytkownikowi na szacowanie ryzyka bezpieczeństwa swoich systemów.

W literaturze opublikowano kilka modeli wykrywania podatności VDM zarówno deterministycznych, jak i stochastycznych. W niniejszym artykule zostały zaprezentowane najważniejsze z istniejących niewielu stochastycznych modeli wykrywania podatności, bazujących na teorii stochastycznych równań różniczkowych. Modele omówiono w kontekście ich deterministycznych odpowiedników, które również przywołano.

## 2. Modele wykrywania podatności oprogramowania

W ostatniej dekadzie badacze zaproponowali kilka modeli wykrywania podatności oprogramowania VDM. Modele te bazują na różnych założeniach, stąd też prezentują różne możliwości predykcji procesu wykrywania podatności oprogramowania. Większość modeli to modele w funkcji czasu. Pierwszy

[1] Wojskowa Akademia Techniczna w Warszawie, Wydział Cybernetyki.

to model termodynamiczny (*Anderson's Thermodynamic* – AT), zaproponowany przez R. Andersona[2] najpierw jako model niezawodności oprogramowania. E. Rescorla[3] zaproponował dwa inne modele – kwadratowy (*Rescorla Quadratic Model* – RQ) i wykładniczy (*Rescorla Exponential Model* – RE). Czwartym modelem jest tzw. model logistyczny Alhazmiego–Malaiyi[4] (*Alhazmi and Malaiya's Logistic Model* – AML). Następne dwa modele to model logarytmiczny Poissona[5] (*Logarithmic Poisson Model* – LP), bezpośrednio wyprowadzony z modelu niezawodnościowego Musa–Okumoto[6], oraz prosty model liniowy. Są to modele zależne od czasu, który jest traktowany jako czas kalendarzowy.

Spośród powyższych modeli model logistyczny AML był przedmiotem wielu eksperymentów i walidacji: zaczynając od systemów operacyjnych[7] (np. Windows NT/95/98/2K/XP, Redhat 6.2/7.1, Fedora), poprzez przeglądarki internetowe[8] (np. IE, Firefox, Mozilla), kończąc na serwerach webowych[9] (np. ISS, Apache). Wyniki eksperymentów walidacyjnych w większości przypadków wskazują na dobre możliwości modelu AML dopasowania do danych rzeczywistych i dobre właściwości predykcji. W literaturze nie ma jednoznacznych wyników wskazujących na przyjęcie lub odrzucenie modelu AML, tak jak w przypadku

---

2 R. Anderson, *Security in Open versus Closed Systems – The Dance of Boltzmann, Coase and Moore*, w: *Confer. on Open Source Software Economics*, Cambridge University, Toulouse 2002.

3 E. Rescorla, *Is finding security holes a good idea?*, „Security and Privacy" 2005, January–February, s. 14–19.

4 O.H. Alhazmi, Y.K. Malaiya, *Quantitative Vulnerability Assessment of Systems Software*, w: *Annual Reliability and Maintainability Symposium, 2005. Proceedings*, IEEE, Alexandria (Virginia) 2005, s. 615–620.

5 Y.K. Malaiya, A. von Mayrhauser, P.K. Srimani, *An Examination of Fault Exposure Ratio*, „IEEE Transactions on Software Engineering" 1993, vol. 19, no. 11, November, s. 1087–1094.

6 J.D. Musa, K. Okumoto, *A logarithmic Poisson execution time model for software reliability measurement*, w: *Proceeding ICSE '84, Proceedings of the 7th international conference on Software engineering*, IEEE Press, Piscataway (New Jersey) 1984, s. 230–238.

7 O.H. Alhazmi, Y.K. Malaiya, *Quantitative Vulnerability Assessment of Systems Software*, w: *Annual Reliability and Maintainability Symposium, 2005. Proceedings*, IEEE, 2005, s. 615–620; O. Alhazmi, Y. Malaiya, *Modeling the vulnerability discovery process*, w: *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, red. S. Kawada, IEEE, Chicago 2005, s. 129–138; O. Alhazmi, Y. Malaiya, *Application of vulnerability discovery models to major operating systems*, „IEEE Transactions on Reliability" 2008, vol. 57, no. 1, s. 14–22; O. Alhazmi, Y. Malaiya, I. Ray, *Security vulnerabilities in software systems: A quantitative perspective*, Data and Applications Security, LNCS 3654, 2005, s. 281–294.

8 S.-W. Woo, O.H. Alhazmi, Y.K. Malaiya, *An analysis of the vulnerability discovery process in web browsers*, w: *Proceedings of the 10th IASTED SEA '06*, 13–15.11.2006, s. 172–177.

9 S.-W. Woo, H.C. Joh, O.H. Alhazmi, Y.K. Malaiya, *Modeling vulnerability discovery process in Apache and IIS HTTP servers*, „Computers & Security" 2011, vol. 30, issue 1, s. 50–62.

modelu AT. Stąd też A.K. Shrivastava, R. Sharma, P.K. Kapur[10] zaproponowali wersję modelu AML bazującą na stochastycznym równaniu różniczkowym.

W literaturze przedmiotu jak do tej pory nie wskazano wprost wersji modelu wykładniczego Rescorli opartego na stochastycznych równaniach różniczkowych. W związku z tym lukę rozszerzenia modelu wykładniczego Rescorli wypełnimy w niniejszym artykule, zauważając, że model ten wywodzi się wprost z modelu Goel–Okumoto, dla którego opublikowano wersję bazującą na stochastycznym równaniu różniczkowym Itô.

W podpunktach przyjęto konwencję jednego oznaczenia łącznej liczby wykrytych podatności do chwili $t$ przez $N(t)$. W zależności od kontekstu $N(t)$ będzie funkcją deterministyczną lub procesem stochastycznym. Aby uniknąć nieporozumienia, za każdym razem zostanie zaznaczony kontekst, w którym używa się danego oznaczenia. Ponadto przez $\sigma \cdot \gamma(t)$ oznaczono standaryzowany biały szum Gaussa, $\gamma(t) = dW(t)/dt$. Przez $W(t)$ oznaczono proces stochastyczny Wienera.

## 2.1. Model logistyczny Alhazmiego–Malaiyi

Jak już wspomniano wcześniej, dotychczas opublikowane prace wskazują, że deterministyczny model logistyczny zaproponowany przez O.H. Alhazmiego i Y.K. Malaiyę w wielu przypadkach najlepiej opisuje, w sensie ilościowym, proces wykrywania podatności oprogramowania. Model ten zależy od czasu oraz od liczby pozostałych w danej chwili podatności do wykrycia.

Model logistyczny Alhazmiego–Malaiyi[11] został zaproponowany w 2005 r. w odniesieniu do podatności oprogramowania systemów operacyjnych Windows 98 i Windows NT 4.0. W modelu tym stopę wykrywalności znanych podatności oprogramowania w funkcji czasu (ang. *a time-based known-vulnerability discovery model*) opisuje następujące równanie różniczkowe:

$$\frac{d}{dt}N(t) = a \cdot N(t) \cdot (b - N(t)) \text{ , t} \geq 0, \tag{3}$$

gdzie: czas $t$ traktuje się jako czas kalendarzowy; $N(t)$ oznacza skumulowaną łączną liczbę podatności do chwili $t$; $a$ i $b$ są stałymi wyznaczonymi empirycznie

[10] A.K. Shrivastava, R. Sharma, P.K. Kapur, *Vulnerability Discovery Model for a Software System Using Stochastic Differential Equation*, w: *Proceedings of 2015 1st International Conference on Futuristic trend in Computational Analysis and Knowledge Management (ABLAZE-2015)*, red. B. Kumar, J.S. Jassi, IEEE, Amity University Greater Noida, Noida 2015, s. 199–205.

[11] O.H. Alhazmi, Y.K. Malaiya, *Quantitative…*, op.cit.

w wyniku analizy zgromadzonych danych historycznych; $a$ interpretuje się jako intensywność wykrywania podatności; $b$ oznacza łączną maksymalną liczbę podatności, które mogą zostać wykryte w oprogramowaniu.

Rozwiązaniem ogólnym równania różniczkowego (3) jest funkcja:

$$N(t) = \frac{b}{b \cdot c \cdot e^{-a \cdot b \cdot t} + 1}, \tag{4}$$

gdzie $c$ jest stałą uzyskaną w wyniku rozwiązania równania (3). Aby obliczyć stałą $c$, przyjmijmy, że $N(0) = k$. Stąd na podstawie (4) współczynnik $b \cdot c$ przyjmuje postać: $b \cdot c = (b - k) / k$. Ostatecznie $N(t)$ przybiera postać:

$$N(t) = \frac{b}{\frac{b-k}{k} \cdot e^{-a \cdot b \cdot t} + 1}. \tag{5}$$

Jeżeli przyjmiemy $a(t) = a \cdot N(t)$, to równanie (3) przyjmuje postać:

$$\frac{d}{dt} N(t) = a(t) \cdot (b - N(t)),\ t \geq 0, \tag{6}$$

gdzie $a(t)$ jest dane wzorem:

$$a(t) = \frac{a \cdot b}{\frac{b-k}{k} \cdot e^{-a \cdot b \cdot t} + 1}. \tag{7}$$

## 2.2. Model logistyczny Alhazmiego–Malaiyi na bazie stochastycznego równania różniczkowego Itô

A.K. Shrivastava, R. Sharma i P.K. Kapur[12] w 2015 r. zaproponowali randomizację modelu logistycznego Alhazmiego–Malaiyi na podstawie stochastycznego równania różniczkowego Itô. Mianowicie zauważono, że stopę wykrywalności znanych podatności oprogramowania opisaną równaniem różniczkowym (6)

[12] A.K. Shrivastava, R. Sharma, P.K. Kapur, op.cit.

można rozszerzyć poprzez randomizację do następującego stochastycznego równania różniczkowego[13]:

$$\frac{d}{dt}N(t) = \left(a(t) + \sigma \cdot \gamma(t)\right) \cdot \left(b - N(t)\right), \tag{8}$$

gdzie: $a(t)$ jest dane wzorem (7), $\sigma \cdot \gamma(t)$ jest standaryzowanym białym szumem Gaussa, $\gamma(t) = dW(t) / dt$, $W(t)$ jest procesem Wienera[14].

Następnie równanie różniczkowe (8) może zostać rozszerzone i wyrażone jako różniczka Itô o następującej postaci:

$$dN(t) = \left(a(t) - \frac{1}{2}\sigma^2\right) \cdot \left(b - N(t)\right)dt + \sigma \cdot \left(b - N(t)\right)dW(t). \tag{9}$$

Stosując formułę Itô[15] do równania (9), otrzymujemy następujące rozwiązanie[16]:

$$N(t) = b - (b - k) \cdot \exp\left(-\int_0^t a(\tau)\,d\tau - \sigma W(t)\right).$$

Stąd też wartość oczekiwana procesu stochastycznego $N(t)$ liczby podatności oprogramowania w chwili $t$ jest wyrażona wzorem:

$$E\left[N(t)\right] = b \cdot \left(1 - \frac{\frac{b-k}{k} \cdot \exp\left(-(a \cdot b \cdot t - \frac{1}{2}\sigma^2 t)\right)}{\frac{b-k}{k} \cdot \exp\left(-a \cdot b \cdot t\right) + 1}\right).$$

Shrivastava, Sharma i Kapur, badając podatności systemu Windows-Vista za lata 2006–2014 oraz Adobe Flash Player za lata 2005–2015 i wykorzystując opisany powyżej model oparty na stochastycznym równaniu różniczkowym (9), doszli do wniosku, że wykazuje on lepszą predykcję liczby podatności niż oryginalny model logistyczny Alhazmiego–Malaiyi. Należy tutaj zauważyć, że Alhazmi

[13] P.K. Kapur, H. Pham, A. Gupta, P.C. Jha, *Software Reliability Assessment with OR Applications*, Springer, London 2011.

[14] K. Sobczyk, *Stochastyczne równania różniczkowe. Teoria i zastosowania*, WNT, Warszawa 1996.

[15] Ibidem.

[16] A.K. Shrivastava, R. Sharma, P.K. Kapur, op.cit.

i Malaiya[17] badali dopasowanie swojego modelu w odniesieniu do podatności systemów operacyjnych Windows 95, Windows XP, RH Linux 6.2.

## 2.3. Model wykładniczy Rescorli

E. Rescorla[18] identyfikował trendy procesów wykrywania podatności oprogramowania, analizując dane historyczne z wykorzystaniem testów statystycznych. Analizował dane dotyczące systemów operacyjnych Windows NT 4.0, Solaris 2.5.1, FreeBSD 4.0 i ReadHat 6.2.

Bazą wyjściową modelu wykładniczego Rescoli jest model rosnącej niezawodności oprogramowania SRGM Goel–Okumoto[19], który został dopasowany do danych historycznych. Model wykładniczy Rescorli wykrywania znanych podatności oprogramowania, w ogólnej postaci, można przedstawić następująco:

$$n(t) = \frac{d}{dt} N(t) = N \cdot \lambda \cdot e^{-\lambda \cdot t}, \tag{10}$$

gdzie: $N$ jest stałe i oznacza całkowitą (maksymalną) liczbę podatności, które można wykryć w systemie; $\lambda$ jest stałym współczynnikiem opisującym stopę wykrywania podatności oprogramowania.

Po scałkowaniu $n(t)$ otrzymujemy skumulowaną liczbę wykrytych podatności oprogramowania w funkcji czasu. Całka ogólna równania (10), opisująca łączną liczbę wykrytych podatności w funkcji czasu, w modelu wykładniczym Rescorli przedstawia się następująco: $N(t) = \int N \cdot \lambda \cdot e^{-\lambda \cdot t}\, dt = -N \cdot e^{-\lambda \cdot t} + C$. Aby $N(0) = 0$, przyjmujemy stałą całkowania $C$ równą $N$ i ostatecznie otrzymujemy zależność na łączną liczbę podatności do chwili $t$:

$$N(t) = N \cdot (1 - e^{-\lambda \cdot t}),\ t \geq 0. \tag{11}$$

[17] O. Alhazmi, Y. Malaiya, *Modeling…*, op.cit.

[18] E. Rescorla, op.cit.

[19] A.L. Goel, K. Okumoto, *Time-Dependent Error Detection Rate Model for Software and Other Performance Measures*, „IEEE Transactions on Reliability" 1979, vol. R-28, no. 3, August, s. 206–211.

## 2.4. Model wykładniczy Rescorli na bazie stochastycznego równania różniczkowego Itô

Aby rozszerzyć model wykładniczy Rescorli, opisany przez wyrażenie (11), wystarczy zauważyć, że model ten wywodzi się wprost z modelu Goel–Okumoto, opisanego następującą postacią równania różniczkowego[20]:

$$\frac{d}{dt}N(t) = \lambda \cdot (N - N(t)),\ t \geq 0,\ N(0) = 0, \tag{12}$$

którego rozwiązaniem jest funkcja $N(t)$, dana wzorem (11).
Zauważmy, że możemy uogólnić równanie (12) następującym równaniem:

$$\frac{d}{dt}N(t) = \lambda(t) \cdot (N - N(t))\ t \geq 0,\ N(0) = 0. \tag{13}$$

Równanie (13), podobnie jak w modelu AML, można rozszerzyć do następującego stochastycznego równania różniczkowego Itô:

$$\frac{d}{dt}N(t) = \left(\lambda(t) + \sigma \cdot \gamma(t)\right) \cdot \left(N - N(t)\right). \tag{14}$$

Następnie, na podstawie wniosków Kapura, Phama, Gupty[21], równanie różniczkowe (14) może zostać rozszerzone i wyrażone jako różniczka Itô o następującej postaci:

$$dN(t) = \left(\lambda(t) - \frac{1}{2}\sigma^2\right) \cdot \left(N - N(t)\right)dt + \sigma \cdot \left(N - N(t)\right)dW(t),$$

dla której rozwiązaniem jest następujący proces stochastyczny:

$$N(t) = N \cdot \left(1 - \exp\left(-\int_0^t \lambda(\tau)\,d\tau - \sigma W(t)\right)\right). \tag{15}$$

[20] P.K. Kapur, H. Pham, A. Gupta, P.C. Jha, op.cit., s. 55–56.
[21] Ibidem, s. 288–291.

Wiemy, że w modelu Goel–Okumoto, a tym samym w modelu wykładniczym Rescorli, $\lambda(t) = const = \lambda$. Stąd też proces stochastyczny $N(t)$, stanowiący o liczbie podatności oprogramowania, przyjmuje postać:

$$N(t) = N \cdot \left(1 - \exp\left(-\lambda t - \sigma W(t)\right)\right), \tag{16}$$

a jego wartość oczekiwana wyraża się wzorem:

$$E\left[N(t)\right] = N \cdot \left(1 - \exp\left(-(\lambda \cdot t - \frac{1}{2}\sigma^2 t)\right)\right). \tag{17}$$

Wyrażenie (16) stanowi model wykładniczy Rescorli zbudowany na bazie stochastycznego równania różniczkowego i stanowi rozszerzony ekwiwalent jego pierwotnej wersji.

## 3. Podsumowanie i kierunki dalszych badań

Z uwagi na to, że modele wykrywania podatności oprogramowania VDM są spokrewnione z modelami niezawodnościowymi, należy odnotować, że w ostatniej dekadzie można zaobserwować zwiększone zainteresowanie zastosowaniem stochastycznych równań różniczkowych Itô do modyfikacji istniejących modeli niezawodnościowych oprogramowania[22]. W efekcie przekłada się to na lepsze właściwości predykcji z wykorzystaniem w taki sposób zbudowanych modeli. Trend ten zarysował się również, jak wynika z badań Shrivastavy, Sharmy i Kapura dotyczących modelu AML, w odniesieniu do modeli wykrywania podatności oprogramowania. W tym kontekście zaproponowana w niniejszym artykule rozszerzona zrandomizowana wersja modelu Rescorli wymaga jeszcze wykonania analiz porównawczych z wyjściowym modelem.

Rozwój zrandomizowanych modeli VDM pozwoli na precyzyjniejszą analizę procesu wykrywania podatności. Oznacza to w praktyce dokładniejsze określenie czasu koniecznego do uzyskania wymaganego poziomu bezpieczeństwa, a także wskazania zasobów osobowych, sprzętowych oraz programowych do jego osiągnięcia. Należy zwrócić uwagę na fakt, że większość interesujących

[22] Ibidem, s. 283–312.

wniosków można wysnuć z systematycznej analizy procesu wykrywania podatności odniesionych do czasu kalendarzowego.

Ciekawym kierunkiem badań może być zatem rozbudowanie istniejących, a także budowa nowych modeli VDM na podstawie stochastycznych równań różniczkowych względem procesu Poissona w połączeniu z procesem Wienera. Pozwoli to na uwzględnienie w konstruowanym modelu stochastycznym zarówno białego szumu Gaussa, jak i sporadycznych, znacznie większych zaburzeń losowych, występujących w wyniku ujawnienia kilku podatności w krótkim czasie.

## Bibliografia

Alhazmi O., Malaiya Y., *Application of vulnerability discovery models to major operating systems*, „IEEE Transactions on Reliability" 2008, vol. 57, no. 1, s. 14–22.

Alhazmi O., Malaiya Y., *Modeling the vulnerability discovery process*, w: *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, red. S. Kawada, IEEE, Chicago 2005, s. 129–138.

Alhazmi O.H., Malaiya Y.K., *Prediction Capabilities of Vulnerability Discovery Models*, w: *RAMS '06. Annual Reliability and Maintainability Symposium*, IEEE, Newport Beach 2006, s. 86–91.

Alhazmi O.H., Malaiya Y.K., *Quantitative Vulnerability Assessment of Systems Software*, w: *Annual Reliability and Maintainability Symposium, 2005. Proceedings*, IEEE, Alexandria (Virginia) 2005, s. 615–620.

Alhazmi O.H., Malaiya Y., Ray I., *Security vulnerabilities in software systems: A quantitative perspective*, w: *Data and Applications Security*, LNCS 3654, 2005, s. 281–294.

Anderson R., *Security in Open versus Closed Systems – The Dance of Boltzmann, Coase and Moore*, w: *Conference on Open Source Software Economics*, Cambridge University, Toulouse 2002.

Goel A.L., Okumoto K., *Time-Dependent Error Detection Rate Model for Software and Other Performance Measures*, „IEEE Transactions on Reliability" 1979, vol. R-28, no. 3, August, s. 206–211.

*Handbook of Software Reliability Engineering*, red. M.R. Lyu, McGraw-Hill, New York 1995.

Kapur P.K., Pham H., Gupta A., Jha P.C., *Software Reliability Assessment with OR Applications*, Springer, London 2011.

Kim J., Malaiya Y.K., Ray I., *Vulnerability Discovery in Multi-Version Software Systems*, w: *Proceedings of 10th IEEE High Assurance Systems Engineering Symposium*, 2007, s. 141–148.

Malaiya Y.K., Mayrhauser A. von, Srimani P.K., *An Examination of Fault Exposure Ratio*, „IEEE Transactions on Software Engineering" 1993, vol. 19, no. 11, November, s. 1087–1094.

Musa J.D., *Software Reliability Engineering*, McGraw-Hill, New York 1999.

Musa J.D., Okumoto K., *A logarithmic Poisson execution time model for software reliability measurement*, w: *Proceeding ICSE '84, Proceedings of the 7th international conference on Software engineering*, IEEE Press, Piscataway (New Jersey) 1984, s. 230–238.

Rescorla E., *Is finding security holes a good idea?*, „Security and Privacy" 2005, January–February, s. 14–19.

Shrivastava A.K., Sharma R., Kapur P.K., *Vulnerability Discovery Model for a Software System Using Stochastic Differential Equation*, w: *Proceedings of 2015 1st International Conference on Futuristic trend in Computational Analysis and Knowledge Management (ABLAZE-2015)*, red. B. Kumar, J.S. Jassi, IEEE, Amity University Greater Noida, Noida 2015, s. 199–205.

Sobczyk K., *Stochastyczne równania różniczkowe. Teoria i zastosowania*, WNT, Warszawa 1996.

Tamura Y., Yamada S., *A flexible stochastic differential equation model in distributed development environment*, „European Journal of Operational Research" 2005, vol. 168, no. 1, s. 143–152.

Woo S.-W., Alhazmi O.H., Malaiya Y.K., *An analysis of the vulnerability discovery process in web browsers*, w: *Proceedings of the 10th IASTED SEA '06*, 13–15.11.2006, red. A.M.K. Cheng, IASTED, Dallas 2006, s. 172–177.

Woo S.-W., Joh H.C., Alhazmi O.H., Malaiya Y.K., *Modeling vulnerability discovery process in Apache and IIS HTTP servers*, „Computers & Security" 2011, vol. 30, issue 1, s. 50–62.

Yamada S., Nishigaki A., Kimura M., *A stochastic differential equation model for software reliability assessment and its goodness of fit*, „International Journal of Reliability and Applications" 2003, vol. 4, no. 1, s. 1–11.

* * *

## Vulnerability Discovery Models for a Software System Using Stochastic Differential Equations

**Abstract**

Vulnerability discovery models (VDMs) illustrate changes in the vulnerability detection processes of software during system lifecycles. So far very few VDMs based on stochastic differential equations have been proposed. In this paper, there were presented two vulnerability discovery models based on Itô-type stochastic differential

equations. The first was the Alhazmi-Malaiya Logistic Model based on the stochastic differential equation proposed by Shrivastava, Sharma and Kapur in 2015. The second one, proposed in this paper, was a modified Rescorla Exponential Model using the Itô stochastic differential equation. The proposed modified Rescorla model was obtained by using the stochastic differential equation approach to the Goel-Okumoto software reliability model.

**Keywords:** Itô-type stochastic differential equation, stochastic differential equation, SDE, Vulnerability Discovery Models, VDM, Alhazmi-Malaiya Logistic Model, AML, Rescorla Exponential Model, RE, vulnerability discovery process in software systems