Analisis dan Deteksi *Backdoor* pada *Content Management System* Menggunakan Metode *Signature-based* dan *Static Analysis*

Chanief Budi Setiawan[1], Dedy Hariyadi[1*], Rama Sahtyawan[1], Arief Ikhwan Wicaksono[1], Akas Wisnuaji[2]
[1] Program Studi Teknologi Informasi, Universitas Jenderal Achmad Yani Yogyakarta
[2] PT Widya Adijaya Nusantara
chanief.b.s@gmail.com, dedy@unjaya.ac.id, ramasahtyawan@gmail.com, ariefikhwanwicaksono@gmail.com, aji@widyasecurity.com

Abstrak

Pemanfaatan *Content Management System (CMS)* sebagai *platform* pengembangan situs web cukup tinggi. *CMS* menawarkan kemudahan dalam implementasi situs web dengan mudah dengan berbagai fitur termasuk didalamnya penggaya dan tema yang disediakan untuk memperindah tampilan. Namun dibalik kemudahan tersebut terdapat ancaman berupa penyisipan kode-kode jahat pada penggaya ataupun tema. Oleh sebab itu dalam pengembangan situs web harus mengimplementasikan dari *Secure-Software Development Life Cycle* dengan memperhatikan setiap tahap pengembangan dengan melakukan analisis dan deteksi setiap kode. Hal ini untuk mendeteksi penyisipan kode-kode jahat oleh pihak yang tidak bertanggung jawab. Melakukan analisis dan deteksi terkait kode jahat yang disisipkan dalam penggaya ataupun tema pada *CMS* sejak awal pengembangan merupakan langkah preventif terhadap serangan siber. Pada penelitian ini diusulkan melakukan *hybrid methods* dalam menganalisis dan mendeteksi *signature-based* dan *static analysis* supaya hasilnya lebih komprehensif. Berdasarkan penelitian ini, hasil dari pemindaian *signature-based* dan *static analysis* menggunakan *function reference* menunjukan hasil yang saling melengkapi.
Kata kunci: *backdoor, CMS*, keamanan siber, *signature-based, static analysis*

Abstract

Utilization of the Content Management System as a website development platform is quite high. CMS offers an easy way to implement websites easily with various features including styles and themes provided to beautify the appearance. But behind this convenience there is a threat in the form of inserting malicious codes in the style or theme. Therefore, the development of a website must implement the Secure-Software Development Life Cycle by paying attention to every stage of development by analyzing and detecting each code. This is to guard against the insertion of malicious codes by the responsible party. Performing analysis and detection of malicious code embedded in a style or theme on a CMS early on in development is a preventive measure against cyber attacks. In this study, it is proposed to carry out multiple methods in analyzing and based on signatures and static analysis so that the results are more comprehensive. Based on this research, the results of signature-based scanning and static analysis using function references show complementary results.
Keywords: *backdoor, CMS, cybersecurity, signature-based, static analyis*

1. PENDAHULUAN

Pengguna internet baik secara global maupun di Indonesia mengalami pertumbuhan yang sangat besar. Berdasarkan populasi jumlah penduduk dan pengguna internet secara global menunjukan penetrasi sebesar 64.2% (Miniwatts Marketing Group, 2020). Sedangkan menurut Asosiasi Penyelenggara Internet Indonesia penetrasi pengguna internet sebesar 64,8% (Asosiasi Penyelenggara Jasa Internet Indonesia, 2019). Artinya sebagian besar saat ini orang sudah memanfaatkan internet untuk berbagai hal kebutuhan. Bahkan berdasarkan studi di Spanyol penyebaran informasi di internet memerlukan sebuah alat bantu berupa *Content Management System* (CMS). Hasil studi tersebut kurang lebih 50% informasi di internet memanfaatkan CMS (Martinez-Caro et al., 2018).

SMA N 1 Magelang menerapkan CMS untuk pengembangan *e-learning* sekolah. *Platform* CMS yang digunakan adalah Wordpress. CMS seperti Wordpress dipilih karena memiliki beberapa kelebihan diantaranya (Risdanto, 2014):

1. Berlisensi *Open Source* sehingga mudah didapatkan dan dimodifikasi sesuai kebutuhan.
2. Tidak perlu kemampuan pemrograman web karena telah tersedia penggaya yang tersedia di publik.
3. Memiliki dukungan komunitas dan kemudahan lainnya.

Penggunaan CMS yang tidak hati-hati dapat menjadi sasaran Peretas, sebagai contoh pengguna penggaya yang tidak resmi sehingga menimbulkan potensi tersisipnya kode jahat pada penggaya tersebut. Peretas sengaja memasang kode jahat berupa *backdoor* untuk mempermudah atau mendapatkan akses ke server. Berdasarkan hal tersebut perlu memperhatikan kode-kode jahat yang disisipkan oleh peretas. Pada artikel ini mengusulkan deteksi kode-kode jahat menggunakan metode *signature-based* dan *static analysis* pada sebuah server bersistem operasi GNU/Linux.

2. METODE

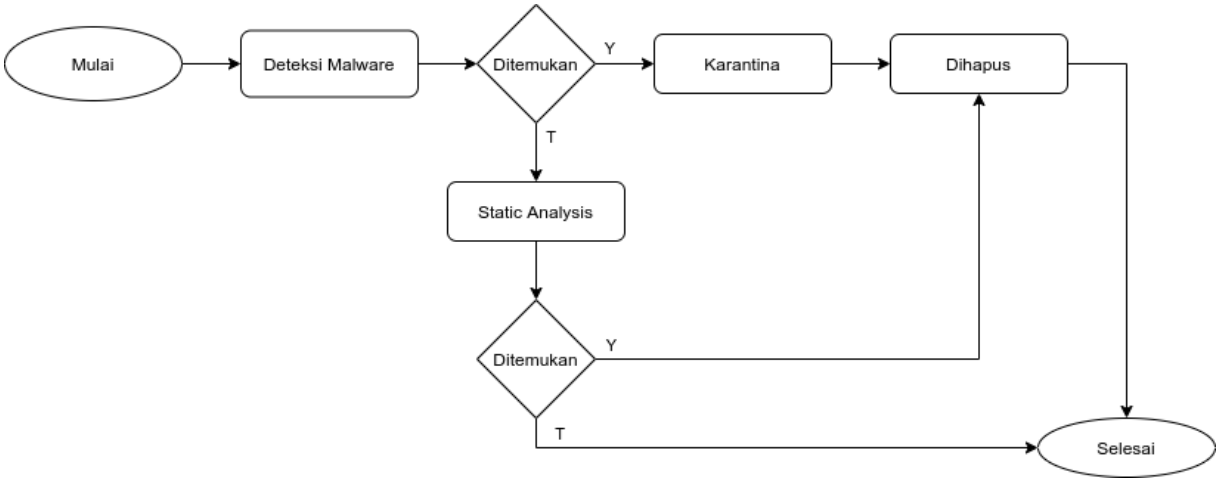
Pada proses pengembangan aplikasi sebelumnya masih menggunakan *Software Development Life Cycle*. Hal ini menimbulkan masalah terkait dengan keamanan sistem dan jaringan komputer seperti kebocoran data karena pada prosesnya tidak memperhatikan tinjauan keamanan ataupun serangan siber. Walaupun telah dilakukan pengujian melalui *penetration testing* pada akhir pengembangan aplikasi tidak memberikan jaminan bahwa aplikasi yang dibangun terbebas dari potensi celah keamanan dan serangan siber (Hariyadi et al., 2020).

Microsoft memberikan usulan dalam pengembangan aplikasi memperhatikan *Secure-Software Development Life Cycle*. Prinsip usulan Microsoft terkait *Secure-Software Development Life Cycle* adalah memperhatikan tinjauan keamanan pada setiap proses pengembangan. Adapun proses pengembangan aplikasi usulan Microsoft dapat dilihat pada **Error! Reference source not found.** (Microsoft Corporation, 2013).

Perlunya menerapkan *Secure-Software Development Life Cycle* pada pengembangan aplikasi untuk menghindari berbagai permasalah yang timbul seperti pemanfaatan penggaya dari pihak ketiga. Penggaya dari pihak ketiga harus diuji terlebih dahulu karena memiliki potensi-potensi yang menimbulkan celah keamanan diantaranya terdapat kode-kode jahat sepeti malware atau backdoor. Laporan Tahunan Badan Siber dan Sandi Negara pada tahun 2020 malware masih digunakan sebagai alat bantu penyerangan ke sebuah target. Adapun *Malware* yang paling banyak menyerang Indonesia bahkan global yaitu Ransom:Win32/Nemty.D dan Trojan:Win32/Eqtonex. Mitigasi terhadap serangan *Malware* di sisi server diantaranya melakukan *update* dan *patching* pada perangkat aplikasi Sistem Operasi (Badan Siber dan Sandi Negara & Indonesia Honeynet Project, 2020). Selain memanfaatkan *Malware*, Peretas juga menanamkan kode jahat yang berfungi mendapatkan akses kembali setelah menembus sistem pertahanan. Teknik ini disebut *backdoor* dengan tujuan dari pemasangan *backdoor* ini sebagai wujud tahapan *maintaining access* pada tahapan asesmen keamanan informasi. Namun, oleh Peretas teknik ini dimanfaatkan untuk tindak kejahatan dengan cara mengambil alih server (*server takeover*) (Lockheed Martin Corporation, 2015).

Gambar 1. Microsoft *Secure-Software Development Life Cycle*

Teknik yang digunakan peretas dalam menyisipkan kode-kode jahat seperti *backdoor* perlu diantisipasi dengan melakukan analisis dan deteksi pada *Secure-Software Development Life Cycle*. Untuk mempermudah dalam melakukan analisis dan deteksi menggunakan dua metode analisis, yaitu *signature-based* dan *static analysis*. Pada artikel ini diusulkan tahapan analisis backdoor seperti pada **Error! Reference source not found.**.



Gambar 2. Tahapan Analisis *Backdoor*

3. HASIL DAN PEMBAHASAN

Pada tahap pengujian di penelitian ini menggunakan sampel *backdoor* yang dipasang pada server yang menggunakan *Content Management System (CMS)*. Berdasarkan penelitian sebelum CMS yang banyak diserang oleh peretas adalah Open Journal System (OJS) (Hariyadi & Nastiti, 2021). Sedangkan untuk sampel *backdoor* yang akan disisipkan pada CMS adalah b374k, hal ini sesuai penelitian sebelumnya yang menyatakan *backdoor* yang sering digunakan oleh Peretas (T. Wijayanto & Susilo, 2017). Sampel *backdoor* b374k diunduh dari halaman https://github.com/b374k/b374k.

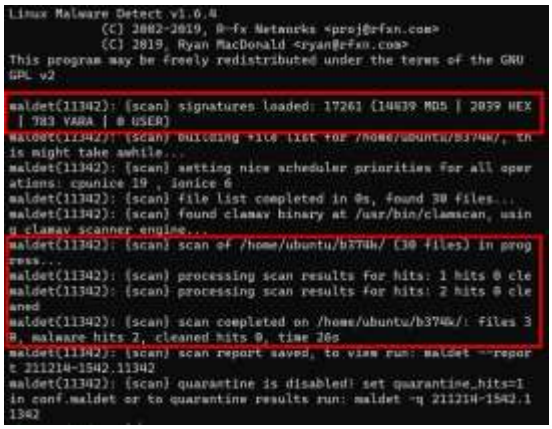
```
1   def Function_Reference(self,disk):
2   try:
3   x = os.listdir(disk)
4   for dirfile in x:
5   assets = os.path.join(disk,dirfile)
6   if os.path.isdir(assets):
7   self.Function_Reference (assets)
8   else:
9   read = open(assets, "rb").read()
10  if "Function_Reference(" in read:
11  print("Potential in ->
    {}").format(assets)
12  except:
13  pass
```

Gambar 3. Format Function Reference

Dalam proses analisis dan deteksi *backdoor* pada OJS *signature-based* dan *static analysis* seperti pada **Error! Reference source not found.**. Untuk pemindai yang *signature-based* menggunakan *Linux Malware Detect (LMD)*, yaitu suatu perangkat lunak yang berfungsi memindai malware pada sistem operasi GNU/Linux berdasarkan basis data *signature* berupa nilai hash dan hex. Sedangkan *static analysis* (analisis statik) menggunakan *function reference* dari bahasa pemrograman Python seperti *Eval, Base64, System, Passthru, Popen, Exec, Shell_exec*, dan *Move_uploaded_file*. *Function reference* merupakan teknik menyederhanakan dalam melakukan analisis statik seperti membaca kode sumber dan mempelajari karakteristik kode (H. Wijayanto et al., 2020). Adapun format kode bahasa pemrograman python untuk melakukan analisis statik seperti tampak pada Gambar 3.

Aplikasi yang digunakan adalah LMD dengan sumber basis data *signature* dan hex yang terbaru yaitu sebanyak 17261 signature yang terdiri dari: 14439 MD5, 2039 HEX, 783 YARA, dan 0 USER. Proses pemindaian memanfaatkan aplikasi ClamAV, seperti pada Gambar 4. Pemindaian dilakukan pada ekosistem yang tertutup dan terbatas untuk mengurangi penyalahgunaan dari backdoor. Berkas backdoor diletakan pada direktori /*home/ubuntu/b374k*. Berdasarkan

pemindaian menggunakan LMD terdapat 2 berkas yang berpotensi membahayakan sistem dari 30 berkas yang dipindai. Sedangkan menggunakan analisis statik ditemukan beragam potensial tergantung dari *function reference* yang digunakan. Adapun temuan dari pemindaian LMD dan analisis statik dapat dilihat pada Tabel ***1***.



Gambar 4. Pemindaian LMD

Tabel 1. Hasil Pemindaian dan Analisis Statik

Proses	Jml	Potensi
LMD	2	/home/ubuntu/b374k/index.php /home/ubuntu/b374k/README.md
Eval	4	/home/ubuntu/b374k/index.php /home/ubuntu/b374k/base/jsPacker.php /home/ubuntu/b374k/base/main.php /home/ubuntu/b374k/module/info.php
Base_64	5	/home/ubuntu/b374k/index.php /home/ubuntu/b374k/base/base.php /home/ubuntu/b374k/base/main.php /home/ubuntu/b374k/module/mail.php /home/ubuntu/b374k/module/convert.php
System	0	
Passthru	1	/home/ubuntu/b374k/base/main.php
Popen	1	/home/ubuntu/b374k/base/main.php
Exec	4	/home/ubuntu/b374k/base/main.js /home/ubuntu/b374k/base/jsPacker.php /home/ubuntu/b374k/base/main.php /home/ubuntu/b374k/module/database.php
Shell_exec	1	/home/ubuntu/b374k/base/main.php
Move_uploaded_file	1	/home/ubuntu/b374k/base/base.php

4. KESIMPULAN

Berdasarkan pengujian secara manual menggunakan LMD dan analisis statik hasilnya saling melengkapi. Artinya proses pemindaian menggunakan LMD akan lebih komprehensif jika digabungkan dengan analisis statik. Hal ini ditunjukan bahwa jumlah temuan dari LMD dan analisis statik berbeda. Bahkan analisis statik menggunakan setiap *function reference* juga berbeda jumlah dan berkas yang berpontensi membahayakan juga berbeda. Pendekatan dalam menganalisis dan mendeteksi pontesi serangan siber berupa penyisipan berkas backdoor sebaiknya menggunakan *hybrid methods* supaya dapat menghasilkan hasil yang lebih komprehensif. Penelitian ini masih terdapat kelemahan dalam mengintergrasikan dua metode yaitu pemindaian menggunakan LMD dan analisis statik. Harapanya kedepan dapat dikembangkan kembali suatu alat bantu yang terintegrasi menggunakan *hybrid methods*.

UCAPAN TERIMA KASIH

Penelitian ini didukung oleh Universitas Jenderal Achmad Yani Yogykarta pada program Hibah Penelitian Dosen Internal. Universitas Jenderal Achmad Yani Yogyakarta bekerjasama dengan PT Widya Adijaya Nusantara (Widya Security) berserta Komunitas LowSec Indonesia menjalin kerjasama dalam penelitian ini sebagai wujud implementasi Kampus Merdeka.

DAFTAR PUSTAKA

Asosiasi Penyelenggara Jasa Internet Indonesia. (2019). *Penetrasi dan Perilaku Pengguna Internet Indonesia 2018*.

Badan Siber dan Sandi Negara, & Indonesia Honeynet Project. (2020). *Laporan Tahunan 2020*.

Hariyadi, D., Kusuma, D. P. I., Maulida, N. H., & Ma'rifat, M. (2020). Evaluasi Potensi Celah Keamanan SQL Injection Menggunakan Nearest Neighbor pada Security-Software Development Life Cycle. *Jurnal Repositor*, *2*(9), 1273–1280. https://doi.org/10.22219/repositor.v2i9.999

Hariyadi, D., & Nastiti, F. E. (2021). Analisis Keamanan Sistem Informasi Menggunakan Sudomy dan OWASP ZAP di Universitas Duta Bangsa Surakarta. *Jurnal Komtika (Komputasi Dan Informatika)*, *5*(1), 35–42.

Lockheed Martin Corporation. (2015). *Seven Ways to Apply the Cyber Kill Chain with a Threat Intelligence Platform*.

Martinez-Caro, J. M., Aledo-Hernandez, A. J., Guillen-Perez, A., Sanchez-Iborra, R., &

Cano, M. D. (2018). A comparative study of web content management systems. *Information (Switzerland)*, *9*(2). https://doi.org/10.3390/info9020027

Microsoft Corporation. (2013). *Microsoft Security Development Lifecycle*.

Miniwatts Marketing Group. (2020). *World Internet Users Statistics and 2021 World Population Stats*. https://www.internetworldstats.com/stats.htm

Risdanto, B. (2014). *Pengembangan E-Learning Berbasis Web Menggunakan CMS (Content Management System) Wordpress di SMA Negeri 1 Kota Magelang*. Universitas Negeri Yogyakarta.

Wijayanto, H., Hariyadi, D., Muhammad, A. H., Nusantara, S., Program,), Informasi, S. T., Achmad, J., & Yogyakarta, Y. (2020). Analisis Penyalahgunaan Data Pribadi Dalam Aplikasi Fintech Ilegal Dengan Metode Hibrid. *Jurnal Ilmiah Sinus*, *18*(1), 2548–4028. https://doi.org/10.30646/sinus.v18i1.433

Wijayanto, T., & Susilo, A. (2017). Implementasi Backdoor Scanner Tool Menggunakan Metode Carving File Pada Server Codepolitan. *I-STATEMENT: Information System and Technology Management*, *3*(2). http://journal.esqbs.ac.id/index.php/I-STATEMENT/article/view/64/66