SNESTIK

Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika

https://ejurnal.itats.ac.id/snestik dan https://snestik.itats.ac.id

Informasi Pelaksanaan :
SNESTIK II - Surabaya, 26 Maret 2022
Ruang Seminar Gedung A, Kampus Institut Teknologi Adhi Tama Surabaya

Informasi Artikel:
DOI : 10.31284/p.snestik.2022.2824

Prosiding ISSN 2775-5126

Fakultas Teknik Elektro dan Teknologi Informasi-Institut Teknologi Adhi Tama Surabaya
Gedung A-ITATS, Jl. Arief Rachman Hakim 100 Surabaya 60117 Telp. (031) 5945043
Email : snestik@itats.ac.id

Penerapan Program Malware Untuk Pengambilan Data User Pada Sistem Operasi Windows Menggunakan Eksploitasi User Account Control (UAC)

Achmad Basyari Mushthofa[1], Danang Haryo Sulaksono[2], Citra Nurina Prabiantissa[3], Gusti Eka Yuliastuti[4], Septiyawan Rosetya Wardhana[5]

Institut Teknologi Adhi Tama Surabaya[1,2,3,4,5]
e-mail: citranurina@itats.ac.id

ABSTRACT

In today's digital era, some people frequently ignore information security. Basically, information security has become an important issue in computer security systems. Malware still becomes one of the most serious threats to an operating system's security. It has a wide variety of roles and behaviors depending on the purpose of malware development. Therefore, the researcher aimed at developing a malware that functions for collecting the socket data and making it undetected by exploiting UAC (User Account Control) on the Windows 7 operating system. By utilizing the slot in one of windows programs, the exploitation entered the operating system and took payload containing a script to catch the socket data. This malware run on the background continuously and retrieved the header information from socket transactions. The results of 10 data capture tests demonstrated that 70% captured data derived from HTTP protocol and by resource usage, the improvement increased less than 1% when the virus was running on the operating system. Accordingly, although the malware did not consume a lot of resources, its ability to capture header data was very good. As a result, the malware could not be detected by most antivirus machines.

Keywords: *User Account Control (UAC); Malware; Socket; Python.*

ABSTRAK

Pada era digital saat ini, keamanan informasi sering kali diabaikan oleh beberapa orang. Keamanan informasi telah menjadi masalah penting dalam sistem keamanan komputer. Seiring dengan pesatnya perkembangan pengguna teknologi komputer dan jaringan tersebut, muncul pihak-pihak tidak bertanggung

jawab yang secara sengaja mengambil informasi data yang bersifat privasi tersebut dan digunakan untuk hal-hal yang merugikan korbannya. *Malware* masih menjadi salah satu permasalahan keamanan bagi sebuah sistem operasi. *Malware* memiliki beragam fungsi dan kebiasaan yang berbeda bergantung pada tujuan malware tersebut dibuat. Aktivitas pencurian data menggunakan *malware* masih sangat memungkinkan dilakukan pada saat ini. Mengacu pada uraian tersebut, penulis ingin mengembangkan sebuah malware yang berfungsi sebagai alat untuk melakukan pengumpulan data *socket* dan membuatnya bekerja secara tidak terdeteksi dengan metode eksploitasi UAC (*User Account Control*) pada sistem operasi *windows 7* menggunakan bahasa pemrograman python3. Data dari hasil tangkapan tersebut berisi informasi *response header* seperti *host, user-agent, web directory, cookie* dan lain-lain pada website yang berasal dari aliran data *socket* ketika *user* melakukan aktivitas dengan jaringannya. Tujuan dari penelitian ini adalah pengambilan data aktivitas *user* yang nantinya data tersebut dapat bermanfaat dan dipergunakan untuk tujuan pengembangan pengolahan data.

Kata kunci: *User Account Control (UAC)*; *Malware*; *Socket*; Python.

PENDAHULUAN

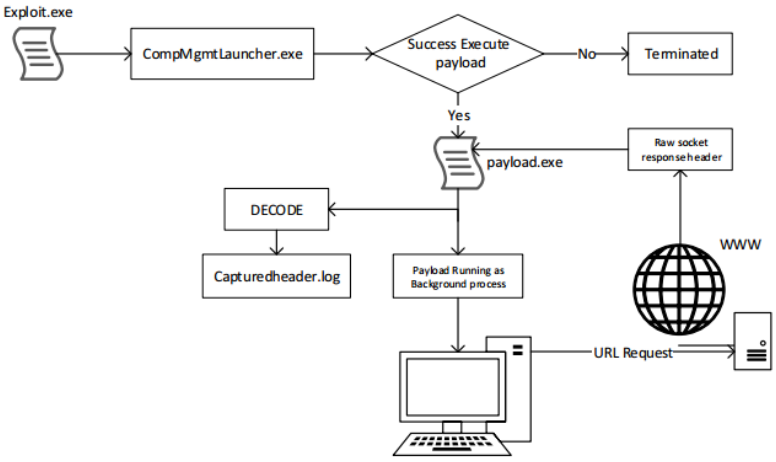
Keamanan informasi sudah menjadi masalah utama dalam sistem keamanan komputer mengingat pesatnya perkembangan teknologi komputer saat ini[7]. Tingkat keamanannya pada sistem informasi sangat bergantung pada sistem operasi. Dalam lingkungan jaringan, keamanan sistem operasi memainkan peran penting dalam keamanan sistem komputer [5]. Sejalan dengan perkembangan yang pesat ini, banyak sekali informasi yang beredar di Internet. Semua ini membuatnya menarik untuk mengumpulkan data tentang kebiasaan dan minat pengguna [2].

Data Mining merupakan cara yang efektif untuk mengetahui adanya serangkaian pola informasi dari sejumlah besar data yang ada[8]. Pola informasi yang didapat akan menjadi sangat berarti apabila bersifat *implicit* (belum diketahui sebelumnya), dan dapat bermanfaat [4]. Mengacu dalam penelitian sebelumnya mengenai analisa pola *HTTP Header Request Messages*, sebuah data trafik header adalah sasaran favorit hacker buat melakukan bermacam suntik malware buat mengekstrak keterangan sensitif user yang dipergunakan untuk kepentingan pribadi juga bisnis[3]. Berbeda jika menggunakan penambangan data tradisional yg memakai perpaduan data statis, terdapat beberapa tantangan buat penambangan aliran data, misalnya, keterbatasan ruang penyimpanan, performa tools, *reaction time*, dan yang lainnya[6].

Berdasarkan uraian di atas, penulis ingin melakukan pengembangan dari sebuah malware yang berfungsi sebagai alat untuk melakukan pengumpulan data dan membuatnya bekerja secara tidak terdeteksi dengan metode eksploitasi User Account Control (UAC) pada sistem operasi windows menggunakan bahasa pemrograman python3. Python sendiri adalah bahasa pemrograman yang sangat cepat dan akurat yang digunakan untuk skrip exploit, merekayasa paket jaringan, dan memanipulasi protokol [1].

METODE

Penelitian ini merupakan pengembangan *program* yang akan digunakan sebagai alat buntuk mengumpulkan *data header* dalam transaksi TCP *socket* antara *client* menggunakan *web server*. Modul socket dalam python membutuhkan sebuah hak akses administrator agar dapat berjalan dalam sebuah sistem operasi. Alat ini nantinya akan berjalan dalam sistem operasi menggunakan memakai hak akses administrator kemudian melewati verifikasi keamanan pada UAC dalam windows dan membuatnya berjalan secara terus menerus dalam background tanpa disadari oleh pengguna. Berikut ini merupakan cara kerja *exploit tool*:

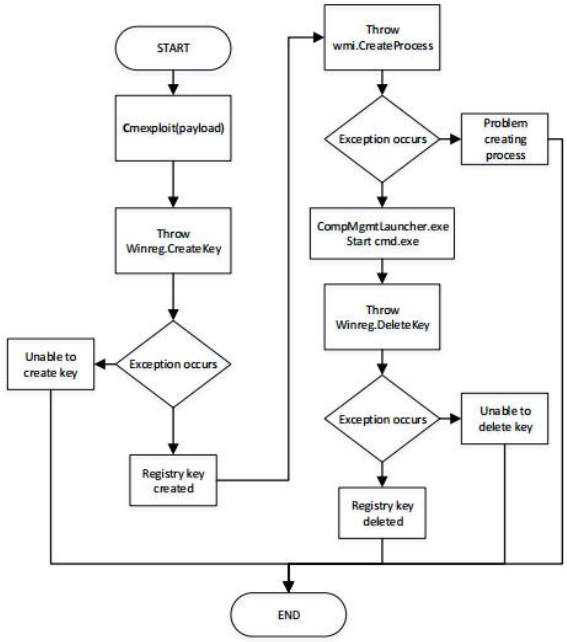


Gambar 1. Cara Kerja *Exploit Tool*

Gambar 1 menjelaskan tentang cara kerja *Exploit Tool* yaitu pada saat program akan melakukan penambahan konfigurasi *registry* baru untuk menciptakan sebuah tiruan perintah manajemen komputer. Kemudian perintah tadi dipergunakan untuk eksekusi *payload* melalui *command prompt administrator* berdasarkan *computer management*. Ketika *payload* berhasil dijalankan maka secara otomatis program akan berjalan dan menjalankan tugasnya di sistem operasi memakai hak akses administrator.

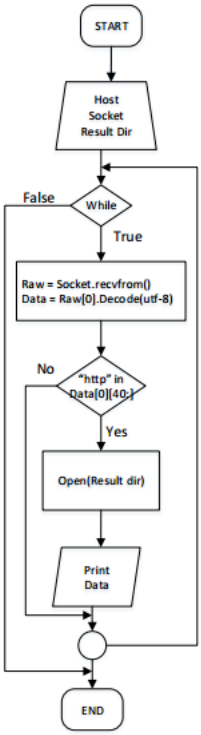
Perancangan Program Malware

Pengembangan yang dilakukan pada penelitian ini terdiri dari dua proses, yaitu tahap pertama melakukan pengembangan pada *exploit* dan tahap kedua melakukan pengembangan pada payload.



Gambar 2. *Flowchart Exploit*

Terdapat 3 tahap dalam flowchart exploit di atas, yaitu melakukan pembuatan *registry* key baru, melakukan pembuatan melalui *CompMgmtLauncher.exe*, dan melakukan penghapusan *registry* yang baru saja ditambahkan.



Gambar 3. *Flowchart Payload*

Gambar 3 menunjukkan tahap payload dimana tahap ini melakukan ekstraksi *HTTP header* pada s*ocket.* Proses pertama yang dilakukan yaitu melakukan deklarasi *hostname*, *variable result directory*, dan *socket function*. Setelah itu, inisialiasi perulangan untuk mendapatkan aliran data socket terus menerus dan disimpan pada *variable Raw.* Dari data *raw* yang didapat, dilakukan decode untuk pengujian pencarian *string http* pada data raw. Ketika *string http* ditemukan maka data langsung disimpan dan dicetak pada *result directory*.

HASIL DAN PEMBAHASAN

Pengujian Pengambilan Data

Pengujian ini dilakukan pada 10 website yang berbeda, menggunakan protokol HTTP atau HTTPS. Setiap website akan diakses secara keseluruhan dari halaman awal sampai sub direktori. Dari hasil tersebut akan diketahui data apa saja yang dapat diambil. Berikut ini merupakan hasil data tangkapan *header*:

Tabel 1. Data Tangkapan *Header*

Website	**Protokol**	***Host***	***Website Directory***	***User Agent***	***Cookie***
www.findglocal.com	HTTP	Ya	Ya	Ya	Ya
www.nachitape.com	HTTP	Ya	Ya	Ya	Ya
www.adidas.co.id	HTTPS	Tidak	Tidak	Ya	Tidak
www.shopieparis.com	HTTP	Ya	Tidak	Ya	Ya

www.uniqlo.com	HTTPS	Ya	Tidak	Ya	Ya
www.kaospremium.com	HTTP	Ya	Ya	Ya	Ya
sid.sidoarjokab.go.id	HTTP	Ya	Ya	Ya	Ya
pn-jakartapusat.go.id	HTTP	Ya	Ya	Ya	Ya
www.dpr.go.id	HTTPS	Tidak	Tidak	Tidak	Tidak
www.itats.ac.id	HTTPS	Tidak	Tidak	Tidak	Tidak

Dari hasil pengujian 10 *website* yang tertera pada tabel 1, didapatkan hasil 6 *website* yang menggunakan protokol HTTP memiliki persentase 100% dalam penangkapan data. Sedangkan untuk *website* lain memiliki presentase 25% di mana terdapat 1 *website* yang menggunakan HTTPS dapat ditangkap oleh *malware*.

Pengujian Performa Pada Sistem Operasi

Pengujian performa pada sistem operasi dilakukan untuk mengetahui pengaruh *malware* pada sumber daya sistem operasi. Pengujian akan dibagi menjadi 4 fase yaitu yang pertama dalam keadaan desktop diam atau tidak menjalankan apapun, fase ketika menjalankan *browser* saja, fase ketika menjalankan *malware* saja dan fase ketika menjalankan *browser* dan *malware* secara bersamaan. Hasil pengujian terdapat pada tabel 2 berikut ini :

Tabel 2. Tabel Pengujian Performa

Proses	**RAM**	**CPU**
Dekstop	23-25%	2-5%
Browser (Firefox)	35-40%	5-10%
Malware (Windows eye.exe)	23-26%	2-5%
Browser dan *Malware*	35-41%	5-10%

Pada tabel 2 menunjukkan hasil pengujian performa pada sistem operasi dari fase pengujian yang berbeda, dimana terdapat kemiripan hasil persentase antara menjalankan dekstop dan malware sebanyak RAM 23-26% dan CPU 2-5%. Selain itu juga pada saat menjalankan browser saja dan browser + malware berada pada persentase RAM 35-41% dan CPU 5-10%. Dari hasil ini, menjalankan browser + malware lebih berat jika dibandingkan saat menjalankan dekstop dan malware saja, tetapi dari persentase dapat dibuktikan malware tidak terlalu membebani kinerja dari sistem operasi karena tidak membutuhkan sumber daya yang besar.

Pengujian Deteksi

Pengujian Deteksi dilakukan adalah untuk mengetahui apakah program yang ditulis terdeteksi sebagai skrip berbahaya atau tidak. Pengujian dilakukan dengan cara mengunggah file eksploit dan payload ke situs multi scanning antivirus *virustotal.com* untuk mengetahui berapa banyak program antivirus yang mendeteksi file tersebut sebagai program berbahaya atau tidak. Dari hasil *scanning file* eksploit pada *website virustotal*. Terdapat 7 dari total 68 *software* antivirus menandai bahwa file tersebut masuk kedalam kategori *malicious* dan *trojan*. Sedangkan dari hasil *scan* untuk file payload. Terdapat 17 dari total 68 software antivirus yang mendeteksi bawa skrip dari payload ini termasuk dalam kategori *malicious* dan *trojan.*

KESIMPULAN

Dari beberapa pengujian yang telah dilakukan, didapatkan kesimpulan sebagai berikut:

1. Dari hasil pengujian 10 website dengan protokol HTTP maupun HTTPS, didapatkan hasil bahwa 6 website yang menggunakan protokol HTTP dapat tertangkap sepenuhnya

atau memiliki persentase keberhasilan penangkapan data 100%. Sedangkan pada 4 website lainnya yang menggunakan protokol HTTPS terdapat 1 website yang dapat tertangkap oleh *malware*. Ini menunjukkan bahwa *website* dengan protokol HTTPS masih dapat terambil datanya dengan persentasi 25 %.

2. Sedangkan dari hasil pengamatan performa, kinerja sistem operasi sebelum dan sesudah dijalankannya *malware* ini tidak terlalu menunjukkan perbedaan performa yang signifikan dari perangkat komputer. Ini menandakan bahwa *malware* tidak begitu memiliki pengaruh beban yang besar bagi sumber daya sistem operasi. Bahkan *malware* tidak sampai menggunakan 1 % dari *memory*.
3. Untuk hasil pengujian deteksi antivirus, bisa dikatakan bahwa malware ini masih dalam status tidak terdeteksi. Dikarenakan hanya beberapa mesin *antivirus* saja yang mendeteksi file termasuk dalam kategori trojan dan malicious dari total keseluruhan 68 *software antivirus*.
4. Ekploitasi pada windows 7 masih sangat memungkinkan dilakukan. Eksploitasi UAC menggunakan *computer management* hanya salah satu dari banyaknya celah pada windows 7. Pengembangan *malware* pada penelitian ini sekali lagi hanya bertujuan untuk kegiatan penelitian dan edukasi semata. Diharapkan kedepannya pengguna windows 7 lebih peduli terhadap keamanan sistem operasi.

DAFTAR PUSTAKA

[1] Andress, J., & Linn, R. (2017). Introduction to Python. In *Coding for Penetration Testers*.https://doi.org/10.1016/b978-0-12-805472-7.00002-4

[2] Anitha, V., & Isakki, P. (2016). A survey on predicting *user* behavior based on web server log *file*s in a web usage mining. *2016 International Conference on Computing Technologies and Intelligent Data Engineering, ICCTIDE 2016*. https://doi.org/10.1109/ICCTIDE.2016.7725340

[3] Calzarossa, M. C., & Massari, L. (2014). Analysis of header usage patterns of HTTP request messages. *Proceedings - 16th IEEE International Conference on High Performance Computing and Communications, HPCC 2014, 11th IEEE International Conference on Embedded Software and Systems, ICESS 2014 and 6th International Symposium on Cyberspace Safety and Security*, 847–853. https://doi.org/10.1109/HPCC.2014.146

[4] Rieck, K., Holz, T., & Willems, C. (2008). Detection of Intrusions and Malware, and Vulnerability Assessment. *Detection of Intrusions and* Malware, *and Vulnerability Assessment, July*. https://doi.org/10.1007/978-3-540-70542-0

[5] Xu, T., Yuan, K., Wang, J., Niu, X., & Yang, Y. (2009). A real-time information hiding algorithm based on HTTP protocol. *Proceedings of 2009 IEEE International Conference on Network Infrastructure and Digital Content, IEEE IC-NIDC2009*, 618–622. https://doi.org/10.1109/ICNIDC.2009.5360969

[6] Yile, F. (2016). *Research on the Security Problem in Windows 7 Operating System*. https://doi.org/10.1109/ICMTMA.2016.139

[7] Zheng, X., Li, P., Chu, Z., & Hu, X. (2020). A Survey on Multi-Label Data Stream Classification. *IEEE Access*, *8*(March 2020), 1249–1275. https://doi.org/10.1109/ACCESS.2019.2962059

[8] Zhu, Z., & Peng, G. (2019). An analysis about the defects of windows UAC mechanism. In *Communications in Computer and Information Science* (Vol. 960). Springer Singapore. https://doi.org/10.1007/978-981-13-5913-2_17