STEGOSPLOIT PREVENTION USING STEGANOGRAPHY DESTROYING ALGORITHM

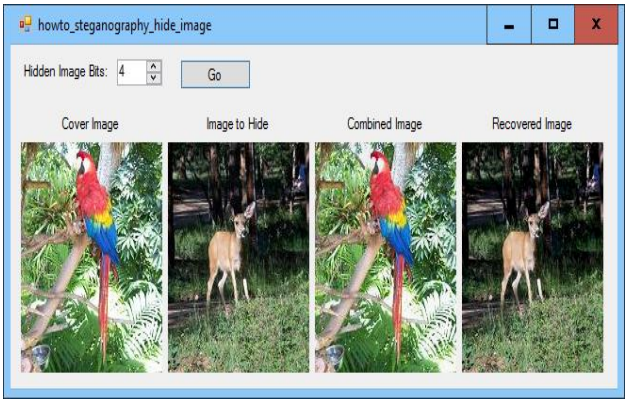
Devin Adam Sanubari (18216006)
Program Studi Sistem dan Teknologi Informasi/Informatika/Teknik Elektro
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 18216006@std.stei.itb.ac.id

***Abstract*—Teknologi terus berkembang seiring berjalannya waktu. Perkembangan ini mempengaruhi banyak sekali hal pada kehidupan manusia, mulai dari cara berkomunikasi hingga mengamankan rahasia. Salah satu metode yang sering digunakan untuk menyembunyikan rahasia adalah Steganografi. Akan tetapi, setiap ada inovasi pasti terdapat orang yang mengeksploitasi inovasi tersebut dan memanfaatkannya untuk hal negatif. Salah satu dari upaya eksploitasi tersebut adalah Stegosploit. Stegosploit merupakan sebuah teknik eksploitasi berbasis Steganografi dengan memanfaatkan celah pada *web browser* untuk menjalankan sebuah program yang bersifat merusak dengan bersembunyi di dalam sebuah gambar yang terlihat tidak berbahaya yang kemudian dijalankan oleh *web* browser secara otomatis tanpa disadari oleh korban. Stegosploit memiliki potensi kerusakan yang sangat besar karena kecepatan penyebaran informasi di media sosial serta karena metode eksploitasi ini sangat sulit dideteksi. Meskipun begitu, Stegosploit bisa dicegah dengan memanfaatkan sebuah algoritma yang dapat menghancurkan informasi Steganografi yang ada pada media yang bersangkutan.**

Keywords— Steganografi, Stegosploit, Destroying Algorithm

I. PENDAHULUAN

Steganografi adalah teknik penyisipan informasi dalam sebuah media. Kata Steganografi sendiri diturunkan dari Bahasa Yunani yang bermakna ”tulisan yang tertutupi” [1]. Cara kerja dari Steganografi adalah dengan menggabungkan dan menyisipkan informasi ke dalam suatu media sehingga informasi tersebut melebur dan tertutupi sepenuhnya oleh media yang bersangkutan dan informasi yang tertanam hanya bisa diambil melalui cara tertentu yang spesifik. Jenis media yang dapat digunakan untuk Steganografi cukup beragam yakni file multimedia seperti citra, audio, ataupun video [2]. Steganografi merupakan teknik penyembunyian yang berada pada ”grey area” atau tidak bisa dikategorikan sifatnya baik ataupun buruk karena pada dasarnya sifat ini tergantung dari pemanfaatan dari teknik tersebut. Steganografi dapat dimanfaatkan untuk merencanakan terorisme, namun di saat yang bersamaan Steganografi juga dapat dimanfaatkan untuk membongkar rencana kejahatan dari suatu kelompok tanpa menimbulkan kecurigaan. Beberapa contoh Steganografi yang pernah dilakukan di masa lalu diantaranya adalah penggunaan tinta transparan berbahan sari buah, urin, atau susu yang dibuat pada masa Romawi Kuno [1]. Tidak hanya itu, tinta transparan yang pernah digunakan pada Perang Dunia II juga merupakan salah satu dari penerapan Steganografi[3]. Di masa kini, algoritma Steganografi ada bermacam-macam. Beberapa teknik Steganografi bahkan dapat digunakan oleh orang awam dengan mudah tanpa harus memahami ilmu Steganografi secara mendalam dengan memanfaatkan tools tertentu yang beredar di internet secara cuma-cuma. Tools ini pada dasarnya akan memanfaatkan suatu algoritma yang ada untuk penyembunyian dan pengambilan informasi [4][5]. Teknik Steganografi yang sering digunakan untuk tools secara umum adalah Least Significant Bit.



Gambar 1 Hasil Penyembunyian Gambar oleh Steganografi (sumber [6])

Least Significant Bit (LSB) adalah algoritma yang memanfaatkan bit paling kanan pada citra dan mengubahnya dengan informasi yang ingin disembunyikan. Karena bit paling kanan merupakan bit yang paling tidak signifikan dalam sebuah data, perubahannya tidak akan mempengaruhi gambar secara keseluruhan dan perubahan ini sama sekali tidak dapat dideteksi oleh mata manusia apabila gambar tersebut memiliki resolusi tinggi karena jangkauan bitnya yang sangat luas. Akan tetapi, pengubahan ini tidak dapat diterapkan untuk gambar monokrom hitam dan putih atau gambar greyscale yang memiliki resolusi yang rendah karena perubahannya akan sangat terlihat[7].

Stegosploit, singkatan dari kata Steganografi dan Exploit, merupakan penerapan Steganografi yang memiliki kemampuan untuk mengubah seluruh eksploitasi yang ada pada browser berbasis JavaScript dalam sebuah citra berformat PNG atau JPG [8]. Cara kerja dari Stegosploit adalah dengan menyisipkan sebuah program dalam sebuah citra yang kemudian akan berjalan secara otomatis apabila citra tersebut dibuka. Stegosploit memiliki potensi yang sangat besar untuk meciptakan kerusakan karena teknik ini sulit dideteksi dan dapat menyamar menjadi sebuah citra yang terlihat tidak berbahaya, namun berpotensi untuk melakukan kerusakan terhadap korban apabila program yang disisipkan berupa eksploitasi. Meskipun Stegosploit mungkin tidak bekerja pada browser yang telah diupdate dan dipatch sehingga keamanannya lebih kuat, ditemukan bahwa hanya sedikit pengguna yang melakukan update pada aplikasi dan operating system-nya secara rutin [7].

II. Rumusan Masalah

Berikut adalah rumusan masalah dari makalah ini.

1. Bagaimana Stegosploit berpotensi untuk menimbulkan kerusakan pada korban?
2. Bagaimana algoritma *Steganography Destroyer* dapat mencegah Stegosploit?

.

III. Tujuan Penulisan

Berikut adalah tujuan dari penulisan makalah ini.

1. Memahami bagaimana Stegosploit berpotensi untuk menimbulkan kerusakan pada korban.
2. Memahami konsep pencegahan Stegosploit menggunakan algoritma *Steganography Destroyer.*

IV. Related Works

Berikut adalah daftar jurnal yang berkaitan dengan topik makalah ini.

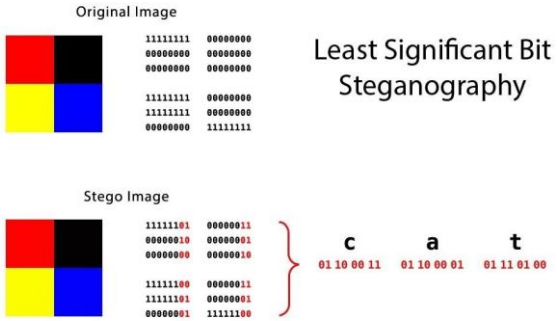
1. Nagham et al. [1] menjelaskan terkait Steganografi secara umum
2. Cox et al. [2] menjelaskan mengenai *digital watermarking* dan Steganografi
3. Cheddad et al. [4] mengadakan survei serta analisis mengenai metode Steganografi
4. Dumitrescu et al. [5] menjelaskan teknik-teknik Steganografi
5. Vaidya et al. [7] berupaya membangun sebuah sistem untuk mendeteksi serangan Stegosploit
6. Corley et al. [11] menjelaskan bagaimana Deep Digital Steganography Purifier (DDSP) dapat membersihkan konten steganografis pada gambar dengan memanfaatkan Generative Adversarial Network (GAN)

Dari keenam jurnal yang telah dicantumkan di atas, satu jurnal membahas tentang pencegahan eksploitasi dari Stegosploit yakni jurnal dari Vaidya et al. [7], satu jurnal membahas pembersihan gambar menggunakan DDSP yakni jurnal dari Corley et al. [11], sedangkan jurnal yang lain membahas secara mendalam mengenai Steganografi.

V. Pembahasan

A. Cara Kerja Steganografi LSB

Salah satu teknik Steganografi yang paling dasar adalah Steganografi LSB. Sesuai namanya, teknik Steganografi ini memanfaatkan bit yang paling tidak signifikan atau *least significant bit*. Citra digital tersusun dari suatu elemen terkecil yang disebut dengan pixel. Pixel dalam citra menyimpan informasi berupa warna yang disajikan dalam bilangan biner. Jumlah pixel dalam citra menunjukkan jumlah representasi informasi dari citra yang bersangkutan. Semakin banyak pixel dalam sebuah citra berarti lebih banyak representasi informasi, sehingga dapat menampakkan representasi gambar asli dengan lebih baik. Begitu pula sebaliknya, semakin sedikit pixel dalam citra berarti lebih sedikit representasi informasi sehingga gambar yang ditampakkan akan lebih buram atau pixelated. Steganografi LSB melibatkan pengubahan data pixel yang terdapat pada citra melalui penyisipan.



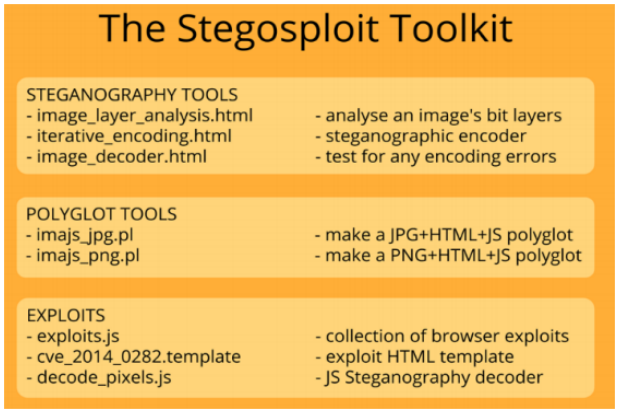
Gambar 2 Contoh Steganografi LSB (sumber [9])

Informasi akan disisipkan pada *least significant bit* (LSB), yakni bit yang terletak di paling kanan. Dalam representasi bilangan biner, bit paling kanan merupakan faktor pengubah nilai yang paling kecil sehingga pada gambar berkualitas tinggi yang memiliki range warna yang luas, perubahan ini tidak akan bisa terlihat oleh mata telanjang. Data akan disisipkan pada pixel pada citra dengan ketentuan dan urutan tertentu. Pada ilustrasi yang diberikan di atas ditunjukkan dua citra, citra asli (gambar atas) dan citra yang sudah disisipi oleh informasi menggunakan Steganografi LSB(gambar bawah). Terlihat bahwa setelah disisipkan informasi berupa tulisan ”cat”, citra tidak mengalami perubahan signifikan yang dapat terlihat oleh mata.

B. Cara Kerja Stegosploit

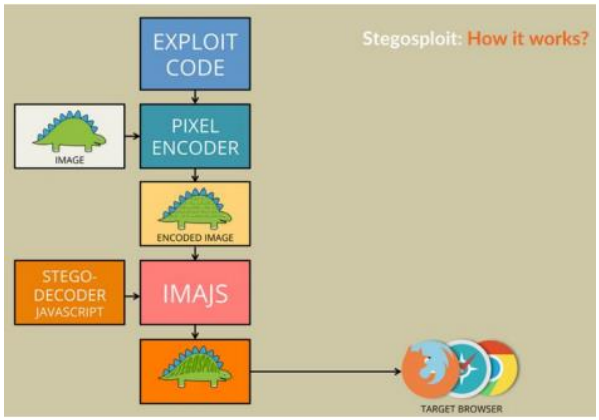
Stegosploit merupakan penerapan lain dari Steganografi. Akan tetapi, pada Stegosploit yang disisipkan pada citra adalah sebuah kode atau program eksploitasi berbentuk JavaScript yang dapat dibuka dan dijalankan di web browser. Stegosploit sangat berbahaya khususnya apabila disebarkan melalui internet yang secara umum memanfaatkan media gambar untuk menyebarkan informasi.

Sebelum menjelaskan proses Stegosploitnya, akan dijelaskan terlebih dahulu toolkit dari Stegosploit.



Gambar 3 Memahami Toolkit dari Stegosploit (sumber [7])

Dijelaskan oleh Vaidya et al.[7], ilustrasi yang disediakan di atas menggambarkan tools dari toolkit Stegosploit. Iterative_encoding.html digunakan untuk melakukan encoding pada exploit code web browser dalam citra secara Steganografis, image¬_decoder.html digunakan untuk mendeteksi error yang memungkinkan pada citra, imajs_jpg dan imajs_png.pl adalah perl script yang membuat citra yang terenkode menjadi citra poliglot menggunakan auto decoder-scritp untuk file jpg dan png secara terpisah. Decode_pixels.js adalah script javascript yang dapat mengeksekusi citra poliglot secara otomatis saat sedang loading. Exploits.js adalah kumpulan eksploitasi browser dan cve_2014_0282.template adalah contoh eksploitasi untuk CVE-2014-0282. CVE-2014-0282 merupakan vulnerabilitas dari IE Use-After-Free dalam Internet Explorer milik Microsoft versi 6 hingga 11. CVE-2014-0282 merupakan vulnerabilitas Internet Explorer terhadap Memory Coorruption, yang mana membuat penyerang bisa mengeksekusi arbitrary code melalui sebuah website khusus dari jauh.



Gambar 4 Proses Pembuatan Citra Poliglot (sumber [7])

Gambar di atas menunjukkan penggunaan toolkit Stegosploit, tepatnya bagaimana pembuatan citra poliglot yang nantinya akan digunakan untuk menyerang browser korban. Dipaparkan oleh Vaidya et al.[7], pertama-tama browser akan mengeksploitasi kode dan citra yang telah dilewatkan suatu encoder citra/pixel buatanan khusus yang kemudian akan mengencode kode eksploitasi browser di dalam citra secara Steganografis dan menciptakan sebuah citra terenkode yang baru. Citra baru ini kemudian akan diteruskan ke library imajs dengan autorun stego decoder-script dan menghasilkan hasil akhir yakni citra poliglot, yang nantinya akan digunakan oleh penyerang untuk menyerang dengan mengirim atau menyebarkannya menggunakan e-mail atau website penyebaran gambar untuk menyerang browser korban.

Berdasarkan pemeriksaan lebih lanjut menggunakan berbagai macam antivirus, ditemukan bahwa penyerangan yang disebabkan oleh Stegosploit sama sekali tidak terdeteksi oleh antivirus. Hal ini disebabkan karena antivirus menganggap bahwa citra poliglot yang diinjeksi oleh program Stegosploit sebagai file gambar biasa dan kode berbahaya yang tersimpan di dalam citra tidak terlihat secara langsung oleh antivirus [7]. Di bawah ini disajikan gambar berupa hasil pemeriksaan file citra poliglot oleh berbagai macam antivirus yang dirangkum oleh Virustotal.

Gambar 5 Hasil Pemeriksaan oleh Virustotal (sumber [7])

Sejauh ini masih belum ada berita mengenai serangan oleh Stegosploit, akan tetapi tidak menutup kemungkin bahwa Stegosploit merupakan salah satu bentuk serangan internet yang memiliki potensi bahaya yang sangat besar. Oleh sebab itu, pengguna harus tetap waspada dalam berinternet dan terhadap serangan Stegosploit.

C. *Deep Digital Steganography Purifier* (DDSP) sebagai *Steganography Destroying Algorithm*

Karena sangat sulitnya pendeteksian serta berbahayanya potensi dampak eksploitasi oleh Stegosploit, berbagai upaya dilakukan untuk mencegah terjadinya eksploitasi. Salah satu upaya pengembangan yang dilakukan adalah dengan membentuk metode Steganalysis, teknik untuk mengidentifikasi Steganografi, memanfaatkan algoritma analitik atau statistik untuk mendeteksi Steganografi sederhana seperti LSB [10]. Akan tetapi, metode ini sangat sulit mendeteksi algoritma modern dan memiliki tingkatan false positive yang sangat tinggi saat diterapkan [11].

Isaac et al.[11] mengajukan sebuah solusi untuk membersihkan konten Steganografi yang ada pada citra dengan Deep Digital Steganography Purifier (DDSP). Teknik ini jauh lebih efektif dan lebih mungkin untuk dilakukan jika dibandingkan dengan pengidentifikasian konten Stegosploit yang memiliki kerumitan yang sangat tinggi serta banyaknya kemungkinan yang ada selama perangkaian pesan, yang mana merupakan sumber utama dari masalah false positive yang dimiliki oleh metode ini.

DDSP memanfaatkan Generative Adversarial Network (GAN), yakni sebuah framework yang dapat dilatih untuk menghapus konten Steganografis tanpa merusak kualitas dari citra yang bersangkutan. Penerapan ini didasarkan pada penelitian yang dilakukan oleh Ledig et al. [12] untuk meningkatkan kualitas sebuah citra beresolusi rendah. Penelitian mereka juga menunjukkan bahwa framework GAN dapat menghasilkan sebuah detail tekstur berkualitas tinggi. Dalam proyek Isaac et al.[11], untuk melatih model DDSP, GAN dilatih dengan membuat *generator* menciptakan gambar yang bersih. Gambar-gambar yang bersih ini, bersamaan dengan gambar *cover* asli akan dilewatkan ke *discriminator* yang kemudian dioptimasi untuk membedakan antara gambar yang telah dibersihkan dengan gambar asli.

DDSP menggunakan *autoencoder* yang telah dilatih sebelumnya sebagai *generator network* dalam framework GAN untuk menghapus konten steganografis tanpa merusak kualitas citra. *Autoencoder* yang digunakan terdiri dari jaringan *encoder* dan *decoder*. *Encoder* mempelajari cara mengurangi ukuran gambar sambil menjaga sebanyak mungkin informasi yang diperlukan. Sedangkan *Decoder* kemudian mempelajari bagaimana cara mengatur ukuran gambar secara optimal mendekati ukuran aslinya sambil menghapus konten steganografis dari gambar tersebut. *Encoder* menjadikan gambar dengan steganografi sebagai input. Untuk melatih *autoencoder* ini sendiri, gambar dengan konten steganografis digunakan sebagai input untuk *encoder*. *Encoder* ini kemudian menciptakan gambar yang telah di*encode* yang kemudian diberikan ke *decoder* untuk di*decode* ke ukuran aslinya. Gambar yang telah di*decode* ini kemudian dibandingkan dengan gambar aslinya menggunakan MSE *loss function* dan kemudian disetel ulang menggunakan framework pelatihan GAN. Proses ini diperlukan karena *autoencoder* dilatih untuk mengoptimasi MSE sehingga dapat menimbulkan penurunan kualitas pada citra [11].

Gambar 5 Arsitektur Deep Digital Steganography Purifier (DDSP) (sumber [11])

D. Hasil Percobaan Pembersihan Gambar DDSP

Dalam percobaan yang dilakukan oleh Isaac et al.[11], yakni pembersihan konten steganografis pada gambar, gambar dianalisis dan dibandingkan kualitasnya dengan hasil dari algoritma steganografi lainnya. Pada percobaan ini DDSP dibandingkan dengan metode *Bicubic Interpolation*, *Denoising Wavelet Filter*, dan *Autoencoder*. Sedangkan data-data yang digunakan sebagai perbandingan diantaranya adalah *Mean Squared Error* (MSE), *Peak Signal-to-Noise Ratio* (PSNR), *Structural Similarity Index* (SSIM) [28], and *Universal Quality Index* (UQI)[13].

Dalam percobaan ini, *Bicubic Interpolation* tidak mampu menjaga kualitas gambar dengan baik karena gambar yang dihasilkan oleh metode ini terlihat buram. *Wavelet Denoising Filter* memiliki hasil yang lebih baik dalam menjaga kualitas gambar jika dibandingkan dengan *Bicubic Interpolation* meskipun mengalami sedikit penurunan kualitas gambar.

Autoencoding dapat bekerja dengan baik dalam menjaga kualitas gambar sambil membersihkan konten steganografis, namun DDSP menghasilkan gambar dengan kualitas yang paling bagus [11].

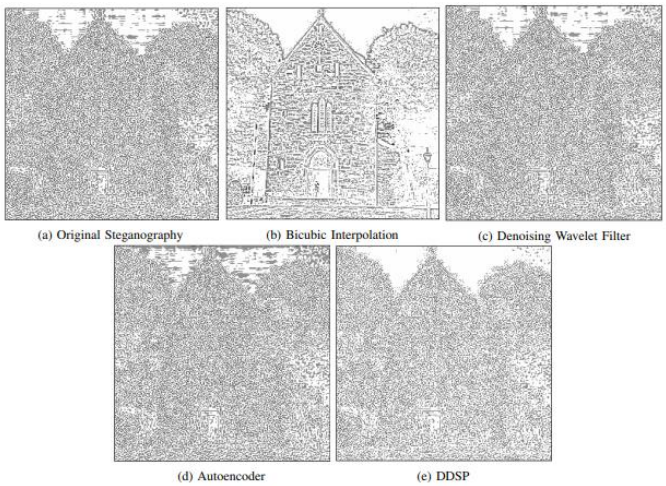
Gambar 6 Hasil Pembersihan oleh Metode-Metode Pembersihan yang bersangkutan (sumber [11])

Model	BER	MSE	PSNR	SSIM	UQI
DDSP	**0.82**	**5.27**	**40.91**	**0.99**	**0.99**
Autoencoder	0.78	5.97	40.37	0.98	**0.99**
Wavelet Filter	0.52	6942.51	9.72	0.19	0.50
Bicubic Inter.	0.53	6767.35	9.82	0.22	0.51

Gambar 7 Perbandingan Kualitas dari Hasil Pembersihan Gambar (sumber [11])

Pada percobaan tersebut, ditemukan bahwa DDSP memiliki *Bit Error Ratio* (BER), PSNR, dan SSIM yang paling besar jika dibandingkan dengan tiga metode lainnya. DDSP juga memiliki MSE yang paling rendah jika dibandingkan dengan metode lainnya, dan UQI yang setara dengan metode *Autoencoder*. Tingginya nilai BER menunjukkan bahwa pada metode DDSP terdapat paling banyak perubahan bit dalam gambar yang diproses. Hal ini menunjukkan bahwa DDSP berhasil menghilangkan konten steganografi yang paling banyak jika dibandingkan dengan metode lainnya. Rendahnya MSE pada DDSP menunjukkan sedikitnya *error* pada pemrosesan gambar dan tingginya PSNR, SSIM, dan UQI menunjukkan bahwa DDSP menghasilkan kualitas gambar yang paling tinggi jika dibandingkan dengan metode lainnya.

Percobaan lain yang dilakukan oleh Isaac et al.[11] untuk membuktikan bahwa DDSP merupakan metode pembersih gambar terbaik adalah dengan melakukan *image substraction*, yakni mengurangkan nilai pixel dari satu gambar dengan gambar lainnya untuk menghasilkan gambar ketiga untuk dijadikan bahan perbandingan. Dalam percobaan ini, gambar asli disubstraksikan dengan hasil pembersihan dari gambar yang bersangkutan menggunakan empat metode yang telah digunakan sebelumnya yakni *Bicubic Interpolation*, *Wavelet Denoising Filter*, *Autoencoder*, dan DDSP.



Gambar 8 Hasil Substraksi dari Gambar Asli oleh Metode-Metode Pembersihan yang Bersangkutan (sumber [11])

Sesuai dengan Gambar 8(a), saat gambar asli disubstraksikan dengan gambar yang telah disisipi Steganografi akan menghasilkan sebuah gambar yang mengandung banyak *noise*. Hal ini merupakan hal yang wajar karena algoritma dari Steganografi sendiri menginjeksikan *noise* dalam jumlah besar ke dalam gambar. Gambar 8(b) menunjukkan bahwa *Bicubic Interpolation* dapat membersihkan sebagian besar dari *noise*, namun seperti yang ditunjukkan oleh percobaan sebelumnya, *Bicubic Interpolation* tidak mampu menjaga kualitas gambar dengan baik. Gambar 8(c) dan 8(d) merupakan hasil dari *Denoising Wavelet Filter* dan *Autoencoder*. Kedua metode ini tidak menghapus *noise* dalam gambar, namun justru menambahkan *noise* untuk menutupi konten Steganografi yang ada pada gambar. Hal ini sangat terlihat dari bagian gambar yang mengandung ruang kosong yang terletak di atas gedung. Dan yang terakhir adalah Gambar 8(e), yang menunjukkan bahwa DDSP tidak menambahkan *noise* dan menghapus *noise* yang ada. Bisa di lihat di ruang kosong di atas gedung bahwa *noise* yang seharusnya ada di area itu dibersihkan dengan baik oleh DDSP. Dari percobaan-percobaan ini dapat disimpulkan bahwa DDSP merupakan metode yang paling baik dan menghasilkan kualitas gambar paling bagus jika dibandingkan dengan metode-metode lainnya.

E. Penerapan DDSP sebagai Pencegah Stegosploit

Stegosploit merupakan eksploitasi *browser* yang berpotensi menimbulkan berbagai macam kerusakan dan tidak terdeteksi oleh antivirus. Hanya dengan membuka citra yang disisipi oleh Stegosploit akan menjalankan program berbahaya yang ada

dalam citra tersebut, sehingga melakukan *scanning* terhadap gambar yang ada dalam *browser* bukanlah ide yang baik karena untuk melakukan *scanning* diperlukan untuk mengunduh file yang bersangkutan.

Penerapan DDSP yang paling efektif adalah dengan memasangkannya di dalam *browser*. DDSP akan berperan sebagai *add-on* yang melakukan pembersihan pada setiap citra yang ada dalam *browser*. Pada umumnya, pengguna awam tidak menggunakan pesan steganografi selama menjelajah internet sehingga pembersihan ini tidak akan mengganggu pengalaman pengguna dalam menggunakan internet. Tidak hanya itu, instalasi DDSP di dalam *browser* lebih efisien jika dibandingkan dengan pembuatan *software* terpisah untuk DDSP secara khusus. Pengguna lebih menyukai hal yang praktis, sehingga instalasi yang hanya perlu membuka *browser* akan jauh lebih disukai dari pada pembuatan *software* yang mengharuskan membuka lebih dari satu perangkat. Tidak hanya itu, penggunaan software secara terpisah juga akan lebih membebani perangkat karena harus menjalankan lebih banyak program di saat yang bersamaan, yang mana mengurangi efisiensi.

VI. Simpulan

Stegosploit merupakan penerapan dari teknik Steganografi yang memiliki potensi kerusakan yang besar. Stegosploit dapat menginfeksi korban hanya dengan membuka citra yang telah disisipi oleh program atau kode Stegosploit, sehingga akan sangat berbahaya apabila disebarkan melalui internet yang banyak menggunakan gambar sebagai sarana penyebaran informasi. Stegosploit juga sulit untuk diidentifikasi dan penyisipannya tak terlihat dan tak terduga, terbukti dari tidak terdeteksinya serangan Stegosploit oleh berbagai macam antivirus. Solusi yang diajukan untuk mencegah serangan Stegosploit adalah dengan memanfaatkan *Steganography Destroying Algorithm* dalam bentuk *Deep Digital Steganography Purifier*. Dengan memanfaatkan DDSP yang ditanamkan ke dalam *browser* atau aplikasi serupa, konten steganografis yang ada pada citra akan dapat dihilangkan dengan mudah sehingga pengguna dapat terlindungi dari Stegosploit.

Video Link at Youtube

https://www.youtube.com/watch?v=8oUNdu-PYH8

Acknowledgment

Penulis ingin mengucapkan terima kasih kepada Bapak Rinaldi Munir selaku dosen mata kuliah Kriptografi dan Koding atas ilmu dan arahan yang telah diberikan.

References

[1] Hamid, Nagham, Yahya, Abid, Ahmad, R. Badlishah, & Al-Qershi, Osamah M. Image Steganography Techniques: An Overview

[2] Cox, I., Miler, M., Bloom, J., Fridrich, J., & Kalker, T. (2007). Digitalwatermarking and steganography. Morgan Kaufman.

[3] Sellars, D. (2011, July). *An introduction to Steganography*. Internet. http://www.cs.uct.ac.za/courses/CS400W/papers99/stego.html.

[4] Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. Signal processing, 90(3), 727-752.

[5] Dumitrescu, D., Stan, I.-M., & Simion, E. (2017). Steganography Techniques.

[6] Hellstrom, J. (2019, January 25). Remember when Steganography was going to be used for good?. PC Perspective. https://pcper.com/2019/01/remember-when-steganography-was-going-to-be-used-for-good/

[7] Vaidya, N., & Rughani, P. (2019). An Efficient Technique to Detect Stegosploit Generated Images on Windows and Linux Subsystem on Windows.

[8] Shah S. (2015), Pastor Manul Laphroaig's, Export–Controlled, Church Newsletter.

[9] *Steganography : hide data in images with Steghide*. (2017, November 12). Technical Foundation. https://technical-foundation.blogspot.com/2017/11/steganography-hide-data-in-images-with.html/

[10] W. Bender, D. Gruhl, N. Morimoto, & A. Lu. (1996). Techniques for Data Hiding. *IBM Systems Journal*, vol. 35, no. 3.4, pp. 313-336.

[11] Corley, I., Lwowski, J., & Hoffman, J. (2019). Destruction of Image Steganography using Generative Adversarial Networks.

[12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et al., "Photo-realistic single image super-resolution using a generative adversarial network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[13] Z. Wang and A. C. Bovik, "A universal image quality index," IEEE signal processing letters, vol. 9, no. 3, pp. 81–84, 2002.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 April 2021

Devin Adam Sanubari (18216006)