Analisis Malware PlasmaRAT dengan Metode Reverse Engineering

M. Hazri
Jurusan Teknik Komputer, Universitas AMIKOM Yogyakarta
e-mail: **hazri.15@students.amikom.ac.id**

Abstrak

Software untuk me-remote sistem dari jarak jauh sangat memudahkan pengguna khususnya untuk perusahaan besar. Akan tetapi tidak sedikit dari software tersebut menyisipkan malware didalam program yang mereka buat yang kemudian dimanfaatkan untuk kepentingan pribadi maupun kelompok. Penting untuk melakukan analisis terhadap program remote sistem tersebut untuk mengetahui hal tidak normal apa yang dilakukan pada sistem ketika program tersebut berjalan. Dengan metode analisis dan tool yang bisa mengekstrak program tersebut. salah satu metode yang bisa digunakan adalah reverse engineering. Sebuah teknik yang dapat membuka source code dari suatu program. Plasma RAT adalah salah satu malware dengan kategori trojan. Penelitian ini bertujuan untuk menunjukkan alur proses analisis dan identifikasi malaware plasma RAT dengan metode reverse engineering. Didalam malware Plasma RAT terdapat beberapa program yang ikut berjalan ketika malware ini diaktfikan pada suatu sistem. Malware Plasma RAT juga menggunakan anti-reverse engineering yang mencegah untuk bisa melakukan reverse engineering pada suatu program.

Kata kunci*—3-5 Malware, Remote Acces Trojan dan Reverse Engineering*

1. PENDAHULUAN

Pada saat ini *internet* sudah menjadi kebutuhan pokok bagi masyarakat. Tingginya penggunaan *internet* menciptakan celah keamanan yang bisa merugikan bagi penggunanya. Karena kemudahan yang bisa didapat dari *internet* menciptakan tindak kejahatan di dunia maya. Kejahatan pada dunia maya disebut dengan *cybercrime*[1]. Kejahatan atau serangan yang terjadi di dunia *cyber* pada saat ini beragam. Penyerang menggunakan berbagai teknik untuk bisa mendapatkan informasi yang terdapat pada korban. Salah satu serangan yang terjadi melibatkan program yang sengaja dibuat untuk mencuri informasi atau biasa disebut *malicious software (Malware)*[2]. Motif dari penyerang beragam diantaranya adalah untuk kesenangan atau mencuri data-data atau informasi penting yang ada pada *victim*[3].

Malware dapat berisi kode-kode program berbahaya seperti *virus, trojan, worm, spyware* dan berbagai program berbahaya lainnya yang dapat merusak sebuah sistem komputer[4]. *Malware* dapat menyebar melalui melalui iklan yang muncul pada *website* atau *pop-up* pada *smartphone*[5]. Ketika *malware* berhasil masuk ke sistem komputer, *malware* akan melakukan aktivitas yang tidak bisa di dekteksi oleh sistem dan secara perlahan membuat kinerja komputer melambat[3]. *Malicious software* saat ini terus berkembang atau berevolusi sehingga dibutuhkan *knowledge base* untuk bisa menganalisis cara kerja dari *malware* tersebut[6].

Malware tidak hanya merusak sistem akan tetapi *malware* bisa menjadi program untuk memonitoring dan mengontrol sistem dari jarak jauh. *Malware* jenis ini termasuk kedalam software legal yang biasanya digunakan oleh perusahaan, *malware* jenis ini termasuk kedalam *Remote Administration Tool (RAT)*[7]. Berbeda dengan *remote administration tool (RAT), remote acces trojan (RAT)* merupakan *malware* yang berfungsi untuk mendapatkan akses ke

sistem tanpa diketahui oleh pemiliknya dan mencuri data yang ada pada sistem tersebut[1]. Analisa *malware* dengan metode *Reverse Engineering* adalah salah satu solusi yang bisa digunakan saat ini. *Reverse engineering* digunakan untuk mengekstrak informasi yang ada pada *malware* untuk mengetahui informasi yang tidak diketahui atau disembunyikan[8].

Penelitian yang pernah dilakukan menjadi sumber referensi mengenai penelitian ini diantaranya [9] menjelaskan bagaimana karakteristik dan cara kerja malware yang terdapat pada aplikasi berbasis *android*. Penelitian ini berhasil mengekstrak keseluruhan *source code java* dari tiga aplikasi android (*iCalender.apk, liveprintslivewallpaper.apk* dan *hippo_sample.apk).* Penelitian ini berhasil menemukan *class file* dengan nama *SmsReciver.class (*pada *iCalender.apk)* yang berfungsi untuk mengirim *SMS Premium* ke nomor tertentu tanpa diketahui oleh user. Ditemukannya *source code* yang berisikan *TelephonyManager (*pada *liveprintslivewallpaper.apk)* yang di maksudkan untuk melihat informasi IMEI dan IMSI dari perangkat seluler user. Sedangkan pada aplikasi *hippo_sample.apk* ditemukan *class file* dengan nama *MessageService.class* yang didalamnya terdapat perintah *"sendsms"* yang ditujukan ke nomor "1066156686" dengan teks sms "8", *SMS* tersebut menyebabkan perangkat seluler user berlangganan *sms premium.*

Penelitian [10] mendeteki serangan *zeus malware* pada jaringan dengan metode *live forensic.* Penelitian ini berhasil menemukan barang bukti digital berupa artefak yang menunjukkan karakteristik dan aktivitas dari *zeus malware.* Temuan tersebut hasil dari analisis dengan menggunakan *tools* yang biasa digunakan untuk menganalisa *image memory* bukan *tools* khusus untuk menganalisa *zeus malware.*

Penelitian [11] menggunakan metode *dynamic analysis* untuk menaganlisa *njrat malware* pada sebuah system untuk mengetahui aktivitas dan proses apa saja yang diaktifkan oleh *malware* tersebut. Penelitian ini berhasil menunjukkan dampak dari performa sistem yang ada pada PC setelah terinfeksi *malware* yang seiring berjalannya waktu performa sistem semakin menurun dan kinerja dari sistem melambat.

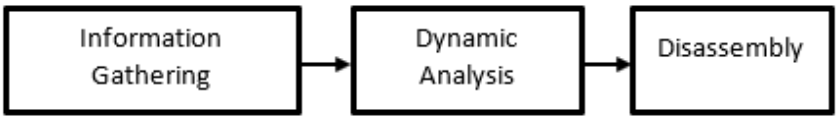
Penelitian [7] menggunakan metode statis dan analisis dinamis untuk menganalisa dan mendeteksi adanya *malware*. Penelitian ini berhasil menganalisa cara kerja dari poison ivy malware yaitu melakukan penambahan dan perubahan terhadap sistem (*WindowsRegistry* dan *file prefatch*) dengan di temukannya berupa artefak kode *string* "*secret_agent*" dan "PIAGENT. EXE_0AEBFBEE.pf", serta upaya dari *malware* tersebut untuk terhubung dengan program iduknya melalui background (tidak kasat mata). penelitian ini juga menunjukkan hasil analisis menggunakan dua metode yang berbeda yaitu metode statis dan dinamis dengan kelebihan dan kekurangan masing-masing.

Penelitian [3] menggunakan metode *reverse engineering* untuk menganalisa *malware Flawed Ammy RAT*. Hasil dari penelitian menunjukan bahwa *malware Flawed Ammyy RAT* bekerja dengan bersembunyi pada aplikasi Ammyy Admin kemudian melakukan koneksi dengan attacker dengan *ip address* 103.208.86.69. *netname ip address* 103.208.86.69 adalah *zappie host*. Perubahan 50 *registry* yang dilakukan *malware* pada sistem yang terinfeksi. Setelah *attacker* terkoneksi dengan korban maka *attacker* dengan mudah melakukan *remote control* tanpa sepengetahuan korban.

Penelitian [12] menggunakan metode analisis statis forensic malware dan *reverse engineering* pada *malware webc2-div*. Hasil dari penelitian ini menunjukkan bahwa *malware* WEBC2DIV merupakan *malicious sofware* terbaik saat melakukan spionasi kegiatan yang dilakukan diantaranya *Phising email, phising login credential*, menyusupkan *backdoor* dan melakukan *remote*. Penelitian ini juga menunjukkan tingkah laku dari malware WEBC2-DIV yang melakukan penanaman nilai pada regedit windows yang beralamat HKEY_CURRENT_USER/Software/Microsoft/Windows/C urrentVersion/Run. Untuk menganalisis *malware* RAT tidak cukup jika hanya dilakukan dengan teknik analisis *basic malware* karena tidak semua *malicious software* RAT berjalan pada sistem[13]. Penelitian ini berfoku pada salah satu sampel *malware* RAT yaitu PlasmaRAT. PlasmaRAT adalah *malware* yang sudah dikenal oleh Sebagian besar perusahaan AV (*antivirus*) di kancah *cybersecurity*, *malware* ini dikenalkan pada awal musim dingin pada tahun 2013.

Tujuan penelitian ini adalah untuk melakukan analisa *malware* PlasmaRAT menggunakan Analisa dinamis dan melakukan *reverse engineering* untuk mengetahui aktivitas apa saja yang bisa dilakukan oleh *malware* tersebut.

2. METODE PENELITIAN



Gambar 1. Alur Penelitian

2.1 Information Gathering

Pada tahap ini dilakukan identifikasi *malware* untuk mengetahui *type* dari *malware* dauntuk mendapatkan informasi lebih rinci mengenai malware tersebut.

2.2 Dynamic Analysis

Tahap dynamic analysis (analisis dinamis) yaitu melakukan pengujian terhadap *malware* dengan cara menjalankan *malware* tersebut pada sebuah sistem untuk mengetahui aktivitas yang dilakukan oleh *malware* tersebut. Pada tahap ini peneliti menggunakan *virtual machine online* untuk mencegah hal-hal yang tidak diinginkan terjadi.

2.3 Disassembly

Pada tahap ini malware akan diekstrak untuk melihat *source code* yang ada pada *malware* tersebut. *Disassembly* adalah teknik *reverse engineering* yang berfungsi untuk menterjemahkan bahasa mesin kedalam bahasa yang lebih mudah dimengerti manusia.

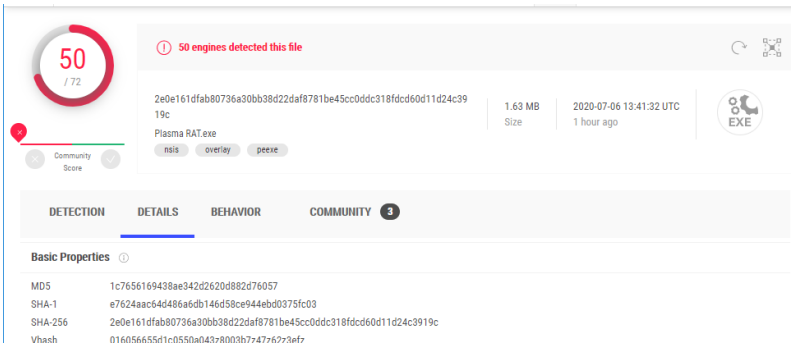
Table I. *Tools* yang digunakan

Tool	Keterangan
Virustotal	Digunakan untuk *iformation gathering malware.*
Hybrid Analysis	*Virtual* mesin *online* untuk menjalankan *malware.*
CFF Explorer	Untuk melakukan proses *disassembly*

3. HASIL DAN PEMBAHASAN

3.1 Information Gathering

Pada tahap ini untuk mendapatkan informasi dari *sample malware* digunakan *tool virustotal. Sample malware* di *upload* ke *website virustotal.*

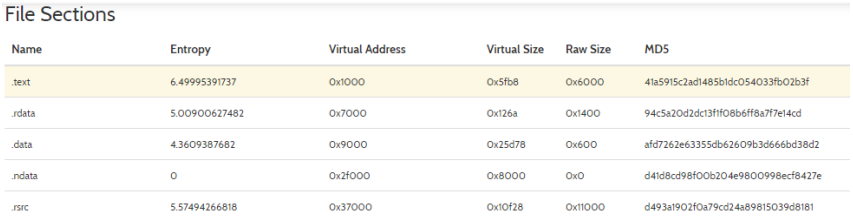


Gambar 2. informasi dari *virustotal*

Pada gambar 2 menunjukkan informasi yang didapatkan dari *website virustotal* mengenai *malware* yang sudah di *upload* adalah dari 72 *antivirus* 50 diantaranya mengkategorikan sebagai *malicious* berbahaya jenis *trojan.* Ukuran *file sample malware* adalah 1.63 *megabyte* dengan *type file executable.*

3.2 Dynamic Analysis

Pada tahap ini analisis dilakukan dengan memanfaatkan *tool online* yaitu *hybrid analysis. Hybrid analysis* merupakan *tool online* yang menyediakan *virtual machine (sandbox)* untuk analisis malware.

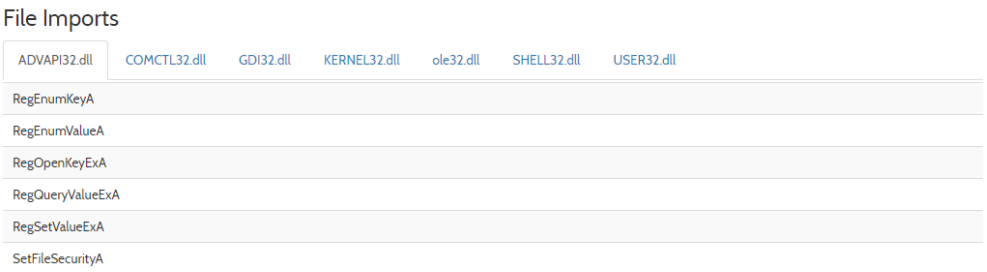


File Sections

Name	Entropy	Virtual Address	Virtual Size	Raw Size	MD5
.text	6.49995391737	0x1000	0x5fb8	0x6000	41a5915c2ad1485b1dc054033fb02b3f
.rdata	5.009000627482	0x7000	0x126a	0x1400	94c5a20d2dc13f1f08b6ff8a7f7e14cd
.data	4.3609387682	0x9000	0x25d78	0x600	afd7262e63355db62609b3d666bd38d2
.ndata	0	0x2f000	0x8000	0x0	d41d8cd98f00b204e9800998ecf8427e
.rsrc	5.57494266818	0x37000	0x10f28	0x11000	d493a1902f0a79cd24a89815039d8181

Gambar 3. Hasil analisis *file selections* pada *hybrid analysis*

Pada gambar 3 menunjukkan ada 5 *file selection* yang terdapat pada *malware plasmaRAT* diantaranya *.txt , .rdata, .data, .ndata* dan *.rsrc.*



File Imports

ADVAPI32.dll | COMCTL32.dll | GDI32.dll | KERNEL32.dll | ole32.dll | SHELL32.dll | USER32.dll

RegEnumKeyA
RegEnumValueA
RegOpenKeyExA
RegQueryValueExA
RegSetValueExA
SetFileSecurityA

Gambar 4. *File imports* pada *malware plasmaRAT*

Merujuk pada gambar 4 ada 7 *file* yang di *imports* oleh *malware plasmaRAT* pada sistem dengan *function* yang berbeda-beda. *File imports* tersebut diuraikan pada Table II.

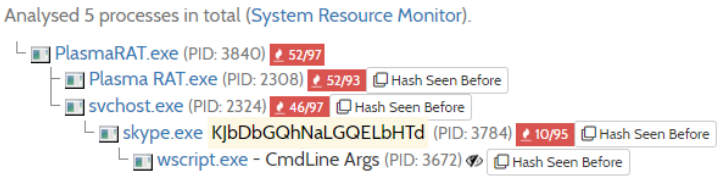
Table II. Uraian *file imports* pada *malware PlasmaRAT*

No	File Imports	Functions

1	ADVAPI32.dll	RegCloseKey RegCreateKeyExA RegDeleteKeyA RegDeleteValueA RegEnumKeyA RegEnumValueA RegOpenKeyExA RegQueryValueExA RegSetValueExA SetFileSecurityA		
2	COMCTL32.dll	ImageList_AddMasked ImageList_Create ImageList_Destroy		
3	GDI32.dll	CreateBrushIndirect CreateFontIndirectA DeleteObject GetDeviceCaps SelectObject SetBkColor SetBkMode SetTextColor		
4	KERNEL32.dll	CloseHandle CompareFileTime CopyFileA CreateDirectoryA CreateFileA CreateProcessA CreateThread DeleteFileA ExitProcess ExpandEnvironmentStringsA FindClose FindFirstFileA FindNextFileA FreeLibrary GetCommandLineA GetCurrentProcess GetFullPathNameA GetLastError GetModuleFileNameA GetModuleHandleA	GetPrivateProfileStringA GetProcAddress GetShortPathNameA GetSystemDirectoryA GetTempFileNameA GetTempPathA GetTickCount GetVersion GetWindowsDirectoryA GlobalAlloc GlobalFree GlobalLock GlobalUnlock lstrcatA lstrcmpA lstrcmpiA lstrcpynA lstrlenA LoadLibraryExA MoveFileA	MulDiv MultiByteToWideChar ReadFile RemoveDirectoryA SearchPathA SetCurrentDirectoryA SetErrorMode SetFileAttributesA SetFilePointer SetFileTime Sleep WaitForSingleObject WriteFile WritePrivateProfileStringA
5	Ole32.dll	CoCreateInstance CoTaskMemFree OleInitialize OleUninitialize		

6	SHELL32.dll	SHBrowseForFolderA ShellExecuteA SHFileOperationA SHGetFileInfoA SHGetPathFromIDListA SHGetSpecialFolderLocation		
7	USER32.dll	AppendMenuA BeginPaint CallWindowProcA CharNextA CharPrevA CheckDlgButton CloseClipboard CreateDialogParamA CreatePopupMenu CreateWindowExA DefWindowProcA DestroyWindow DialogBoxParamA DispatchMessageA DrawTextA EmptyClipboard EnableMenuItem EnableWindow EndDialog EndPaint ExitWindowsEx	FillRect FindWindowExA GetClassInfoA GetClientRect GetDC GetDlgItem GetDlgItemTextA GetMessagePos GetSysColor GetSystemMenu GetSystemMetrics GetWindowLongA GetWindowRect InvalidateRect IsWindow IsWindowEnabled IsWindowVisible LoadBitmapA LoadCursorA LoadImageA MessageBoxIndirectA	OpenClipboard PeekMessageA PostQuitMessage RegisterClassA ScreenToClient SendMessageA SendMessageTimeoutA SetClassLongA SetClipboardData SetCursor SetDlgItemTextA SetForegroundWindow SetTimer SetWindowLongA SetWindowPos SetWindowTextA ShowWindow SystemParametersInfoA TrackPopupMenu wsprintfA

Selain *file imports* terdapat aplikasi lain didalam *sample malware* yang berjalan pada sistem. Hal tersebut ditunjukan pada gambar 5 dibawah ini.

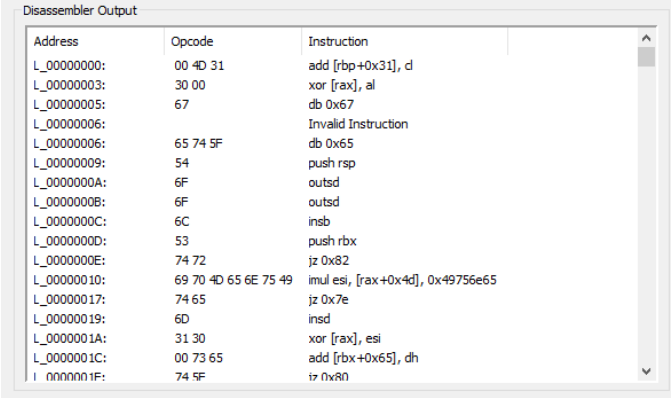


Gambar 5. proses yang berjalan pada sistem

Gambar 5 menunjukkan ada beberapa proses yang dijalankan pada sistem oleh *sample malware* yang dimana proses-proses tersebut dibuat dengan nama *(variable)* yang dikenal oleh sistem, sehingga sistem menganggap bahwa proses tersebut tidak termasuk *malicious* berbahaya.

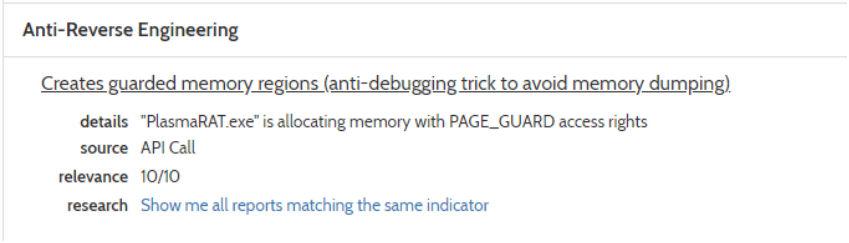
3.3 Disassembly

Pada tahap ini proses *disassembly* akan dilakukan menggunakan *tool CFF explorer*. Setelah *sample malware* dibuka selanjutnya adalah melakukan analisa *functions* atau perintah yang ada pada *sample malware.* Hasil *disassembly* ini ditunjukan pada gambar 6.



Gambar 6. Hasil *disassembly* dari *tool CFF explorer*

Dari hasil *disassembly* yang dilakukan pada *sample malware* hanya menghasilkan berupa kode seperti pada gambar 6. Setelah dilakukan analisa lebih lanjut pada *sample malware* ditemukan adanya *anti-reverse engineering. Anti-reverse engineering* adalah teknik yang memungkinkan untuk mencegah *reverse engineering* pada suatu program. Adanya anti reverse engineering ditunjukan pada gambar 7 dibawah ini.



Gambar 7. *Anti-reverse engineering* pada *malware Plasma RAT*

4. KESIMPULAN

Berdasarkan hasil penelitian dan analisa yang dilakukan dapat ditarik kesimpulan bahwa proses analisis pada *malware plasma RAT* dengan teknik *dynamic analysis* dan reverse engineering dimulai dengan mendapatkan informasi (*information gathering)* dari *tool virustotal*, proses *dynamic analysis* menggunakan virtual machine online (sandbox online) hybrid analysis dan proses reverse engineering menggunakan tool CFF Explorer.

Dynamic analysis menggunakan virtual machine online menunjukkan file imports dan functions yang ada didalam file tersebut dan menunjukkan adanya program lain yang ikut berjalan pada sistem ketika malware tersebut diaktifkan. Kemudian proses reverse engineering yang dilakukan bisa dikatakan tidak berhasil, karena didalam sample malware yang di analisa terdapat anti-reverse engineering yang mencegah untuk dilakukannya proses reverse engineering.

DAFTAR PUSTAKA

[1] P. S. Helode, Dr. K. H. Walse, and Karande M.U., "A Methodology of Malware Analysis, Tools and Technique for windows platform – RAT Analysis," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 4, pp. 8198–8205, 2017, doi: 10.15680/IJIRCCE.2017.

[2] E. Haryanto, "Analisis Forensik Malicious Software Wso Webshell Pada Platform Linux," pp. 1–11, 2015.

[3] T. Pajar Setia, N. Widiyasono, and A. Putra Aldya, "Analysis Malware Flawed Ammyy RAT Dengan Metode Reverse Engineering," *J. Inform. J. Pengemb. IT*, vol. 3, no. 3, pp.

371–379, 2018, doi: 10.30591/jpit.v3i3.1019.

[4] J. Li, D. Gu, and Y. Luo, "Android malware forensics: Reconstruction of malicious events," *Proc. - 32nd IEEE Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2012*, pp. 552–558, 2012, doi: 10.1109/ICDCSW.2012.33.

[5] V. Sahfitri, M. Universitas, B. Darma, D. Universitas, and B. Darma, "Analisis Forensik Malware Pop-Up Ads Iklan Pada Platform Android," no. 03, 2015.

[6] A. H. Muhammad, B. Sugiantoro, A. Luthfi, M. Teknik, I. Universitas, and I. Indonesia, "METODE KLASIFIKASI DAN ANALISIS KARAKTERISTIK MALWARE MENGGUNAKAN KONSEP ONTOLOGI," no. 1, 2004.

[7] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Justindo, J. Sist. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 19–30, 2017, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/1037.

[8] H. A. Nugroho and Y. Prayudi, "Penggunaan Teknik Reverse Engineering Pada Malware Analysis Untuk Identifikasi Serangan Malware," *Knsi*, pp. 27–28, 2014.

[9] R. Novrianda, Y. N. Kunang, and P. . Shaksono, "Analisis Forensik Malware Pada Platform Android," *Konf. Nas. ilmu Komput.*, pp. 141–148, 2014.

[10] A. Kurniawan and Y. Prayudi, "Teknik Live Forensics Pada Aktivitas Zeus Malware Untuk Mendukung Investigasi Malware Forensics," *HADFEX (Hacking Digit. Forensics Expo.*, pp. 1–5, 2014.

[11] D. R. Septiani, N. Widiyasono, and H. Mubarok, "Investigasi Serangan Malware Njrat Pada PC," *J. Edukasi dan Penelit. Inform.*, vol. 2, no. 2, pp. 123–128, 2016, doi: 10.26418/jp.v2i2.16736.

[12] R. Faisal, T. Hidayat, and I. Alam, "Reverse Engineering Analysis Statis Forensic Malware Webc2-Div," vol. 4, no. 1, pp. 15–19, 2018.

[13] A. P. Aldya, N. Widiyasono, and T. P. Setia, "Reverse Engineering untuk Analisis Malware Remote Access Trojan," *J. Edukasi dan Penelit. Inform.*, vol. 5, no. 1, p. 40, 2019, doi: 10.26418/jp.v5i1.28214.