Journal of Informatics, Information System, Software Engineering and Applications

Reverse Engineering Analysis Forensic Malware WEBC2-Div

Raditya Faisal Waliulu^{*1}, Teguh Hidayat Iskandar Alam^{*2}

12Department of Informatics Engineering, Universitas Muhammadiyah Sorong Sorong, Papua Barat, Indonesia

> ¹ raditya@um-sorong.ac.id ² teguhhidayat@gmail.com

accepted on September 24, 2018

Abstract

At this paper focus on Malicious Software also known as Malware APT1 (Advance Persistent Threat) codename WEBC2-DIV the most variants malware has criteria consists of Virus, Worm, Trojan, Adware, Spyware, Backdoor either Rootkit. Although, malware could avoidance scanning antivirus but reverse engineering could be know how dangerous malware infect computer client. Lately, malware attack as a form espionage (cyberwar) one of the most topic on security internet, because of has massive impact. Forensic malware becomes indicator successful user to realized about malware infect. This research about reverse engineering. A few steps there are scanning, suspected packet in network and analysis of malware behavior and disassembler body malware.

Keywords: forensic malware, analysis, advance persistent threat, cyberwar, disassemble, static analysis, dynamic analysis

I. INTRODUCTION

Recently a number of program created for criminal and illegal purpose growth fast. This Program is malware that creates a growing organization, a criminal computer. Definitely, criminal malware take over client's computer and steal personal data, confidential or information of a beneficial nature. this case pressure investigation digital forensic and research security to secure malware attack analysis and use tools that can be relied on beside antivirus.

Today, malware forensics take a part [1]. The aim malware forensic that can identified and analyzed malware which undefined. Many malware created has capable to avoid detection antivirus. Because of that, needs to know analyze malware should be detail about malware capability it self until known impact damage and theft personal data.

Privacy safety, integrity and availability in a real computer system is a challenging task. Increasing amount of system and complex malware between both of them makes secure protection and accurate every system could take time and prone to error.

A discussion of the fundamental challenges and issues/characteristics of malware has been done. Identification of security and privacy issues within this framework are highlighted. Study of the widely used encryption techniques by malware damage in securing sensitive information on cloud is debated. Scope has been set for academicians and researchers. Diverse versions of the encryption techniques surveyed and analyzed to identify harmful or damage for cloud security [2]

II. LITERATURE REVIEW

Malware analysis must be detailed and it take a long time. Malware avoid faced antivirus categorized good one. But, any aspect malware hide from antivirus and it's hard to detected. A few malware forensic tools can show value hidden malware is. In addition, forensic techniques on various tools and plugins more than avoidance analysis techniques. This has become one of the bases for software investigated [3]. Malware analysis one of security computer analyzed malware, learn how and malware's behave. Malware analysis has two method static analysis and dynamic analysis. Analysis static is method disassemble malware without running. But, dynamic analysis is running malware and look for behave itself [4].

Framework or pattern recognition techniques are applied for detection of packed malware binaries. The proposed divided in two phases, first phase it classified packed and non-packed executables. Once an executable is classified as packed, the second phase of classification finds packed benign or packed malware executable. Result framework gain more than 99.9% accuracy in the first phase of classification and 95% accuracy in the second phase of classification [5]. High demand Internet data transfer needs is highly dependent on social factors. because the development of technology is increasingly encouraged to understand the mobility of end user needs. not limited only that Human Resources must also be encouraged to know more about the latest technology updates [6]. At this paper focus on malware forensic, a few malware has typical one of virus, trojan, adware, spyware, backdoor, and rootkit capable attack fast to infect operating system [7].

III. TAXONOMY OF MALICIOUS SOFTWARE

Created malicious software high growth for cyberwar and spionage there are computer virus which might be confused, such as backdoor, worm and etc [5]. According that following paragraphs offer definitions of these types of malicious software and explanations:

A. Malware

Contraction of malicious software. Put simply, malware is any piece of software that was written with the intent of doing harm to data, devices or to people. Theses family of malware, including worms, viruses, Trojan horses, backdoors, bombs and rootkits.

B. A Trojan horse

a program that appears to be legal and executed by victim that gives the attacker unauthorized remote access to a system it can be harmful or advantages by attacker.

C. A virus

Recursive code it can replicates itself. In other words, virus could attach in processes or be harm on computer.

D. A worm

Infect computer needs skill social engineering, does not host or human to propagate. Worm works on file-transport or information-transport features on the system, allowing it to travel unaided.

E. Rootkits

Special tools used to attacker that allows someone to takeover a computer without the computer user/owner knowing about.

F. A backdoor

After take over computer victim's by bypass defense system operating. in could gain unauthorized access and remote.

IV. ANALYSIS

At this section we describe our proposed malware analysis schema research forensic malware host and guest Windows XP SP3. Physic host IP 192.168.56.1 and Guest IP 192.168.56.101 this we use bridge interface, fig 1



Fig. 1. Proposed model forensic malware WEBC2-DIV

Analysis malware, there are two main techniques for analysis malware that are the most commonly used method was static analysis and dynamic analysis. Static analysis is a method of analysis of malware that done without running the malware, so analysis using this method is much more secure than using the method of dynamic analysis. fig 2



Fig. 2. Works analysis forensic malware WEBC2-DIV

Malware WEBC2-DIV running at Guest, at Figure line blue malware running in name Div.exe, fig 3

File Machine View Input D	evices I	Help					
Process Hacker [DARKMESSAGE\DARKMESSAGE-COMOUTER]							
Hacker View Tools Users Help							
🕾 Bafrach 🖄 Ontione 🗱 Find handlas or DLLs 🛹 Sustam information 📃 🗔 😽							
Frocesses Services Network							
Name	PID	CPU	1/O total r	Private by	User name	Description	
System Idle Process	0	97,00		0	NT AUTHORITY\SYSTEM		
E System	4		236 B/s	0	NT AUTHORITY\SYSTEM	NT Kernel & System	
smss.exe	452			168 KB	NT AUTHURITY\SYSTEM	Windows NT Session Manager	
Csrss.exe	656			1,58 MB	NT AUTHORITY\SYSTEM	Client Server Runtime Process	
E s winlogon.exe	680			6,9 MB	NT AUTHORITY\SYSTEM	Windows NT Logon Application	
E E services.exe	1124	2,00	56 B/s	2,96 MB	NT AUTHORITY\SYSTEM	Services and Controller app	
VBoxService	1328			1,36 MB	NT AUTHORITY\SYSTEM	VirtualBox Guest Additions Service	
svchost.exe	1380			3,1 MB	NT AUTHORITY\SYSTEM	Generic Host Process for Win32	
svchost.exe	1496		52 B/s	1,88 MB	NWETWORK SERVICE	Generic Host Process for Win32	
E svchost.exe	1740		218 B/s	12,69 MB	NT AUTHORITY\SYSTEM	Generic Host Process for Win32	
wscntfy.exe	620			660 kB	\DARKMESSAGE-COMC	Windows Security Center Notifica	
svchost.exe	1796			1,35 MB	NVNETWORK SERVICE	Generic Host Process for Win32	
svchost.exe	1848			1,65 MB	NT A\LOCAL SERVICE	Generic Host Process for Win32	
spoolsv.exe	1004		56 B/s	3,24 MB	NT AUTHORITY\SYSTEM	Spooler SubSystem App	
svchost.exe	1644			1,37 MB	NT A\LOCAL SERVICE	Generic Host Process for Win32	
cisvc.exe	1676			1,04 MB	NT AUTHORITY\SYSTEM	Content Index service	
alg.exe	1952			1,25 MB	NT A\LOCAL SERVICE	Application Layer Gateway Service	
lsass.exe	1136		64 B/s	3,93 MB	NT AUTHORITY\SYSTEM	LSA Shell (Export Version)	
DPCs				0			
Interrupts				0			
🖃 😼 explorer.exe	608		56 B/s	16,23 MB	\DARKMESSAGE-COMC	Windows Explorer	
MoxTray.exe	576			1,63 MB	\DARKMESSAGE-COMC	VirtualBox Guest Additions Tray A	
ctfmon.exe	604			0,98 MB	\DARKMESSAGE-COMC	CTF Loader	
E 🛣 clover.exe	636			6,18 MB	\DARKMESSAGE-COMC	Clover	
🖃 😼 explorer.exe	2648			8,55 MB	\DARKMESSAGE-COMC	Windows Explorer	
div.exe	3560			1,47 MB	\DARKMESSAGE-COMC		
🛫 pythonw.exe	784		32 B/s	6,28 MB	\DARKMESSAGE-COMC		
ProcessHacker.exe	3368	1,00		8,85 MB	\DARKMESSAGE-COMC	Process Hacker	
🚯 GoogleCrashHandler.exe	1412			1,91 MB	NT AUTHORITY\SYSTEM	Google Crash Handler	
PU Usage: 3.00% Physical memory: 305	,83 MB (14.	34%) Pi	ocesses: 27				
🎦 Start 🛛 😚 malware - Clover	Pro	cess H	acker [DA				

Fig. 3. Process hacker programs to determine WEBC2-DIV

After div.exe running at guest, wireshark on host trying to suspect through network, string cleartext we get and malware trying to connect to thecrowngolf.org, fig 4.



Fig. 4. Host Excute Wireshark

Step before doing reverse engineering, host to do dissembler at OS Parrot OS and Kernel 3.16-04, fig 5.



Fig. 5. Host disassembler

First of all overview about file div.exe malware using Cutter software. Running cutter and insert div.exe we get information about Hash and Library, at fig 6

OVERVIEW

Info					
File:	arkmessage/Pictures/div.exe	FD:	3	Architecture:	x86
Format:	pe	Base addr:	0	Machine:	i386
Bits:	32	Virtual addr:	True	OS:	windows
Class:	PE32	Canary:	False	Subsystem:	Windows GUI
Mode:	-r-x	Crypto:	False	Stripped:	True
Size:	7168	NX bit:	False	Relocs:	True
Type:	EXEC (Executable file)	PIC:	False	Endianness:	little
Language	:	Static: Relro:	False	Compiled:	on Mar 28 14:35:35 2011
Hash	es		Libraries		
MD5: SHA1:	1e5ec6c06e4f6bb958dcbb9fc63 ed47563dd5cc300716a9ba794	wininet.dll 6 mfc42.dll			

Fig. 6. Overview body malware WEBC2-DIV

It's time to do dissamble on the malware body so it is found that the behavior of the parent malware on the guest and do the planting value in regedit windows address

HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Run. Can be seen in color blocks like fig 7

	Culter	
File Edit View Windows Help	ag name or address here	
Address: 0x00401267 Sections: .data	Hendump Hendump Image: Constraint of the second se	
Quick Filter Dashboard Disassembly Graph	resource entry roints strings imports exports symbols Notepad	•
> Populating UI > Populating UI > Finished, happy reversing :) Project saved: WEBC2-DIV Project saved: WEBC2-DIV	Name Size Address End Address Sections ddata 5632 0x00401000 0x00403000 .	
Type "?" for help 🗧	Sections Comments	

Fig. 7. Address malware on registry Windows

State disassemble not stop at this, body malware got encryption that held important message can infected security computer then could be harmful. Like the encryption block color as fig 8 and fig 9

×0-		C
File Edit View Windows Help		
🗲 🔶 Type fla	ag name or address here	
Functions @18	Hexdump Ø 🕅	Sidebar 🛛 🗷
Name ▼ Size Imp. Offset 0 ffset 0 ffset 0 ffset 0 ffset 0x00401160 50 00 00 00 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 00 01 0 00 01 0 00 00 01 0 00 01 0 00 00 00 0 00 00 00 <td>4 5 6 7 8 9 A 8 C D E F 01245578948C0EF Parsing Information 5 00 00 00 17 00 00 00 00 00 00 00 Disasembly Endian Littl* 6 00 00 06 17 00 00 00 00 00 00 00 Disasembly Endian Littl* 7 7 5 5 7 7 3 7 4 32 7 8 2 7 5 8 7 1 1001/22785284000 Disasembly Endian Littl* 7 4 5 5 7 7 3 7 4 32 7 8 2 7 5 8 7 1 1001/22785284000 Disasembly Endian Littl* 7 4 5 5 7 7 3 7 3 6 6 6 7 2 0 4 6 5 7 7 3 6 6 7 7 2 0 4 6 6 Disasembly Robit 2 2 3 3 4 3 3 3 4 3 7 3 7 3 6 6 7 7 2 0 4 6 6 7 4 5 5 6 7 7 8 5 7 8 6 0 7 6 0 9 6 0 0 0 0 0 0 Disasembly Endian Littl* 7 4 5 5 7 7 7 3 6 6 6 7 4 2 0 6 0 0 0 0 0 Disasembly Robit 2 2 3 3 4 3 3 3 4 3 7 3 7 3 6 6 7 7 2 0 5 7 2 2 0 1 crmet Explorer 6 9 15 6 0 7 6 0 9 0 0 0 0 0 0 0 0 0 0 Disasembly Disasembly Robit 2 3 3 3 4 3 7 3 7 3 6 6 7 7 2 0 5 7 2 0 1 2 0 crmet Explorer 17 17 2 5 5 1 1 0 4 0 0 0 9 0 9 0 9 0 9 0 1 1.8. N.4. N.4. N.4. N.4. 17 17 2 5 5 1 1 0 4 0 0 0 9 0 9 0 9 0 9 0 9 1 0 1.8. N.4. N.4. N.4. N.4. 17 17 2 5 5 1 1 0 4 0 0 0 9 0 9 0 9 0 9 0 9 1 0 1.8. N.4. N.4. N.4. N.4. 18 0 2 0 0 0 0 3 5 0 8 7 C 4 3 3 3 0 5 0 7 7 1 0 5 1 1 0 0 0 0 9 0 9 0 9 0 0 0 0 0 0 0 0 0</td> <td>STACK null ESIL 71.esi.[4].*,esi,= TYPE mul SIZE 4 REFPTR 0 BVTES 3663647 PREFIX 0 MMEMONIC Imul OPCODE imul esi, dword ss AnDREFS 0x40116 ▼ Opcode description: ▼ X-Refs to current address: Address Instruction</td>	4 5 6 7 8 9 A 8 C D E F 01245578948C0EF Parsing Information 5 00 00 00 17 00 00 00 00 00 00 00 Disasembly Endian Littl* 6 00 00 06 17 00 00 00 00 00 00 00 Disasembly Endian Littl* 7 7 5 5 7 7 3 7 4 32 7 8 2 7 5 8 7 1 1001/22785284000 Disasembly Endian Littl* 7 4 5 5 7 7 3 7 4 32 7 8 2 7 5 8 7 1 1001/22785284000 Disasembly Endian Littl* 7 4 5 5 7 7 3 7 3 6 6 6 7 2 0 4 6 5 7 7 3 6 6 7 7 2 0 4 6 6 Disasembly Robit 2 2 3 3 4 3 3 3 4 3 7 3 7 3 6 6 7 7 2 0 4 6 6 7 4 5 5 6 7 7 8 5 7 8 6 0 7 6 0 9 6 0 0 0 0 0 0 Disasembly Endian Littl* 7 4 5 5 7 7 7 3 6 6 6 7 4 2 0 6 0 0 0 0 0 Disasembly Robit 2 2 3 3 4 3 3 3 4 3 7 3 7 3 6 6 7 7 2 0 5 7 2 2 0 1 crmet Explorer 6 9 15 6 0 7 6 0 9 0 0 0 0 0 0 0 0 0 0 Disasembly Disasembly Robit 2 3 3 3 4 3 7 3 7 3 6 6 7 7 2 0 5 7 2 0 1 2 0 crmet Explorer 17 17 2 5 5 1 1 0 4 0 0 0 9 0 9 0 9 0 9 0 1 1.8. N.4. N.4. N.4. N.4. 17 17 2 5 5 1 1 0 4 0 0 0 9 0 9 0 9 0 9 0 9 1 0 1.8. N.4. N.4. N.4. N.4. 17 17 2 5 5 1 1 0 4 0 0 0 9 0 9 0 9 0 9 0 9 1 0 1.8. N.4. N.4. N.4. N.4. 18 0 2 0 0 0 0 3 5 0 8 7 C 4 3 3 3 0 5 0 7 7 1 0 5 1 1 0 0 0 0 9 0 9 0 9 0 0 0 0 0 0 0 0 0	STACK null ESIL 71.esi.[4].*,esi,= TYPE mul SIZE 4 REFPTR 0 BVTES 3663647 PREFIX 0 MMEMONIC Imul OPCODE imul esi, dword ss AnDREFS 0x40116 ▼ Opcode description: ▼ X-Refs to current address: Address Instruction
Quick Filter X Dashboard Disassembly Graph	Hexdump Pseudocode Entry Points Strings Imports Exports Symbols Notepad Classes	▼ X-Refs from current addres
Entry Points 🛛 🕅	Sections	0 8
> Populating UI > Finished, happy reversing :)	Name - Size Address End Address data 552 000401000 0000000 .ssrc 512 0x00403000 0x00404000	
Type "?" for help	Sections Comments	

Fig. 8. HexDump value encrypt malware WEBC2-DIV

× = _			Ø	
File Edit View Windows	Help			
+ 		Type flag name or address here		
Functions	Ø 🗙	Disassembly	Sidebar 🖉 🗷	
Name* Size Imp. Offset	0x004401167 0x00440116c 0x00440116c 0x00440116c 0x00440116c 0x00440116c 0x00440116c 0x004401192 0x004401194 0x004401194 0x00440114c 0x00440114c 0x00440114c 0x00440114c 0x00440114c 0x00440114c 0x00440114c 0x00440114c 0x00440114c 0x00440115c 0x00440117c 0x00440117c 0x00440117c	<pre>add byte [ecs], edx + 0x40], ch * add byte [ess], dh str.6K66pesgiHreT08MPrO_30dg2Z7st2x_uXqkz: .string "SoftwareVNOv(\$30dg2Z7st2x*uXqkz"; len=38 add byte [eax], al str.Software_Nkicrosoft\Windows\UurrentVersion_Run: .string "Intersoft Internet Explorer Exelon: .string "Microsoft Internet Explorer Exelon "; len=36 js 0x401250 add byte [eax], al add byte [eax], al add byte [eax], al add byte [eax], al add byte [eax], al</pre>	STACK null ▲ ESL 71.esi.[4],*.esi.# SZE 4 REFPTR 0 BYEES 366b3647 ID 213 PREFNONC mull esi. dword ss OPNOES mull esi. dword ss Yopcode description: ▼	
■ Ouick Filter	0x004011fb 0x004011fd 0x004011ff 0x00401201 Dashboard Disasse	add byte [eax], al add byte [eax], al add bh, bh Invalld mubly Graph Hexdump Pseudocode Entry Points Strings Imports Exports Symbols Notepad Classes	- X Boto from current addres	
Fr	ntry Points	R Sections	* X-Reis from current addres *	
> Populating UI > Finished, happy revers	sing :)	Name Size Address End Address Size Size Size Size Address Size Size <td></td>		
Type "?" for help		Sections Comments		

Fig. 9. Value string WEBC2-DIV encrypt on disassembler

Decode encrypt to get information about malware's behave. One another hand, details where's malware sent data to his creator. Result decode we get as fig 10.



Fig. 10. Value string WEBC2-DIV encrypt on disassembler

V. Conclusion

WEBC2-DIV is malicious software best malware to espionage activities performed there are : (1) Phising email, (2) Phising login credential, (3) Backdoor, (4) Remote trojan. This malware well-known since 2010. That does not out posibility malware WEBC2-DIV do update itself by creator then encryption in body malware more difficult than before.

At future research faced automation scanned challenge. Recognize malware by hash then clustering by type malware. This will be innovation and largest contribution to malware research.

REFERENCES

- [1] P., dan Grance, T Mell, "The NIST definition of cloud," U.S, 2011.
- [2] D dan Nandi, S Devi, "Detection of Packed Malware," in Proceeding SecurIT '12 Proceedings of the First International Conference on Security of Internet of Things, NY, 2012, pp. 22 - 26.
- [3] Joshua.I.J., Alan.H., Chen-Ching. L dan Pavel. G Ahmed.F.S., "Towards Automated Malware Behavioral Analysis and Profiling for Digital Forensic Investigation Purposes," in 4th International Conference on Digital Forensics and Cyber Crime ICDF2C 2012, Lafayette, Indiana, USA, 2012.
- [4] H and Lee Jeong K, "Code graph for malware detection, in:Information Networking," ICOIN (International Conference), pp. 1-5, 2008
- [5] E. Al., Jebril, I. H., dan Zaqaibeh, B Daoud, "Vol 1. No.2 Computer Virus Stategies and Detection Methods," in Int. J. Open Probles Compt. Math., 2 September 2008.
- [6] M., Yegneswaran, V., Saidi, H., Porras, P dan Lee, W Sharif, Eureka: A Framework for Enabling Static Malware Analysis. Berlin, Heidelberg: Springer, 2008, pp. 481-500.
- [7] C., Merwe, A.V.D dan Paula, k Mariana, "Secure Computing Benefits, Risk and Controls," IEEE-Information Security, p. Soutch Africa, 2011.