

# Gh0stRAT

## 악성코드 분석 보고서



이름	전동혁
----	-----

# 목차

1. 개요 .....	3
1.1 관련 기술 .....	3
2. Gh0stRAT 분석 .....	4
2.1 특징 .....	4
2.2 기능 .....	5
2.3 지속 메커니즘 등록 .....	5
3. 테스트 환경 .....	6
3.1 테스트 절차 .....	7
4. 기능 .....	8
4.1 프로그램 기능 소개 .....	8
4.2 작동되지 않는 기능 .....	13
5. 트래픽 분석 .....	15
5.1 초기 연결 .....	15
5.2 HeartBeat .....	15
5.3 기능별 명령 트래픽 .....	16
6. Virustotal 확인 결과 .....	19
6.1 Gh0st RAT v1.0.exe (Master) .....	19
6.2 Gh0st.exe (Agent) .....	20
7. Gh0st.exe 동적 분석 (xdbg, cuckoo) .....	22
7.1 xdbg 분석 .....	22
7.2 cuckoo 샌드박스 결과 .....	32
8. 탐지 Signature .....	33

## 1. 개요

gh0stRAT 프로그램은 중국 해커 그룹이 개발한 유명한 원격 관리 도구 (Remote Administration Tool) 의 일종으로 감염된 PC를 원격으로 제어할 수 있도록 설계되었으며 2009년 “GhostNet” 해킹 캠페인에서 사용되면서 널리 알려졌음.

Gh0stRAT는 백도어 악성코드로서 C&C 서버로부터 공격자의 명령을 받아 다양한 악성 행위를 수행할 수 있음. 기본적으로 파일 탈취, 업로드, 실행, 삭제, 기능을 포함하며 프로세스 관리, 키로깅 및 스크린 로깅 등 일반적인 백도어 악성코드에서 제공하는 기능들이 지원됨. 기능적인 내용 외에도 진단 시스템(안티 바이러스)를 회피하기 위한 목적으로 다양한 Stealth 기법도 제공함. 정상적인 프로세스로 위장하는 기능, 프로세스를 Hide 하는 Rootkit 기능, 언제 악성코드가 실행 될지 Delay를 설정하는 기능 등 다양한 스텔스 기법을 지원함.

본 문서는 gh0stRAT를 이용한 공격들에 대한 전체적인 흐름을 분석하고 확인된 기능들을 분석 후 C&C 서버와 클라이언트 간 통신 데이터의 특정 시그니처를 기반으로 침입 탐지 시스템(IDS)에 룰을 적용하여 차단하는 과정에 대해 각 단계별로 상세하게 정리함.

Gh0stRAT는 주로 Windows 시스템을 대상으로 하며, 오픈소스 버전이 존재하고 다양한 변형이 등장하여 배포되고있음.

### 1.1 관련 기술

#### RAT

RAT (Remote Administration Tool)는 원격에서 감염된 시스템을 완전히 제어할 수 있도록 설계된 도구임. 공격자가 RAT를 이용하면 피해자의 컴퓨터를 마치 자신이 직접 조작하는 것처럼 사용 할수 있음.

#### Backdoor

컴퓨터 시스템의 백도어(Backdoor)는 일반적인 인증과 암호화를 우회해 원격 접속 및 암호화된 텍스트에 대한 권한을 취득하는 등 은밀히 악성코드를 실행하는 전형적인 방법. 백도어는 설치된 프로그램의 형태를 취하기도 하고, 기존 프로그램 또는 하드웨어의 변형일 수도있음

### 좀비 컴퓨터

좀비 컴퓨터(zombie computer)는 악성코드에 감염된 컴퓨터를 뜻함. C&C 서버의 제어를 받아 주로 DDOS 공격 등에 이용됨

### 트로이 목마

트로이 목마(Trojan horse)는 악성 루틴이 숨어 있는 프로그램으로 겉보기에는 정상적인 프로그램으로 보이지만 실행하면 악성 코드를 실행함. 이 이름은 트로이 목마 이야기에서 따온 것으로 겉보기에는 평범한 목마 안에 적군의 병사가 숨어 있었다는 것에 비유한 것임. 주로 사회공학 기법의 형태로 퍼지고 많은 트로이 목마들은 백도어로서 사용됨.

## 2. Gh0stRAT 분석

### 2.1 특징

Gh0stRAT는 Visual Basic 6.0으로 개발되었으며 C&C 서버에서 공격자의 명령을 전달받아 수행하는 백도어 악성코드임. 현재 배포된 1.0버전 같은 경우는 빌드된지 기간이 오래 지난 관계로 최신 윈도우 (Windows 10,11) 버전에서는 정상적으로 동작이 되지 않았으며 정상적으로 동작하는 환경 WinXP x86에서 분석을 진행하였음.

Master (C&C)	
파일 설명	Gh0stRAT v1.0
유형	응용 프로그램
파일 버전	1.0.0.0
크기	144KB
동작 OS	WinXP, Win10
원본 파일 이름	Project1.exe
패킹 여부	X
개발 언어	Visual Basic 6.0

Agent (Client)	
파일 설명	
유형	응용 프로그램
파일 버전	
크기	28.4KB
동작 OS	WinXP x86
원본 파일 이름	
패킹 여부	O
개발 언어	Visual C/C++

## 2.2 기능

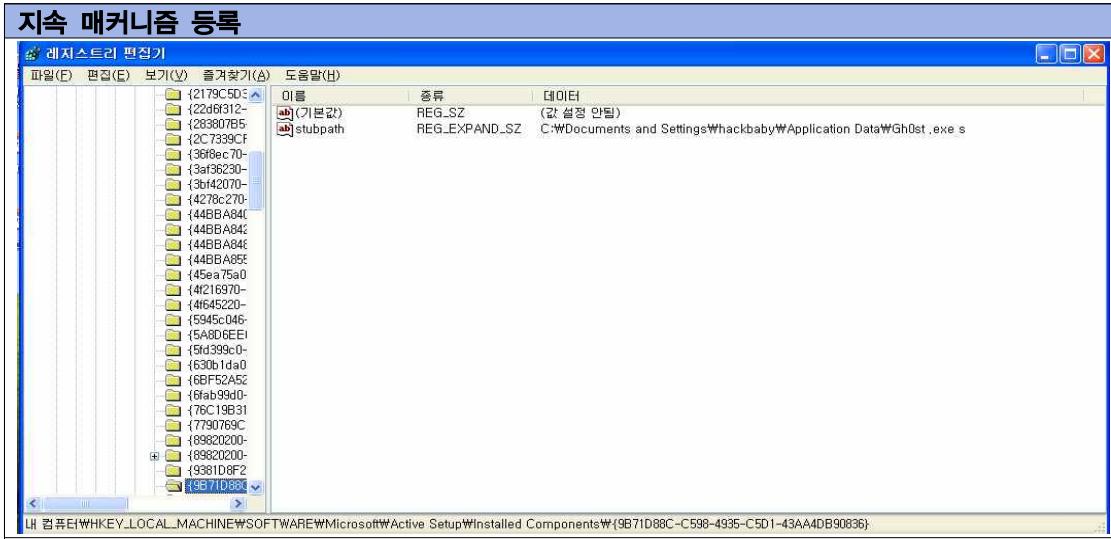
Gh0stRAT 배포된 1.0 버전의 주요 기능은 다음과 같음.

- 키로깅
- 파일 관리
- 실시간 화면 캡처
- 웹캠 제어
- 원격 쉘, 네트워크

## 2.3 지속 메커니즘 등록

Gh0st RAT는 C2 서버와 지속적인 세션을 유지하기 위해서 레지스트리 키

{9B71D88C-C598-4935-C5D1-43AA4DB90836}를 등록해

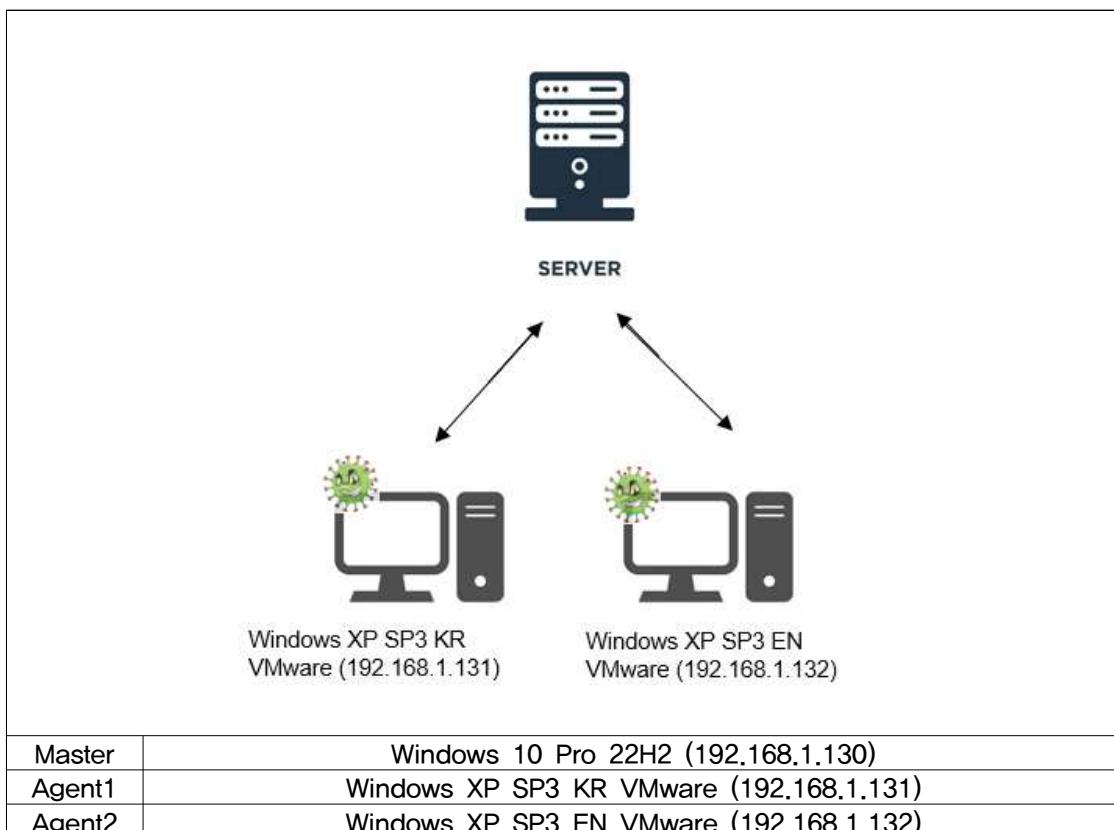
C:\Documents and Settings\hackbaby\Application Data\Gh0st.exe
경로에 Agent 프로그램이 재부팅 시 실행되도록 지속 메커니즘을 등록한다.
※ 해당 레지스트리 키는 기본 값이므로 변경될 수 있음.
 <b>지속 매커니즘 등록</b> 파일(E) 폴더(E) 보기(V) 즐겨 찾기(A) 도움말(H) (2179C5D3) 이름 종류 데이터 (22d9312-... (기본값) REG_SZ (값 설정 안됨) (239307B5-... stubpath REG_EXPAND_SZ C:\Documents and Settings\hackbaby\Application Data\Gh0st.exe (2C7339CF-... (36f8ec70-... (3a156230-... (3bf42070-... (4278c270-... (44BA84C-... (44BA84E-... (44BA85E-... (45e5a75a0-... (47216970-... (47645220-... (5945c046-... (5A8DDEE)-... (5d399c0-... (630b1da0-... (68F52A52-... (6ab99d0-... (78C19831-... (7790769C-... (89820200-... (89820200-... (938108F2-... (9B71D88C-...  내 컴퓨터\HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components\{9B71D88C-C598-4935-C5D1-43AA4DB90836}
HKLM\Software\Microsoft\Active Setup\Installed Components Microsoft Outlook Express Setup... Microsoft Corporation c:\program files\outl... 2008-04-14 오전 3:30 n/a Microsoft Corporation c:\documents and set... 2007-01-18 오전 7:19 주소록 6 Outlook Express Setup... Microsoft Corporation c:\program files\outl... 2008-04-14 오전 3:30
<b>HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components\{9B71D88C-C598-4935-C5D1-43AA4DB90836}</b>

### 3. 테스트 환경

테스트 환경은 다음과 같다.

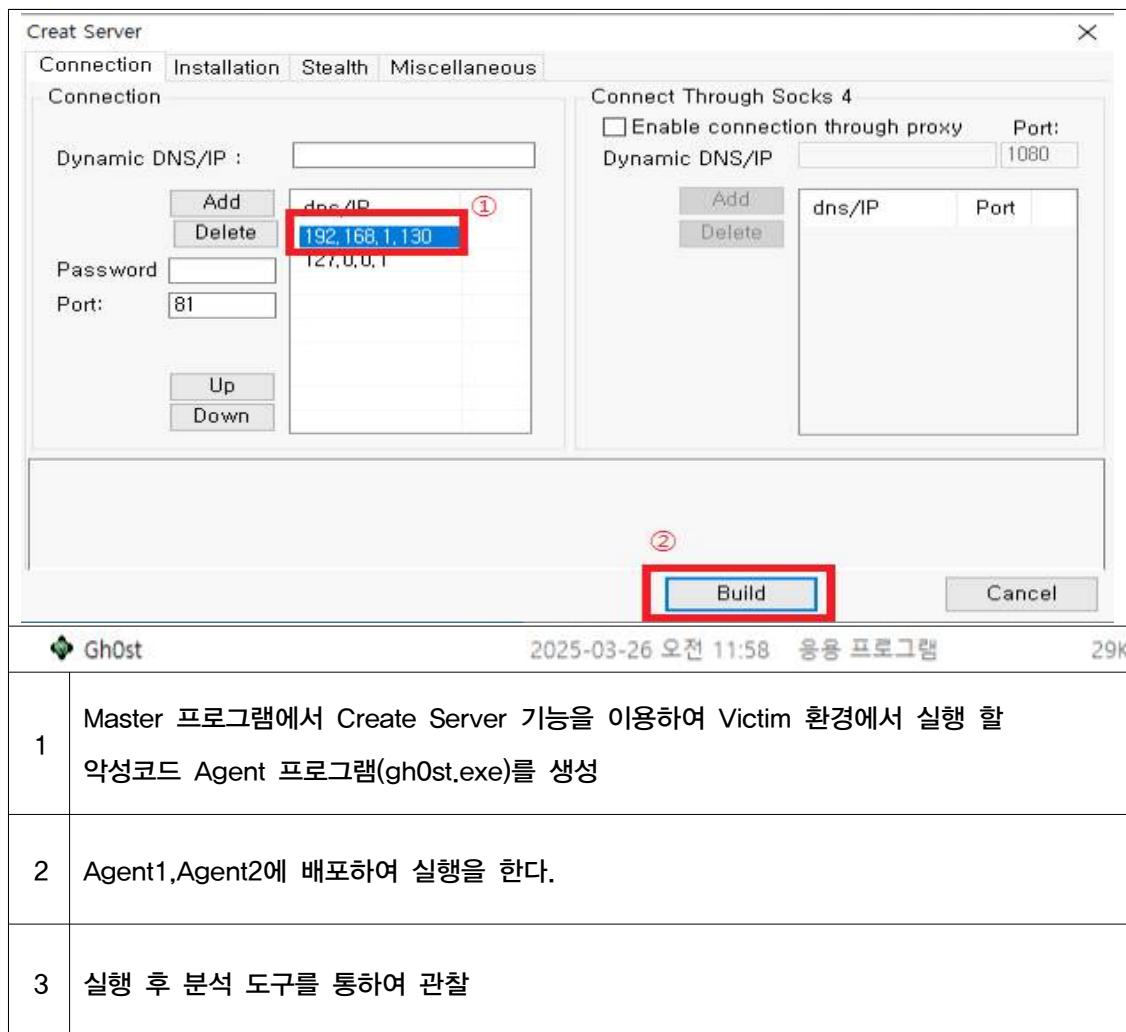
C&C (Windows 10 Pro 22H2 192.168.1.130)에 Master 프로그램을 배치해두고

Client (Windows XP SP3 KR VMware 192.168.1.131), (Windows XP SP3 KR VMware 192.168.1.132)를 배치해둔 뒤 C&C와 Client가 서로 통신이 되도록 구성한다.



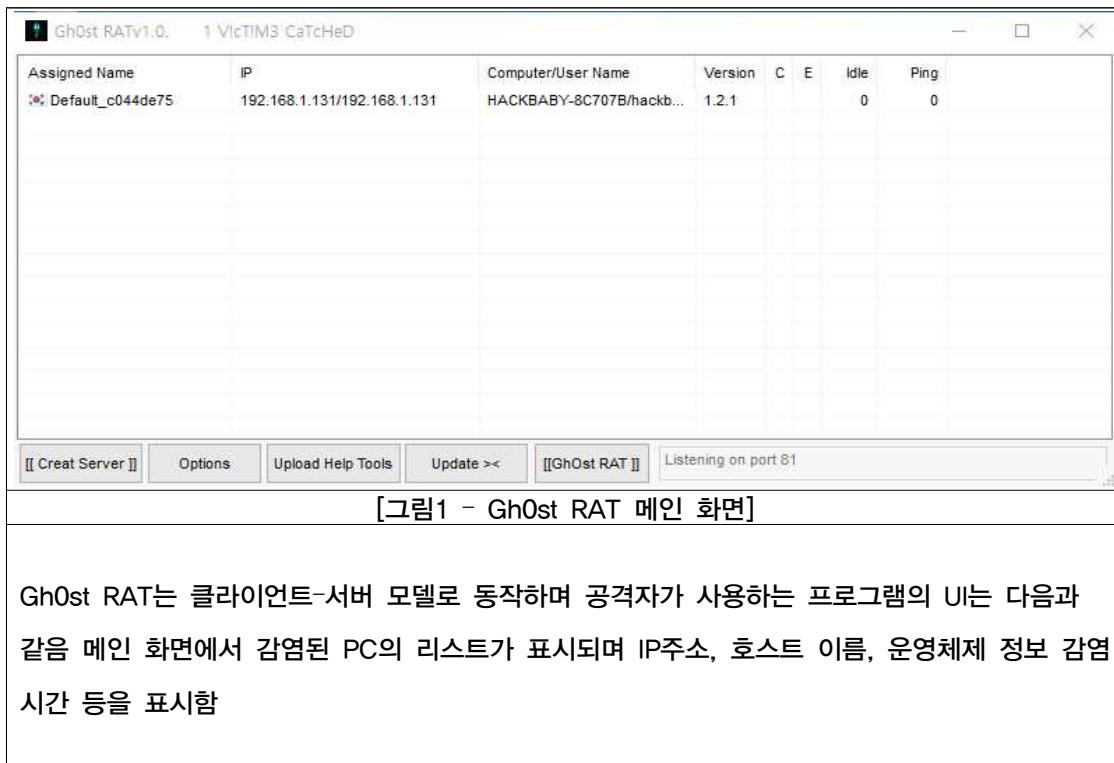
## 3.1 테스트 절차

테스트는 다음과 같은 절차로 진행한다.

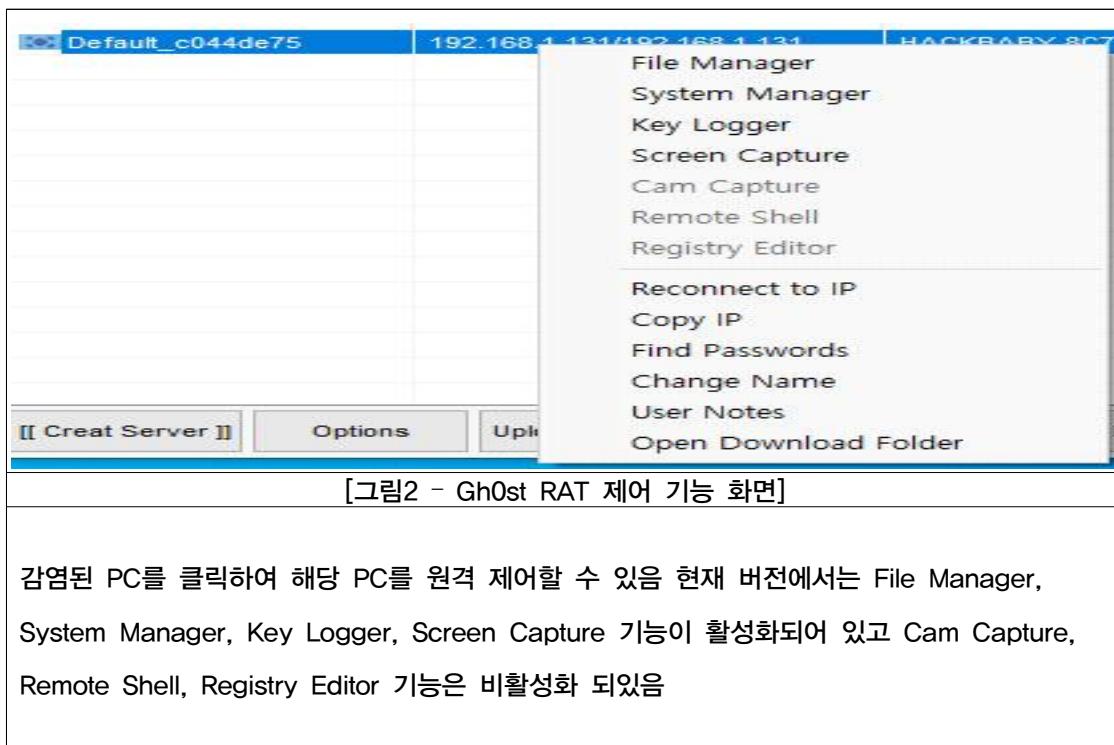


## 4. 기능

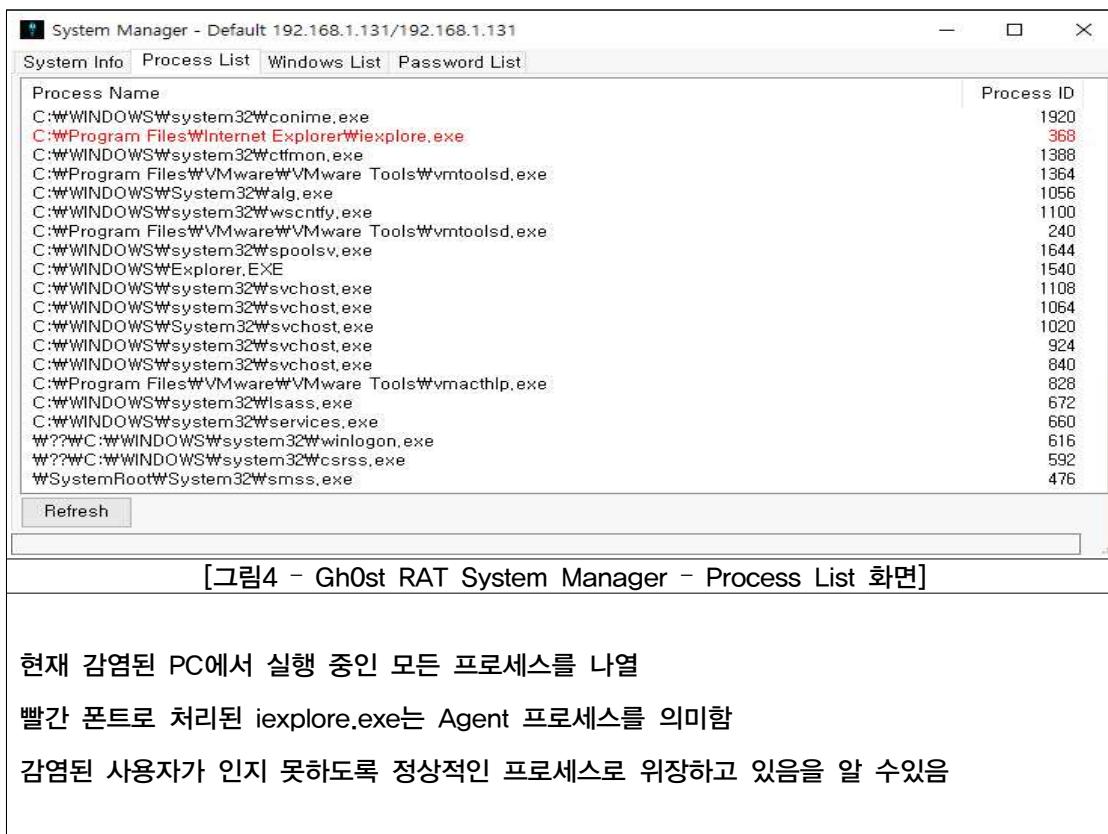
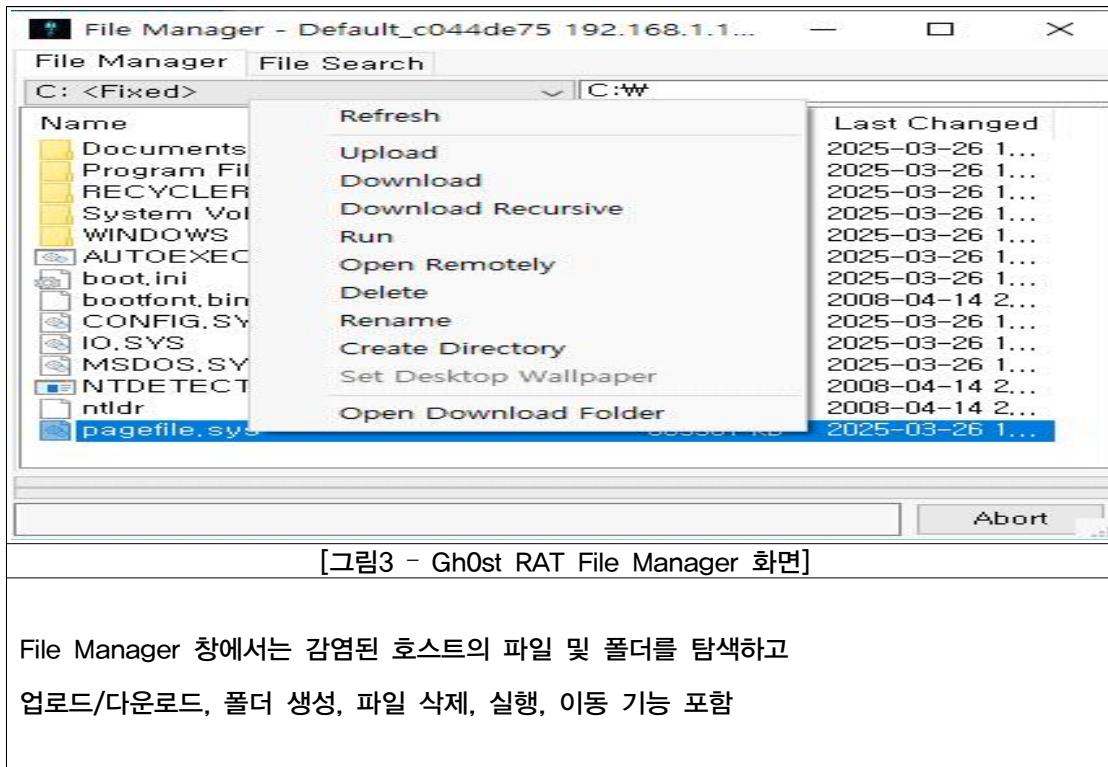
### 4.1 프로그램 기능 소개

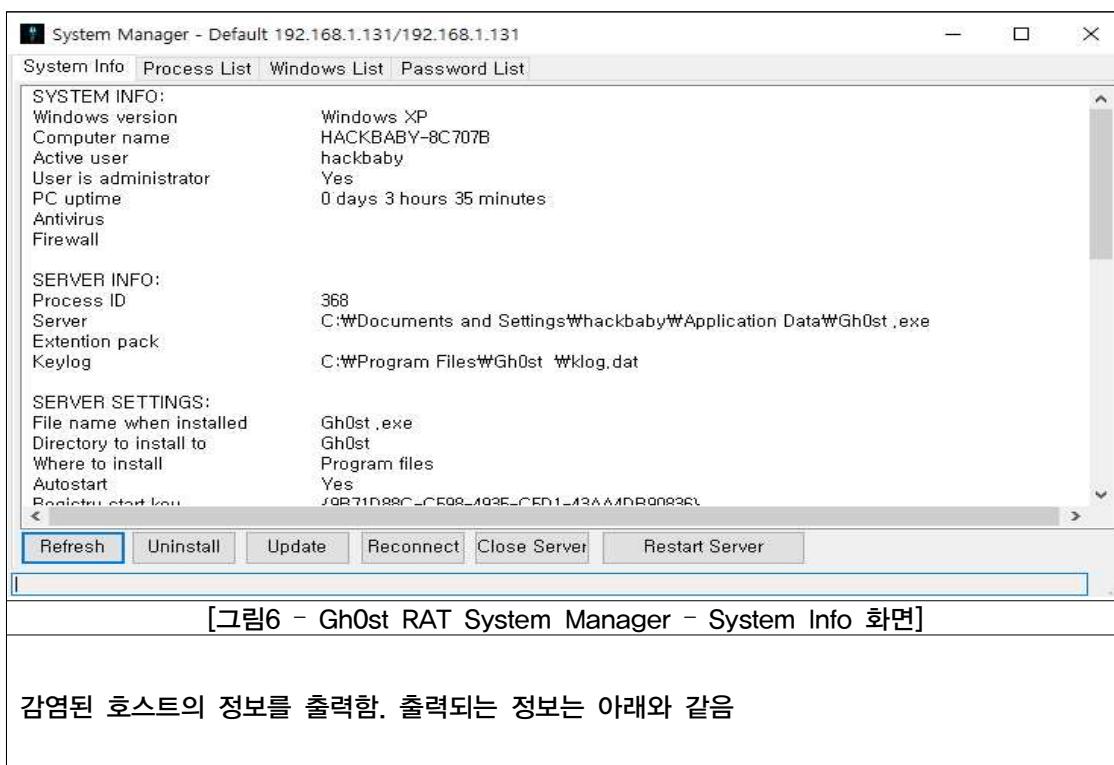
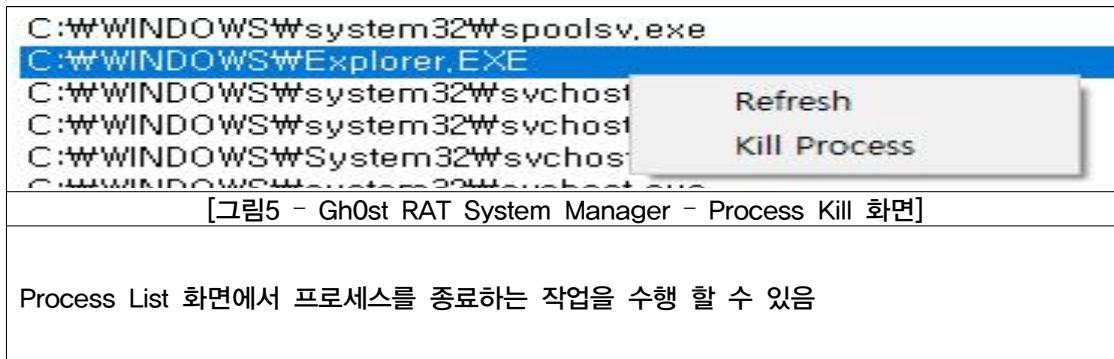


Gh0st RAT는 클라이언트-서버 모델로 동작하며 공격자가 사용하는 프로그램의 UI는 다음과 같은 메인 화면에서 감염된 PC의 리스트가 표시되며 IP주소, 호스트 이름, 운영체제 정보 감염 시간 등을 표시함



감염된 PC를 클릭하여 해당 PC를 원격 제어할 수 있음 현재 버전에서는 File Manager, System Manager, Key Logger, Screen Capture 기능이 활성화되어 있고 Cam Capture, Remote Shell, Registry Editor 기능은 비활성화 되있음





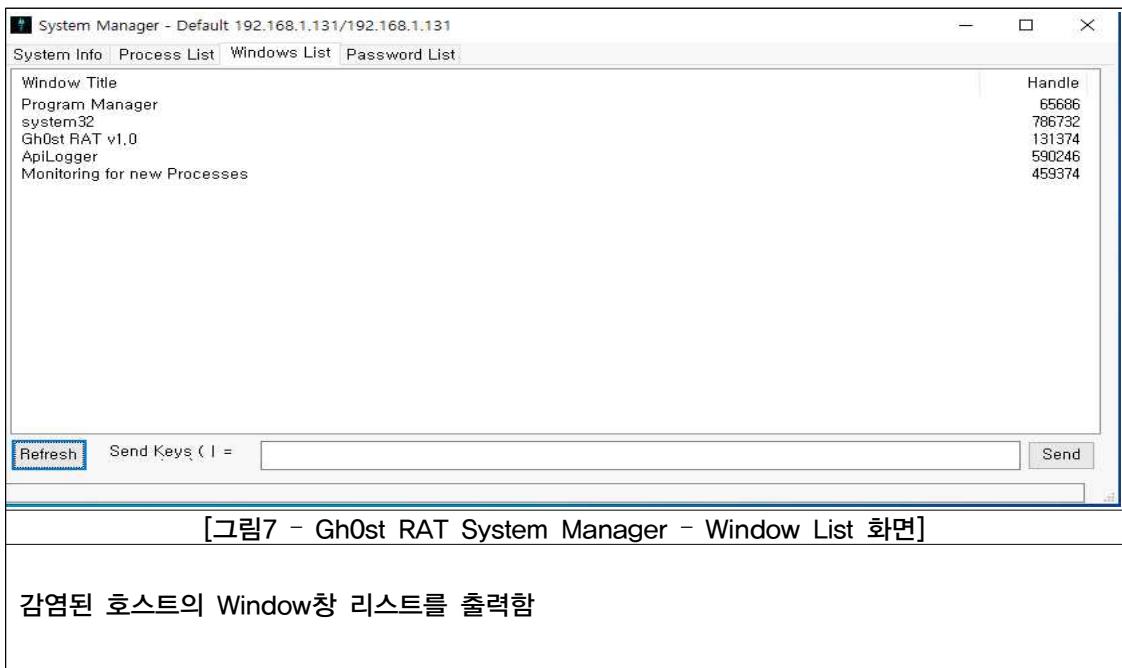
SYSTEM INFO	
Windows version	
Computer name	
Active user	
User is administrator	
PC uptime	
Antivirus	
Firewall	



ServerInfo
ProcessID
Server
Extention pack
Keylog

Server SETTINGS
File name when installed
Directory to install to
Where to install
Autostart
Registry start key
Mutex name
Registry setting key
Include extension pack when
Offline Keylogger
Keylogger logfile
Exclude Shift and Ctrl from key
Exclude backspace fromkeylog
Inject to a specified process
Persistant server
Assigned name
Unique identifier
Stealth mode
Server file attribut hidden
Server file set to old date
Melte server
Delayed server start when first
Rootkit hide process
Kernel level unhooking
Use TOR plugin
Password
Port
Connect through proxy
Dynamic DNS/IP 1
Dynamic DNS/IP 2





## 4.2 작동되지 않는 기능





기능	
File Manager	파일 및 폴더를 탐색하고 업로드 / 다운로드 가능 파일 삭제, 실행, 이동 기능 포함
System Manager	프로세스 관리, 실행 중인 모든 프로세스를 나열 특정 프로세스를 종료하거나 새로운 프로세스를 실행 가능 Windows 서비스 목록을 표시, 서비스 시작, 중지, 삭제 가능
Key Logger	키보드 입력을 실시간으로 캡처
Screen Capture	PC 화면을 정기적으로 캡처하여 저장
Cam Capture	X
Remote Shell	X
Registry Editor	X

## 5. 트래픽 분석

### 5.1 초기 연결

3way Handshake 이후 Master에서 Agent로 초기 연결을 위한 패킷을 요청한다. 요청 받은 클라이언트는 응답하고 Master와 Agent 연결 세션이 생성된다.

Agent	방향	트래픽 페이로드
Windows XP KR Agent1	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 30 97 60 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 05 05 28 14 85 81 97 18 8e f8 70 12 fa f0 84 78 00 00 02 04 05 b4 01 01 04 02
	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 30 0c 54 40 00 80 06 6a 1e c0 a8 01 83 c0 a8 01 82 05 05 00 51 97 18 8e f7 00 00 00 00 70 02 ff ff d3 63 00 00 02 04 05 b4 01 01 04 02
Windows XP EN Agent2	M→A	00 0c 29 e4 93 bc 00 0c 29 09 25 6f 08 00 45 00 00 30 5e 47 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 85 00 51 04 06 d6 3d 8b 7e 08 1d 2c f8 70 12 fa f0 84 7a 00 00 02 04 05 b4 01 01 04 02
	A→M	00 0c 29 09 25 6f 00 0c 29 e4 93 bc 08 00 45 00 00 28 00 31 40 00 80 06 76 47 c0 a8 01 85 c0 a8 01 82 04 06 00 51 08 1d 2c f8 d6 3d 8b 7f 50 10 ff ff 90 53 00 00 00 00 00 00 00 00 00 00 00 00

### 5.2 HeartBeat

15초 간격으로 Master에서 Agent로 하트비트 패킷을 요청한다. 요청 받은 클라이언트는 응답하고 응답을 받은 Master에서 확인 패킷을 다시 클라이언트에게 전송한다.

Agent	방향	트래픽 페이로드
Windows XP KR Agent1	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 31 97 6f 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 05 05 28 14 85 c1 97 18 90 3f 50 18 f9 a9 84 79 00 00 05 00 00 00 bc 60 74 1c cc
	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 35 0c 62 40 00 80 06 6a 0b c0 a8 01 83 c0 a8 01 82 05 05 00 51 97 18 90 3f 28 14 85 ca 50 18 ff b7 21 f0 00 00 09 00 00 00 9a 60 74 1c cc 83 d7 33 74
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00

		00 28 97 70 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 05 05 28 14 85 ca 97 18 90 4c 50 10 f9 9c 84 70 00 00
Windows XP EN Agent2	M→A	00 0c 29 e4 93 bc 00 0c 29 09 25 6f 08 00 45 00 00 31 5e 4c 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 85 00 51 04 06 d6 3d 8b 91 08 1d 2d eb 50 18 f9 fd 84 7b 00 00 05 00 00 00 bc af e9 48 cc
	A→M	00 0c 29 09 25 6f 00 0c 29 e4 93 bc 08 00 45 00 00 35 00 49 40 00 80 06 76 22 c0 a8 01 85 c0 a8 01 82 04 06 00 51 08 1d 2d eb d6 3d 8b 9a 50 18 ff e4 c6 15 00 00 09 00 00 00 9a af e9 48 cc 09 fc 32 74
	M→A	00 0c 29 e4 93 bc 00 0c 29 09 25 6f 08 00 45 00 00 28 5e 4d 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 85 00 51 04 06 d6 3d 8b 9a 08 1d 2d f8 50 10 f9 f0 84 72 00 00

### 5.3 기능별 명령 트래픽

기능	방향	트래픽 페이로드
파일 삭제	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 62 f7 7e 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 10 26 65 d6 26 0a 95 63 55 50 18 f6 be 84 aa 00 00 36 00 00 00 23 3d ba 1a 88 2b c1 47 19 a6 a3 00 89 b3 3a 09 6e 46 b8 49 1b 13 16 9d c0 d6 21 04 15 5c bd bf de 63 61 93 04 50 10 12 60 14 9c 3c 93 94 36 98 bd 26 01 7e 92 f5
	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 62 01 72 40 00 80 06 74 ce c0 a8 01 83 c0 a8 01 82 04 10 00 51 0a 95 63 55 26 65 d6 60 50 18 ff 20 fa 32 00 00 36 00 00 00 8f 3d ba 1a 88 2b c1 47 19 a6 a3 00 89 b3 3a 09 6e 46 b8 49 1b 13 16 9d c0 d6 21 04 15 5c bd bf de 63 61 93 04 50 10 12 60 14 9c 3c 93 94 36 98 bd 26 01 7e 92 f5
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 5b f7 7f 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 10 26 65 d6 60 0a 95 63 8f 50 18 f6 84 84 a3 00 00 2f 00 00 00 2a 3d ba 1a 88 2b c1 47 19 a6 a3 00 89 b3 3a 09 6e 46 b8 49 1b 13 16 9d c0 d6 21 04 15 5c bd bf de 63 61 93 04 50 10 12 60 14 9c 3c 93 94 57

	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 a2 01 73 40 00 80 06 74 8d c0 a8 01 83 c0 a8 01 82 04 10 00 51 0a 95 63 8f 26 65 d6 93 50 18 fe ed 8e a4 00 00 76 00 00 00 b6 79 b1 68 cb 77 e6 35 40 f1 fd 46 cf be 6b 54 27 54 dc 0c 5f 51 45 c3 93 a2 4c 42 5a 38 e5 99 b8 35 2a ff 8f bc f8 e3 73 f1 03 b3 5a f8 61 d1 e9 3c 62 23 b6 9d 0c 43 9b 32 19 73 19 5f 54 49 d2 92 d6 b3 5d 3f 12 64 8c 27 18 a9 99 80 30 c4 5a 29 10 53 f9 b9 48 94 1d 4d dd e5 31 25 68 03 4a 77 fb 05 26 b2 77 e7 c8 ce 21 fc 2f 18 08 cb ec ee c4 6e 0a c8
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 28 f7 80 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 10 26 65 d6 93 0a 95 64 09 50 10 f6 0a 84 70 00 00
	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 2e 01 74 40 00 80 06 75 00 c0 a8 01 83 c0 a8 01 82 04 10 00 51 0a 95 64 09 26 65 d6 93 50 18 fe ed 31 0c 00 00 02 00 00 00 89 7e
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 28 f7 81 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 10 26 65 d6 93 0a 95 64 0f 50 10 f6 04 84 70 00 00
파일 실행	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 6d f7 ce 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 7e b1 1b 55 78 b8 50 18 fa f0 84 b5 00 00 41 00 00 00 27 3d ba 1a 88 2b c1 47 19 a6 a3 00 89 b3 3a 09 6e 46 b8 49 1b 13 16 9d c0 d6 21 04 15 5c bd bf de 63 61 93 04 50 10 12 60 14 9c 3c 93 94 ec 1e f9 cd bd ac 4b 33 92 10 56 bc 8b bb 4f 4e 2b 0e e0
	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 28 01 fd 40 00 80 06 74 7d c0 a8 01 83 c0 a8 01 82 04 1f 00 51 1b 55 78 b8 b4 b5 7e f6 50 10 fd 56 61 fe 00 00 00 00 00 00 00 00 00
프로세스 종료	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 32 f7 e3 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 7f 4a 1b 55 8b d2 50 18 f6 d1 84 7a 00 00 06 00 00 00 16 c6 85 46 cc 44

	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 32 02 0e 40 00 80 06 74 62 c0 a8 01 83 c0 a8 01 82 04 1f 00 51 1b 55 8b d2 b4 b5 7f 54 50 18 fc f8 a9 cc 00 00 06 00 00 00 f2 4f b4 70 f8 44
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 28 f7 e4 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 7f 54 1b 55 8b dc 50 10 f6 c7 84 70 00 00
원도우(창) 종료	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 32 f7 fd 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 7f ba 1b 55 8c bb 50 18 f5 e8 84 7a 00 00 06 00 00 00 30 46 81 4a cc 44
	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 45 02 1f 40 00 80 06 74 3e c0 a8 01 83 c0 a8 01 82 04 1f 00 51 1b 55 8c bb b4 b5 7f c4 50 18 fc 88 55 eb 00 00 19 00 00 00 9c c5 75 66 09 9c 1f 88 b2 6d ed cd 54 2f f6 47 27 46 53 f2 d7 97 bf 16 a7
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 28 f7 fe 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 7f c4 1b 55 8c d8 50 10 f5 cb 84 70 00 00
	M→A	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 2f 02 31 40 00 80 06 74 42 c0 a8 01 83 c0 a8 01 82 04 1f 00 51 1b 55 8d 94 b4 b5 80 15 50 18 fc 37 0d c6 00 00 03 00 00 00 bc 4c 80
	A→M	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 28 f8 15 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 80 15 1b 55 8d 9b 50 10 fa c7 84 70 00 00
	M→A	00 0c 29 e2 ea 69 00 0c 29 09 25 6f 08 00 45 00 00 3e f8 34 40 00 80 06 00 00 c0 a8 01 82 c0 a8 01 83 00 51 04 1f b4 b5 80 87 1b 55 8e 5c 50 18 fa 06 84 86 00 00 12 00 00 00 1f 38 82 46 cc d6 a3 32 74 c2 cd 74 fa 83 5b 67 0a 66
스크린캡처	A→M	00 0c 29 09 25 6f 00 0c 29 e2 ea 69 08 00 45 00 00 46 02 4a 40 00 80 06 74 12 c0 a8 01 83 c0 a8 01 82 04 1f 00 51 1b 55 8e 5c b4 b5 80 9d 50 18 fb af c2 51 00 00 1a 00 00 00 fb 4a b6 71 f5 7c 90 4e 45 f5 ff 43 c3 a3 27 52 32 54 97 18 5f 55 03 c3 db a5

## 6. VirusTotal 확인 결과

### 6.1 GhOst RAT v1.0.exe (Master)

Community Score		① 13/74 security vendors flagged this file as malicious		Reanalyze	Similar	More
cce43f10fbef0d1cfac4da491697a368ba594a5403e9ecd5b3e28d1476c38b89	Project1.exe	Size	144.00 KB	Last Analysis Date	7 months ago	 EXE
MD5	c11cb98ab1dc8671b964dbe1ee42422b					
SHA-256	cce43f10fbef0d1cfac4da491697a368ba594a5403e9ecd5b3e28d1476c38b89					
DetectItEasy	PE32 Compiler: Visual Basic (6.00.8041) [Native] Linker: Microsoft Linker (6.0)					
File size	144.00 KB (147456 bytes)					
Creation Time	2012-09-04 12:40:31 UTC					
Names	GhOst RAT v1.0.exe Project1 Project1.exe Project1_FC10EBA6.exe vti-rescan GhOst RAT v1.0.exe_ file-4936171_exe					

AhnLab-V3	Trojan/Win32.Birfost.C301735
Cylance	Unsafe
Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS
Kingsoft	Malware.kb.a.830
MaxSecure	Trojan.Malware.300983.susgen
Microsoft	PUA:Win32/Presenoker
Rising	PUA.Presenoker!8.F608 (CLOUD)
Sophos	Generic Reputation PUA (PUA)
Symantec	Hacktool
Trapmine	Malicious_high.ml.score
TrendMicro-HouseCall	TROJ_GEN.R002H05H724
Xcitium	TrojWare.Win32.VB.NMV@4yuc48

## 6.2 Gh0st.exe (Agent)

67 / 73 security vendors flagged this file as malicious

Community Score: 67 / 73

Gh0st.exe

peexe overlay

MD5: 36197700bab421d192795c15a45537bb

SHA-256: 1f18dc69772f12c3775bf4c485a5669cceaf9e2b6df32fb5ed80ea1561487cc

File size: 28.40 KB (29085 bytes)

Creation Time: 2007-01-17 22:19:02 UTC

Names: Gh0st .exe

Last Analysis Date: a moment ago

Size: 28.40 KB

Type: EXE

File Details	
MD5	36197700bab421d192795c15a45537bb
SHA-256	1f18dc69772f12c3775bf4c485a5669cceaf9e2b6df32fb5ed80ea1561487cc
DetectItEasy	PE32 Compiler: Microsoft Visual C/C++ (12.00.8966) [C++] Linker: Microsoft Linker (6.0) Tool: Visual Studio
File size	28.40 KB (29085 bytes)
Creation Time	2007-01-17 22:19:02 UTC
Names	Gh0st .exe

AhnLab-V3	Trojan/Win32.Bifrose.R1454
AliCloud	Trojan[dropper]:Win/Bifrose.8f26a0df
ALYac	Trojan.Crypt.BH
Antiy-AVL	Trojan[Backdoor]/Win32.Bifrose
Arcabit	Trojan.Crypt.BH
Avast	Win32:Agent-AAZQ [Trj]
AVG	Win32:Agent-AAZQ [Trj]
Avira (no cloud)	BDS/Bifrose.keiqw
Baidu	Win32.Trojan.Agent.dm
BitDefender	Trojan.Crypt.BH
Bkav Pro	W32.AIDetectMalware
ClamAV	Win.Trojan.Agent-36385
CrowdStrike Falcon	Win/malicious_confidence_100% (D)
CTX	Exe.trojan.crypt
Cylance	Unsafe
Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS
DrWeb	BackDoor.Bifrost.779
Elastic	Malicious (high Confidence)
Emsisoft	Trojan.Crypt.BH (B)
eScan	Trojan.Crypt.BH
ESET-NOD32	Win32/Bifrose.ADR
Fortinet	W32/Bifrose.BBT!tr
GData	Win32.Trojan.PSE.1HV1UKG
Google	Detected
Gridinsoft (no cloud)	Trojan.Win32.Gen.bot!i
Huorong	Backdoor/Bifrose.z
Ikarus	Backdoor.Win32.Bifrose
Jiangmin	Backdoor/Bifrose.ks
K7AntiVirus	Trojan ( 000158851 )

K7GW	Trojan ( 004bff5e1 )
Kaspersky	Backdoor.Win32.Bifrose.bgn
Kingsoft	Win32.Hack.Bifrose.73757
Malwarebytes	Generic.Malware.AI.DDS
MaxSecure	Poly.Trojan.Agent.BCN
McAfee Scanner	Real Protect-LS!36197700BAB4
Microsoft	Backdoor:Win32/Bifrose
NANO-Antivirus	Trojan.Win32.Agent.cojafi
Panda	Generic Malware
QuickHeal	Backdoor.Bifrose.7730
Rising	Backdoor.Bifrose!1.A05C (CLASSIC)
Sangfor Engine Zero	Suspicious.Win32.Save.a
SecureAge	Malicious
SentinelOne (Static ML)	Static AI – Malicious PE
Skyhigh (SWG)	BehavesLike.Win32.Backdoor.mc
Sophos	Troj/Agent-JZZ
SUPERAntiSpyware	Trojan.Agent/Gen-Bifrose
Symantec	Infostealer
TACHYON	Trojan/W32.Agent.29085.M
Tencent	Trojan.Win32.Agent.bcn
Trapmine	Malicious.high.mi.score
Trellix (ENS)	BackDoor-CEP.w
Trellix (HX)	Generic.mg.36197700bab421d1
TrendMicro	BKDR_BIFROSE.AFU
TrendMicro-HouseCall	BKDR_BIFROSE.AFU
Varist	W32/Backdoor.HNRS-5187
VBA32	Backdoor.Bifrose
VIPRE	Trojan.Crypt.BH
VirIT	Backdoor.Win32.Generic.PZ
ViRobot	Backdoor.Win32.Bifrose.10240
Webroot	W32.Malware.Gen
WithSecure	Backdoor:W32/Bifrose.gen!A
Xcitium	Backdoor.Win32.Bifrose.ADR@3xn7
Yandex	Trojan.GenAsa!HgSSZWe0hGI
Zillya	Backdoor.Bifrose.Win32.41900
ZoneAlarm by Check Point	Troj/Agent-JZZ
Zoner	Trojan.Win32.22108

## 7. Gh0st.exe 동적 분석 (xdbg, cuckoo)

### 7.1 xdbg 분석

Gh0stRAT의 CreateServer 기능을 이용하여 악성코드를 생성하면 Gh0st.exe Agent 프로그램이 생성된다. 이때 Gh0st.exe는 안티 바이러스의 파일 진단을 우회하고 분석을 방해하기 위해 자체 실행 압축 기술을 사용한다. 처음 실행 시 sub\_407028 루틴은 인코딩되어 있고 실행되면서 sub\_004073C0 루틴에서 디코딩된다.

00407028	75 5D CC 55 CC FA 73 80 77 3E 20 D6 20 D6 79 5F	u]íuiúus.w> ö öy_
00407038	6D 32 CB 94 76 BF 52 A2 55 B7 4C 97 4C BA 4F B5	m2E.v;RdU.L.L°oh
00407048	20 9D 45 A4 4E B3 4C E5 12 F8 44 BA 4C D6 76 BF	.E#N°Lå.øD°Lövå
00407058	52 A2 55 B7 4C 90 52 B3 45 D6 6C B9 41 B2 6C BF	RdU.L.R°EÖl'A=1ç
00407068	42 A4 41 A4 59 97 20 80 49 A4 54 A3 41 BA 70 A4	B¤A¤Y. .I¤T¤A¤p¤
00407078	4F A2 45 B5 54 D6 AB A3 C4 5D 7D DE AD 90 33 86	O¢EµTÖ«£Å»}p..3.
00407088	DF 85 30 5D D8 5B 66 D0 70 81 DF 85 34 5F 65 22	ß.0]Ø[fØp.ß.4_e"
00407098	AD 90 00 86 77 29 73 C2 A9 93 C4 5B 66 FA 70 81	...w)sAø.A[füp.
004070A8	DF 85 34 55 E6 EF A9 93 C0 80 77 29 73 C2 A9 93	ß.4Uæ1ø.A.w)sAø.
004070B8	F4 5D 23 69 20 E6 20 D6 4A 96 AB 9E 1C 81 AD E2	ô]#i æ öj «...å
004070C8	21 5F 55 26 2F 61 66 D0 AB 98 70 5F 65 3A AB 90	!_U&/afD«.p_e:«.
004070D8	14 87 70 5F 6D 2E DF 83 D4 53 EO 5F 65 DE 55 C3	..p_m.ß.ösa_ePUA
004070E8	4A 96 77 29 55 2E 70 29 75 22 A5 16 A9 93 28 D9	J.w)U.p)u"¥.©.(ü
004070F8	A4 BD 22 D6 20 5D 6E 82 AB E5 AB AB 28 5D E1 17	¤%"ö ]n.«å««(]å.
00407108	C9 D4 D3 73 AB 1E A3 37 23 25 84 5D 23 5D 68 EA	ÉööS«.£7#%.]#]hè
00407118	AD 52 21 2E 20 D6 20 5D 6D 3A A5 1F A9 93 D4 A8	.R!. ö ]m:¥.©.ö
00407028	75 5D	jne gh0st .407087
0040702A	CC	int3
0040702B	55	push ebp
0040702C	CC	int3
0040702D	FA	cli
0040702E	73 80	jae gh0st .406FB0
00407030	77 3E	ja gh0st .407070
00407032	20D6	and dh,d1
00407034	20D6	and dh,d1
00407036	79 5F	ins gh0st .407097
00407038	6D	int3
00407039	32CB	xor cl,b1
0040703B	94	xchg esp,eax
0040703C	76 BF	jbe gh0st .406FFD
0040703E	52	push edx
0040703F	A2 55B74C97	mov byte ptr ds:[974CB755],al
00407044	4C	dec esp
00407045	BA 4FB5209D	mov edx,9D20B54F
0040704A	45	inc ebp
0040704B	A4	movsb
0040704C	4E	dec esi
0040704D	B3 4C	mov b1,4C
0040704F	E5 12	sub eax,12

[그림11 - 인코딩 되어있는 바이너리 sub\_407028]



004073C0	55	push ebp
004073C1	8BEC	mov ebp,esp
004073C3	56	push esi
004073C4	33F6	xor esi,esi
004073C6	3975 0C	cmp dword ptr ss:[ebp+C],esi
004073C9	7E 1B	jle gh0st_4073E6
004073CB	8B45 08	mov eax,dword ptr ss:[ebp+8]
004073CE	33D2	xor edx,edx
004073D0	8D0C06	lea ecx,dword ptr ds:[esi+eax]
004073D3	8BC6	mov eax,esi
004073D5	F775 14	div dword ptr ss:[ebp+14]
004073D8	8B45 10	mov eax,dword ptr ss:[ebp+10]
004073DB	8A0402	mov al,byte ptr ds:[edx+eax]
004073DE	3001	xor byte ptr ds:[ecx],al
004073E0	46	inc esi
004073E1	3B75 0C	cmp esi,dword ptr ss:[ebp+C]
004073E4	7C E5	j1 gh0st_4073CB
004073E6	5E	pop esi
004073E7	5D	pop ebp
004073E8	C3	ret

[그림12 - 디코딩 함수 실행 sub\_4073C0]

00407028	55 8B EC 83	EC 2C 53 56	57 E8 00 00	00 00 59 89	U.i.i.SWw...Y.
00407038	4D E4 EB 42	56 69 72 74	75 61 6C 41	6C 6C 6F 63	MæBVirtualAlloc
00407048	00 4B 65 72	6E 65 6C 33	32 2E 64 6C	00 56 69	.Kernel32.dll.vi
00407058	72 74 75 61	6C 46 72 65	65 00 4C 6F	61 64 4C 69	rtualFree.LoadLi
00407068	62 72 61 72	79 41 00 56	69 72 74 75	61 6C 50 72	braryA.VirtualPr
00407078	6F 74 65 63	74 00 8B 75	E4 8B 5D 08	8D 46 13 50	otect..uä.]..F.P
00407088	FF 53 10 8B	F8 8D 46 06	50 57 FF 53	14 89 45 F4	ÿS..ø.F.PwÿS..Eö
00407098	8D 46 20 50	57 FF 53 14	89 45 E4 8D	46 2C 50 57	.F.PwÿS..Eä.F.Pw
004070A8	FF 53 14 83	C6 39 89 45	E0 56 57 FF	53 14 89 45	ÿS..æ9.EäVwÿS..E
004070B8	D4 88 03 BF	00 30 00 00	6A 40 8B 48	3C 57 8D 34	ö..ç..j@.H<W.4
004070C8	01 89 75 F0	0F B7 46 06	8B 4E 50 89	45 EC 8B 46	..uô..F..NP.Eì.F
004070D8	34 51 50 89	4D F8 FF 55	F4 85 C0 89	45 08 75 15	4QP.MøyUô.A.E.u.
004070E8	6A 40 57 FF	75 F8 50 FF	55 F4 85 C0	89 45 08 0F	j@wyuøPÿUô.A.E..
004070F8	84 6B 02 00	00 8B 4E 54	8B 33 8B 7D	08 8B C1 C1	.k...NT.3.}..AA
00407108	E9 02 F3 A5	8B C8 83 E1	03 F3 A4 8B	03 8B 48 3C	é.ó¥.É.á.ó¤..H<
00407118	8D 84 01 F8	00 00 00 8B	4D EC 85 C9	89 45 F4 7E	..ø...Mí.É.E~
0407028	55	push ebp			
0407029	8BEC	mov ebp,esp			
040702B	83EC 2C	sub esp,2C			
040702E	53	push ebx			
040702F	56	push esi			
0407030	57	push edi			
0407031	E8 00000000	call gh0st_407036			call \$0
0407036	59	pop ecx			
0407037	894D E4	mov dword ptr ss:[ebp-1C],ecx			
040703A	EB 42	jmp gh0st_40707E			
040703C	56	push esi			
040703D	6972 74 75616C41	imul esi,dword ptr ds:[edx+74],416C6175			
0407044	6C	insb			
0407045	6C	insb			
0407046	6F	outsb			
0407047	6300	arpl word ptr ds:[eax],ax			
0407049	4B	dec ebx			
040704A	65:72 6E	jb gh0st_4070BB			
040704D	65:6C	insb			
040704F	3332	xor esi,dword ptr ds:[edx]			
0407051	2E64:6C	insb			
0407054	6C	insb			

[그림13 - 디코딩된 sub\_407028 루틴]

00407499	FFD3	call ebx	
0040749B	8945 F4	mov dword ptr ss:[ebp- <span style="color:blue">C</span> ],eax	
0040749E	8B45 10	mov eax,dword ptr ss:[ebp+10]	
004074A1	8365 F0 00	and dword ptr ss:[ebp-10],0	
004074A5	83A5 74FAFFFF 00	and dword ptr ss:[ebp-58C],0	[ebp-F84]:&"SSHd,"
004074AC	8985 5CAFxFFFF	mov dword ptr ss:[ebp-5A4],eax	
004074B2	8D85 7CF0FFFF	lea eax,dword ptr ss:[ebp-F84]	
004074B8	8945 E8	mov dword ptr ss:[ebp-18],eax	
004074BB	8045 E4	lea eax,dword ptr ss:[ebp-1C]	
004074BE	50	push eax	
004074BF	89B5 80FAFFFF	mov dword ptr ss:[ebp-580],esi	
004074C5	8975 E4	mov dword ptr ss:[ebp-1C],esi	
004074C8	89BD 78FAFFFF	mov dword ptr ss:[ebp-588],edi	
004074CE	C785 7CAFxFFFF 5903	mov dword ptr ss:[ebp-584],359	
004074D8	C785 84FAFFFF 0060	mov dword ptr ss:[ebp-57C],6000	
004074E2	FFD7	call edi	call 00407028

[그림14 - 디코딩된 함수 실행]

[그림15 – VirtualAlloc → Write]

이후 디코딩된 sub\_407028 루틴에서 VirtualAlloc를 호출하여 0x10000000 번지에 가상 메모리를 생성하고 암호화된 Payload를 해당 메모리에 풀어 숨겨진 악성코드 함수를 실행한다. 이는 암호화된 바이러스를 회피하기 위한 행위임을 알 수 있다.



00407342	FF73 04	push dword ptr ds:[ebx+4]	
00407345	6A 01	push 1	
00407347	52	push edx	
00407348	FFD0	call eax	1000F820

[그림16 - 할당한 가상 메모리 함수 실행]

1000F953	47	inc edi	
1000F954	08C0	or al,al	
1000F956	74 DC	je 1000F934	
1000F958	89F9	mov ecx,edi	
1000F95A	79 07	jns 1000F963	
1000F95C	0FB707	movzx eax,word ptr ds:[edi]	
1000F95F	47	inc edi	
1000F960	50	push eax	
1000F961	47	inc edi	
1000F962	B9 5748F2AE	mov ecx,AEB24857	
1000F967	55	push ebp	
1000F968	FF96 CCF00000	call dword ptr ds:[esi+F0CC]	
1000F96E	09C0	or eax,eax	
1000F970	74 07	je 1000F979	
1000F972	8903	mov dword ptr ds:[ebx],eax	
1000F974	83C3 04	add ebx,4	
1000F977	EB D8	jmp 1000F951	

[그림17 - GetProcAddress를 호출하여 수입 테이블을 가져옴]

가상 메모리 함수 sub\_1000F8200 | 0x407348 주소에서 호출되고 호출된 가상 메모리는 우선적으로 사용하기 위한 API들을 GetProcAddress로 동적으로 API 주소를 가져오고 인코딩된 페이로드를 디코딩하여 할당한 가상 메모리에 덮어씌우는 작업을 수행한다. 이때 수입 테이블로 가져오는 API 들을 살펴보았을 때 악성코드 기능을 수행하는 API 들이 노출되는 것을 확인 할 수 있다.

```
WriteFile
IstrlenA
IstrcpyA
IstrcatA
GetFileSize
SetFilePointer
OpenMutexA
IstrcmpiA
Sleep
CreateProcessA
IstrcmpA
GetModuleFileNameA
GetModuleHandleA
GetProcAddress
LoadLibraryA
GetVersionExA
```



```
FileTimeToSystemTime  
SystemTimeToFileTime  
ReleaseMutex  
TerminateProcess  
GetCurrentProcess  
GetCurrentThreadId  
OpenProcess  
CompareFileTime  
GetSystemTimeAsFileTime  
GetSystemTime  
GetComputerNameA  
CreateToolhelp32Snapshot  
CopyFileA  
DeleteFileA  
SetFileAttributesA  
CreateDirectoryA  
GetWindowsDirectoryA  
GetSystemDirectoryA  
CreateMutexA  
CreateThread  
WriteProcessMemory  
VirtualProtectEx  
MoveFileA  
ReadProcessMemory  
WaitForSingleObject  
DuplicateHandle  
VirtualProtect  
VirtualFree  
RemoveDirectoryA  
FreeLibrary  
GetTickCount  
FindClose  
FindNextFileA  
FileTimeToLocalFileTime  
FindFirstFileA  
GetDriveTypeA  
GetTempPathA  
GetCurrentProcessId  
GetLocaleInfoA  
GetVolumeInformationA  
InterlockedDecrement
```

```
InterlockedIncrement  
LoadLibraryExA  
SetEndOfFile  
GetLocalTime  
HeapAlloc  
GetProcessHeap  
HeapFree  
Process32First  
Process32Next  
SetFileTime  
CreateFileA  
GetLastError  
ReadFile  
CloseHandle  
ResumeThread  
GetPriorityClass  
VirtualAlloc  
SetLastError  
GetFileAttributesExA  
CreateRemoteThread  
RegCreateKeyExA  
RegCloseKey  
RegEnumKeyExA  
RegOpenKeyExA  
RegSetValueExA  
OpenProcessToken  
LookupPrivilegeValueA  
AdjustTokenPrivileges  
RegDeleteValueA  
GetUserNameA  
RegQueryValueExA  
StretchBlt  
GetDIBColorTable  
DeleteObject  
SetStretchBltMode  
SelectObject  
CreateDIBSection  
CreateCompatibleDC  
CreateDCA  
DeleteDC  
GetDeviceCaps
```

```
__CxxFrameHandler
__CxxThrowException
_strerror
_strerror_s
_free
_malloc
_strchr
_strncpy
_strchr_s
_atoi
_snprintf
_rename
 strstr
_strcmp
ShellExecuteA
SHGetSpecialFolderPathA
SHDeleteKeyA
CallNextHookEx
SetWindowTextA
GetForegroundWindow
keybd_event
GetKeyboardState
VkKeyScanA
GetWindowTextA
IsWindow
SendMessageA
DestroyWindow
PostThreadMessageA
PeekMessageA
DispatchMessageA
wsprintfA
SetWindowsHookExA
RegisterClassExA
CreateWindowExA
GetMessageA
TranslateMessage
UnhookWindowsHookEx
PostQuitMessage
DefWindowProcA
GetKeyState
MapVirtualKeyA
```



```
ToAscii  
GetKeyNameTextA  
MessageBoxA  
IsWindowVisible  
GetWindowLongA  
GetWindowThreadProcessId  
GetKeyboardLayoutNameA  
EnumWindows  
ShowWindow  
mouse_event  
SetForegroundWindow  
InternetOpenA  
InternetOpenUrlA  
InternetReadFile  
InternetCloseHandle  
InternetGetConnectedState
```

10002554	55	push ebp
10002555	8BEC	mov ebp,esp
10002557	81EC C8040000	sub esp,4C8
1000255D	53	push ebx
1000255E	56	push esi
1000255F	33DB	xor ebx,ebx
10002561	57	push edi
10002562	53	push ebx
10002563	E8 D7200000	call 1000463F
10002568	A1 D8AA0010	mov eax,dword ptr ds:[1000AAD8]
1000256D	59	pop ecx
1000256E	6A 01	push 1
10002570	05 48010000	add eax,148
10002575	5E	pop esi
10002576	56	push esi
10002577	6A 1A	push 1A
10002579	50	push eax
1000257A	53	push ebx
1000257B	FF15 D8110010	call dword ptr ds:[<&SHGetSpecialFolderPathA>]
10002581	8B3D 68100010	mov edi,dword ptr ds:[<&lstrcpy>]
10002587	85C0	test eax,eax
10002589	75 18	jne 100025A3
10002588	A1 D8AA0010	mov eax,dword ptr ds:[1000AAD8]
10002590	68 10150010	push 10001510
10002595	05 48010000	add eax,148
1000259A	50	push eax
10002598	FF15 64100010	call dword ptr ds:[<&lstrcpy>]

[그림17 - 악성행위를 수행하는 함수 EntryPoint]



10002B2E	05 CB000000	add eax,cb push eax	eax:"Bif123" eax:"Bif123"
10002B33	50	push eax	
10002B34	6A 01	push 1	
10002B36	68 01001F00	push 1F0001	
<b>10002B38</b>	<b>FF15 74100010</b>	<b>call dword ptr ds:[&lt;&amp;openMutexA&gt;]</b>	
100048B0	50	push eax	
100048B1	6A 01	push 1	
100048B3	6A 00	push 0	
<b>100048B5</b>	<b>FF15 E4100010</b>	<b>call dword ptr ds:[&lt;&amp;CreateMutexA&gt;]</b>	
100048B8	0000 FFFFFF	mov byte ptr [eax],al	

[그림18 – Mutex를 열거하여 중복 실행 방지]

처음 실행 시 CreateMutexA를 호출하여 Bif123 라는 뮤텍스 객체를 생성한다. 생성하기 전에 OpenMutexA로 Bif123 라는 뮤텍스가 존재하는지 체크하여 악성코드가 중복으로 실행되는 것을 방지한다.

100029D7	8B3D 0C100010	mov edi,dword ptr ds:[<&RegOpenKeyExA>]	
100029DD	8D45 F4	lea eax,dword ptr ss:[ebp-C]	
100029E0	50	push eax	
100029E1	6A 08	push 8	
100029E3	53	push ebx	
100029E4	BE 02000080	mov esi,80000002	
100029E9	68 A4140010	push 100014A4	
100029EE	56	push esi	
<b>100029EF</b>	<b>FFD7</b>	<b>call edi</b>	
0019EA38	80000002	"SOFTWARE\\Microsoft\\Active Setup\\Installed Components"	
0019EA3C	100014A4		

[그림19 – 재시작 매커니즘에 등록하기 위해 Registry를 열기]

Master 서버와 Agent가 계속해서 세션을 유지하기 위해서 재부팅 시에도 gh0st.exe가 실행될 수 있도록 레지스트리 키를 열거하고 등록한다.



FF15 38110010	call dword ptr ds:[<&GetVolumeInformationA>]
8B3D D8110010	mov edi,dword ptr ds:[<&SHGetSpecialFolderPathA>]
53	push ebx
8D85 64D1FFFF	lea eax,dword ptr ss:[ebp-2E9C]
6A 08	push 8
50	push eax
53	push ebx
FFD7	call edi
53	push ebx
8D85 60D0FFFF	lea eax,dword ptr ss:[ebp-2FA0]
6A 10	push 10
50	push eax
53	push ebx
FFD7	call edi
53	push ebx
8D85 58CCFFFF	lea eax,dword ptr ss:[ebp-33A8]
6A 05	push 5
50	push eax
53	push ebx
FFD7	call edi
8D85 30FDFFFF	lea eax,dword ptr ss:[ebp-2D0]
6A 0C	push C
50	push eax
6A 5A	push 5A
68 00080000	push 800
FF15 34110010	call dword ptr ds:[<&GetLocaleInfoA>]
8D85 3CFFFFFF	lea eax,dword ptr ss:[ebp-C4]
50	push eax
FF15 60120010	call dword ptr ds:[<&GetKeyboardLayoutNameA>]
8D85 3CFFFFFF	lea eax,dword ptr ss:[ebp-C4]

[그림20 - 에이전트 정보를 갱신하기 위해 컴퓨터 정보를 추출]

에이전트 정보를 갱신하기 위해 GetComputerNameA, GetLocaleInfoA, GetKeyboardLayoutNameA... 정보 수집을 하여 Master 서버에 전송한다.

100053CA	53	push ebx
100053CB	6A 01	push 1
100053CD	6A 02	push 2
<b>100053CF</b>	<b>FF15 A8120010</b>	<b>call dword ptr ds:[&lt;&amp;socket&gt;]</b>
100053D5	8BF0	mov esi, eax
100053D7	83FE FF	cmp esi, FFFFFFFF
100053DA	74 2D	je 10005409
100053DC	FF75 10	push dword ptr ss:[ebp+10]
100053DF	8D45 F0	lea eax,dword ptr ss:[ebp-10]
100053E2	57	push edi
100053E3	50	push eax
100053E4	E8 4AFFFFFF	call 10005333
100053E9	83C4 0C	add esp,C
100053EC	84C0	test al,al
100053EE	74 19	je 10005409
100053F0	8D45 F0	lea eax,dword ptr ss:[ebp-10]
100053F3	6A 10	push 10
100053F5	50	push eax
100053F6	56	push esi
100053F7	FF15 AC120010	call dword ptr ds:[<&connect>]
100053FD	83F8 FF	cmp eax, FFFFFFFF
10005400	75 0E	jne 10005410
10005402	56	push esi
10005403	FF15 B0120010	call dword ptr ds:[<&closesocket>]
10005403	22C0	ret

[그림21 - Agent에서 Master과 연결하기 위해 소켓 생성후 연결 요청]

Master 서버에 접속하기 위해 TCP/IP 프로토콜을 이용해 소켓을 생성하여 Connect 요청을 한다.

## 7.2 cuckoo 샌드박스 결과

### Score

This file is **very suspicious**, with a score of **10 out of 10!**

[그림22 - cuckoo 샌드박스 결과 멀웨어 지수 10점]  
<cuckoo.cert.ee/analysis/6181267/summary/>

Signatures	
Allocates read-write-execute memory (usually to unpack itself) (1 event)	>
One or more potentially interesting buffers were extracted, these generally contain injected code, configuration data, etc.	>
Searches running processes potentially to identify processes for sandbox evasion, code injection or memory dumping (1 event)	>
The binary likely contains encrypted or compressed data indicative of a packer (2 events)	>
Allocates execute permission to another process indicative of possible code injection (12 events)	>
Creates a thread using CreateRemoteThread in a non-child process indicative of process injection (3 events)	>
Manipulates memory of a non-child process indicative of process injection (13 events)	>
Potential code injection by writing to the memory of another process (6 events)	>
Expresses interest in specific running processes (1 event)	>
File has been identified by 13 AntiVirus engine on IRMA as malicious (13 events)	>
File has been identified by 67 AntiVirus engines on VirusTotal as malicious (50 out of 67 events)	>

[그림23 - cuckoo 샌드박스 결과]  
<cuckoo.cert.ee/analysis/6181267/summary/>

## 8. 탐지 Signature

기능	Text	HEX
초기 연결	초기 연결 시 중복 되는 시그니처	05 b4 01 01 04 02

```
alert tcp any any -> any any (
    msg:"gh0st RAT initial handshake detected";
    flow:established,to_server;
    content:"|05 B4 01 01 04 02|";
    offset:0; depth:6;
    classtype:trojan-activity;
    sid:1001001;
    rev:2;
)
```

```
C:\#Snort>type C:\#Snort\#rules\local.rules
alert tcp any any -> any any (msg:"Initial connection signature detected"; content:"|05 B4 01 01 04 02|"; sid:1000001; rev:1;)
```

[그림24 – local.rules]