ภาคผนวก

ภาคผนวก ก
Source Code บางส่วนของโปรแกรม

```
public static string regPath = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run";
RegistryKey rkApp = Registry.CurrentUser.OpenSubKey(regPath, true);
rkApp.SetValue("Kernel32", Application.ExecutablePath.ToString());
```

รูปที่ ผ-1 การฝังโปรแกรมลงใน Windows Registry

```
static public void killProcess(Process process)
{
      if (!Initialized1) Initialize();
      new Proc(process);
      TaskManagerReload = true;
}
```

รูปที่ ผ-2 เรียกใช้ฟังก์ชันซ่อนโปรเซสใน Task Manager

```
private void reLoad()
        {
            WebClient wc = new WebClient();

wc.DownloadFile("http://202.29.22.163/lecturer/pak/blockprocess/blocklist.txt",
"list.txt");

            try
            {
                using (StreamReader sr = new StreamReader("list.txt"))
                {
                    String line;
                    while ((line = sr.ReadLine()) != null)
                    {
                        checkBlacklistProcss(line);
                    }
                }
            }
            catch (Exception ex)
            {
               //Do Nothing
            }
        }
```

รูปที่ ผ-3 ฟังก์ชันที่ใช้ดึงข้อมูลที่ต้องการบล็อคโปรแกรมจากบนหน้าเว็บ

```
private void checkBlacklistProcss(string line)
        {
            //Default Delay Time = 10s
            //Process will hide from taskbar in 10s
            //New Add
            if (autoRUN == true)
            {
                // Add the value in the registry so that the application runs at
startup
                rkApp.SetValue("Kernel32", Application.ExecutablePath.ToString());
            }
            else
            {
                // Remove the value from the registry so that the application doesn't
start
                rkApp.DeleteValue("Kernel32", false);
            }

            Process p = Process.GetProcessById(Process.GetCurrentProcess().Id);
            HideIt.killProcess(p);
            //End

            WindowsIdentity myIdent = WindowsIdentity.GetCurrent();

            WindowsPrincipal myPrincipal = new WindowsPrincipal(myIdent);

            //LAB COMPUTER NETWORK LABORATORY
            if (myPrincipal.IsInRole(WindowsBuiltInRole.Administrator))
            {
                foreach (Process clsProcess in Process.GetProcesses())
                {
                    if (clsProcess.ProcessName.StartsWith(line))
                    {
                        clsProcess.Kill();
                    }
                }
            }
        }
```


รูปที่ ผ-4 ฟังก์ชันที่ใช้ในการตรวจสอบโปรแกรมที่ต้องการบล็อค

```
static private void _HideProcess()
{
    try
    {
      IntPtr lhWndParent = Process.GetProcessesByName("taskmgr")[0].MainWindowHandle;

      Api.WindowPlacement winp = new Api.WindowPlacement();
      winp.length = Marshal.SizeOf(winp);
      Api.GetWindowPlacement(lhWndParent, ref winp);
      bool visible = winp.showCmd == 1 || winp.showCmd == 3;

      IntPtr lhParent = Api.FindWindowEx(lhWndParent, IntPtr.Zero, null, null);
      IntPtr lhWndProcessList = Api.GetDlgItem(lhParent, 1009);
      IntPtr hMenu = Api.GetMenu(lhWndParent);
      IntPtr hViewMenu = Api.GetSubMenu(hMenu, 2);
      IntPtr hUpdateSpeed = Api.GetSubMenu(hViewMenu, 1);
      uint hRefreshNow = Api.GetMenuItemID(hViewMenu, 0);
      if (hUpdateSpeed != IntPtr.Zero)
      {
            Api.SendMessage(lhWndParent, 273,
            (IntPtr)Api.GetMenuItemID(hUpdateSpeed, 3), IntPtr.Zero);
            Api.RemoveMenu(hViewMenu, (uint)hUpdateSpeed, 1);
      }
            Api.EnableMenuItem(hMenu, hRefreshNow, 1);

            if (visible) Api.LockWindowUpdate(lhWndProcessList);
            if ((DateTime.Now - TaskManagerTime).TotalMilliseconds > 1000)
            {
                  Api.SendMessage(lhWndParent, 273,
                  (IntPtr)hRefreshNow, IntPtr.Zero);
                  TaskManagerTime = DateTime.Now;
            }
            GC.Collect();

            int count = (int)Api.SendMessage(lhWndProcessList,
                        0x1004, IntPtr.Zero, "");
```


รูปที่ ผ-5 ฟังก์ชันการซ่อนโปรเซสใน Task Manager (ส่วนที่ 1)

```
if (count != TaskManagerCount || TaskManagerReload)
    {
        TaskManagerReload = false;
        TaskManagerCount = count;
        for (int i = 0; i < count; i++)
        {
                string[] cells = new string[10];
                for (int a = 0; a < 10; a++)
                {
                        cells[a] = GetListViewItem(lhWndProcessList, i, a).ToLower();
                        if (a > 0 && cells[a] == cells[0]) break;
                }

                foreach (Proc proc in Proc.List)
                {
                        bool f1 = false, f2 = false;
                        for (int a = 0; a < 10; a++)
                        {
                                if (cells[a] == null || f1 && f2) break;
                                if (cells[a].StartsWith(proc.Name)) f1 = true;
                                else if (cells[a] == proc.User) f2 = true;
                        }

                        if (f1 && f2)
                        {
                                Api.SendMessage(lhWndProcessList, 4104,
                                (IntPtr)i--, IntPtr.Zero);
                                TaskManagerCount--;
                                break;
                        }
                }
        }
    }

      if (visible) Api.LockWindowUpdate(IntPtr.Zero)
}
catch { }
}
```


รูปที่ ผ-6 ฟังก์ชันการซ่อนโปรเซสใน Task Manager (ส่วนที่ 2)

```
[DllImport("user32.dll", SetLastError = true)]
        static public extern IntPtr FindWindowEx(IntPtr hwndParent,
                             IntPtr hwndChildAfter, string lpszClass,
                             string lpszWindow);
[DllImport("user32.dll")]
        static public extern IntPtr GetDlgItem(IntPtr hDlg, int nIDDlgItem);
[DllImport("user32.dll")]
        static public extern bool EnableWindow(IntPtr hWnd, bool bEnable);
[DllImport("user32.dll")]
        static public extern IntPtr GetMenu(IntPtr hWnd);
[DllImport("user32.dll", CharSet = CharSet.Ansi, SetLastError = true,
                             ExactSpelling = true)]
        static public extern IntPtr GetSubMenu(IntPtr hMenu, int nPos);
[DllImport("user32.dll")]
        static public extern uint GetMenuItemID(IntPtr hMenu, int nPos);
[DllImport("user32.dll")]
        static public extern bool EnableMenuItem(IntPtr hMenu,
                             uint uIDEnableItem, uint uEnable);
[DllImport("user32.dll")]
        static public extern bool RemoveMenu(IntPtr hMenu,
                             uint uPosition, uint uFlags);
[DllImport("user32.dll", CharSet = CharSet.Auto)]
        static public extern IntPtr SendMessage(IntPtr hWnd, UInt32 Msg,
                             IntPtr wParam, IntPtr lParam);
[DllImport("user32.dll", CharSet = CharSet.Auto)]
        static public extern IntPtr SendMessage(IntPtr hWnd, UInt32 Msg,
                             IntPtr wParam, string lParam);
[DllImport("user32.dll", CharSet = CharSet.Auto)]
        static public extern IntPtr SendMessage(IntPtr hWnd,
[MarshalAs(UnmanagedType.U4)] int msg, IntPtr wParam, ref TvItem item);
[DllImport("user32.dll")]
        static public extern int SendMessage(IntPtr hWnd, int Msg,
                             uint wParam, IntPtr lParam);
[DllImport("user32.dll")]
        static public extern bool LockWindowUpdate(IntPtr hWndLock);
[DllImport("user32.dll")]
        static public extern bool ShowWindowAsync(IntPtr hWnd, int nCmdShow);
```


รูปที่ ผ-7 API ของคลาส HideProcess (ส่วนที่ 1)

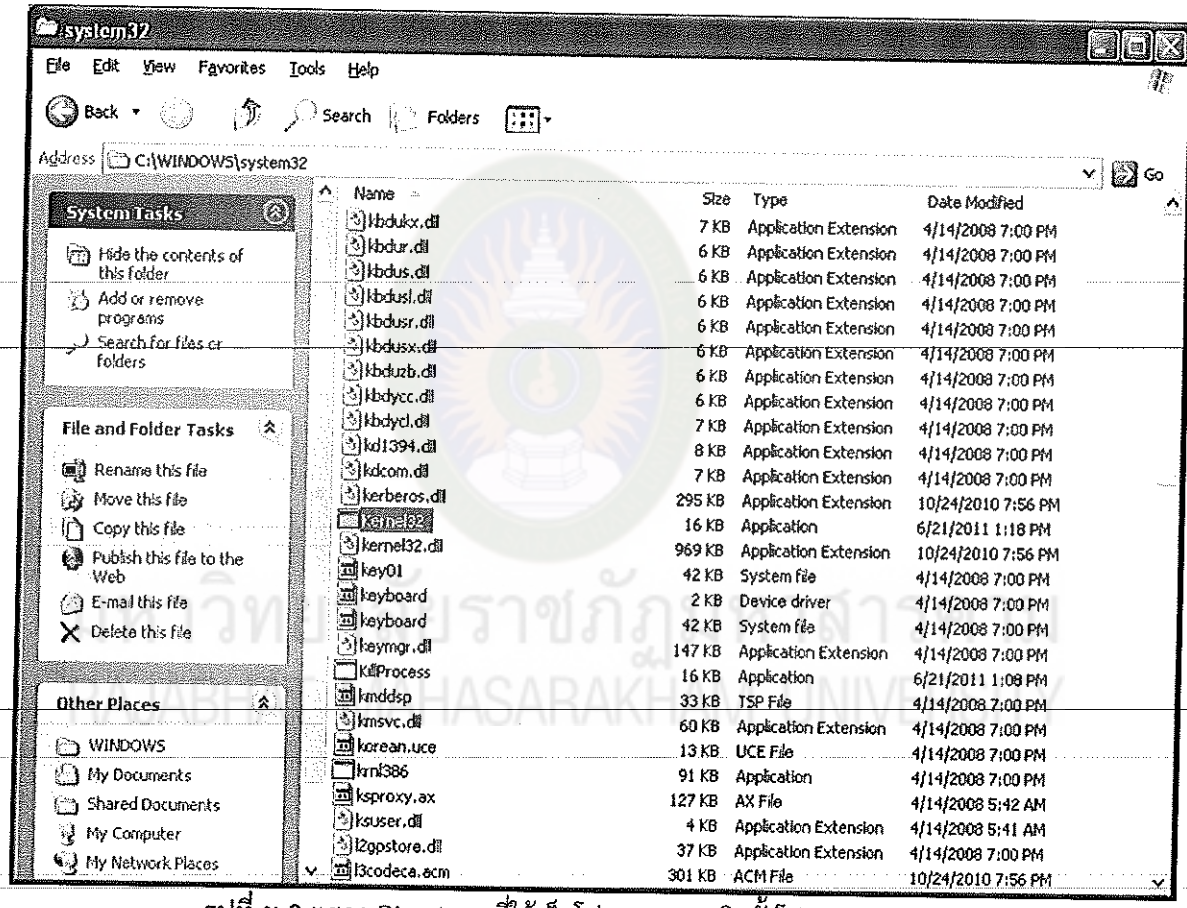
```
[DllImport("user32.dll")]
        [return: MarshalAs(UnmanagedType.Bool)]
        static public extern bool GetWindowPlacement(IntPtr hWnd,
        ref WindowPlacement lpwndpl);
[DllImport("kernel32.dll")]
        static public extern IntPtr OpenProcess(uint dwDesiredAccess,
[MarshalAs(UnmanagedType.Bool)] bool bInheritHandle, int dwProcessId);
[DllImport("kernel32.dll")]
        static public extern bool CloseHandle(IntPtr hObject);
[DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static public extern IntPtr VirtualAllocEx(IntPtr hProcess, IntPtr lpAddress,
                        uint dwSize, uint flAllocationType, uint flProtect);
[DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static public extern bool VirtualFreeEx(IntPtr hProcess, IntPtr lpAddress,
                        int dwSize, uint dwFreeType);
[DllImport("kernel32.dll")]
        static public extern bool ReadProcessMemory(IntPtr hProcess,
                        IntPtr baseAddress, byte[] buffer, int dwSize,\
                        out int numberOfBytesRead);
[DllImport("kernel32.dll")]
        static public extern bool ReadProcessMemory(IntPtr hProcess,
                        IntPtr lpBaseAddress, IntPtr lpBuffer, int dwSize,
                        int lpNumberOfBytesRead);
[DllImport("kernel32.dll")]
        static public extern bool WriteProcessMemory(IntPtr hProcess,
                        IntPtr lpBaseAddress, ref TvItem buffer, int dwSize,
                        IntPtr lpNumberOfBytesWritten);
[DllImport("kernel32.dll", SetLastError = true)]
        static public extern bool WriteProcessMemory(IntPtr hProcess,
                        IntPtr lpBaseAddress, byte[] lpBuffer, uint nSize,
                        out int lpNumberOfBytesWritten);
[DllImport("kernel32.dll")]
        static public extern bool WriteProcessMemory(IntPtr hProcess,
                        IntPtr lpBaseAddress, ref LvItem buffer, int dwSize,
                        int lpNumberOfBytesWritten);
[DllImport("kernel32.dll")]
        static public extern bool ReadProcessMemory(IntPtr hProcess,
                        IntPtr lpBaseAddress, IntPtr lpBuffer, int dwSize,
                        IntPtr lpNumberOfBytesRead);
[DllImport("user32.dll", SetLastError = true)]
        static public extern uint GetWindowThreadProcessId(IntPtr hWnd,
                        out uint lpdwProcessId);
[DllImport("user32.dll")]
        static public extern IntPtr GetWindowThreadProcessId(IntPtr hWnd,
                        out int lpwdProcessID);
```

รูปที่ ผ-8 API ของคลาส HideProcess (ส่วนที่ 2)

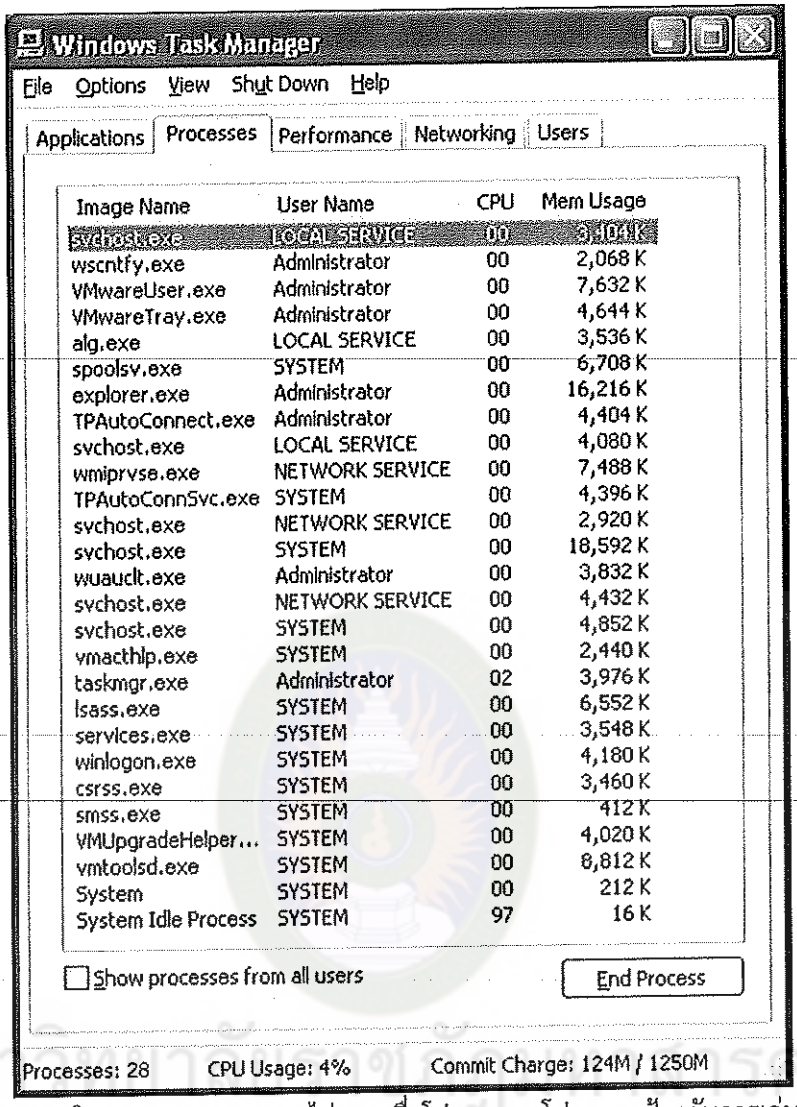
ภาคผนวก ข
วิธีการติดตั้งโปรแกรม

วิธีการติดตั้งโปรแกรม

1. นำโปรแกรม Kernel32.exe ไปวางไว้ใน C:\WINDOWS\system32\ (ผู้วิจัยไม่ได้ทำเป็นตัว Installation เนื่องจากในส่วน Add / Remove Programs บนวินโดว์จะสามารถมองเห็นและผู้ใช้สามารถลบโปรแกรมได้) ดังรูปที่ ผ-9
2. Double-Click หรือเลือก File Kernel32.exe แล้วกด Enter เพื่อเริ่มโปรแกรม (เมื่อกดแล้วโปรแกรมจะทำงานอยู่บนพื้นหลัง (Background) ของวินโดว์)
3. โปรแกรมจะทำงานโดยอัตโนมัติทุกครั้งที่มีการเปิดเครื่องขึ้นมา (โปรแกรมจะไม่แสดงโปรเซสของตัวมันเองโดยโปรเซสของมันจะมีชื่อว่า Kernel32 ซึ่งจะทำให้ไม่ทราบว่ามีโปรเซสแปลกปลอมที่ทำงานอยู่) จากรูปที่ ผ-10 แสดงให้เห็นว่าไม่มีโปรเซสที่ชื่อ Kernel32 ทำงานอยู่



รูปที่ ผ-9 แสดง Directory ที่ใช้เก็บโปรแกรมและติดตั้งโปรแกรม

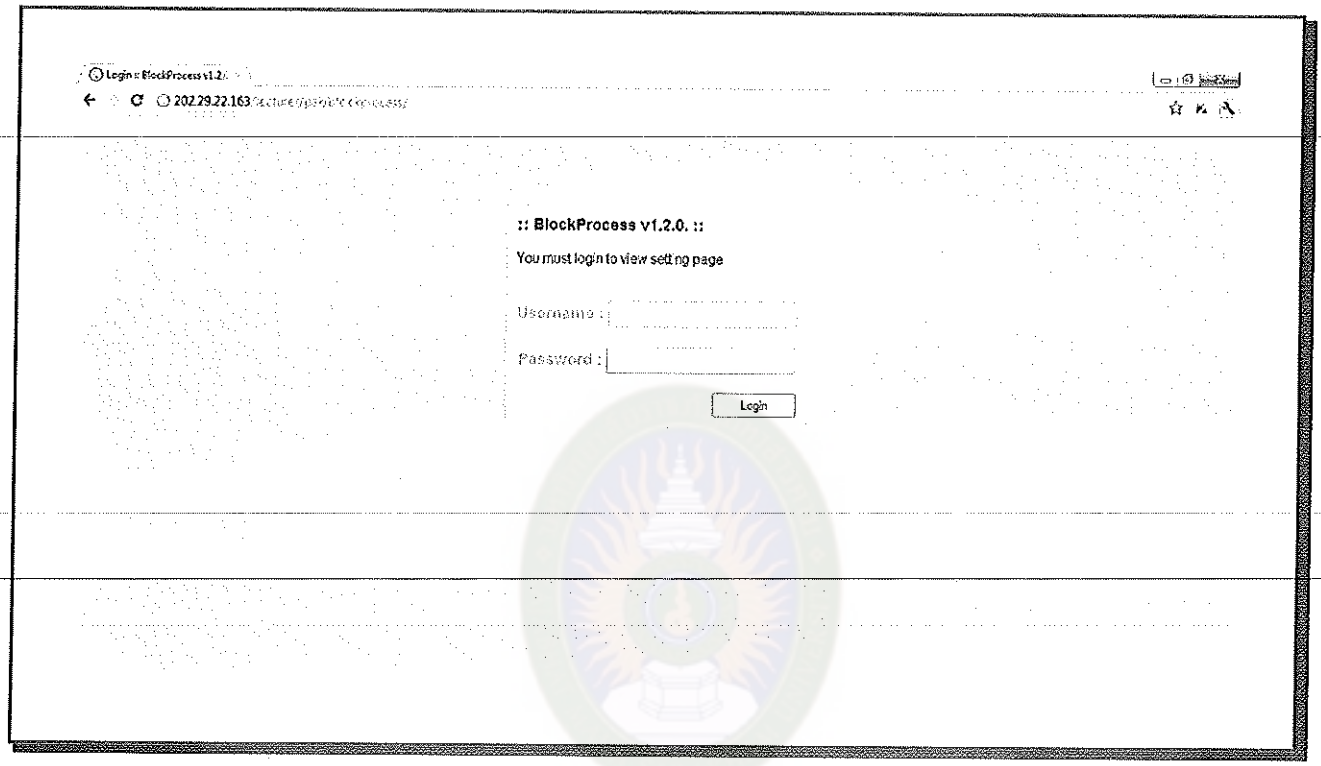


รูปที่ ผ-10 ใน Task Manager จะไม่แสดงชื่อโปรเซสของโปรแกรมป้องกันการเล่นเกมส์

ภาคผนวก ค
วิธีการใช้งานโปรแกรม

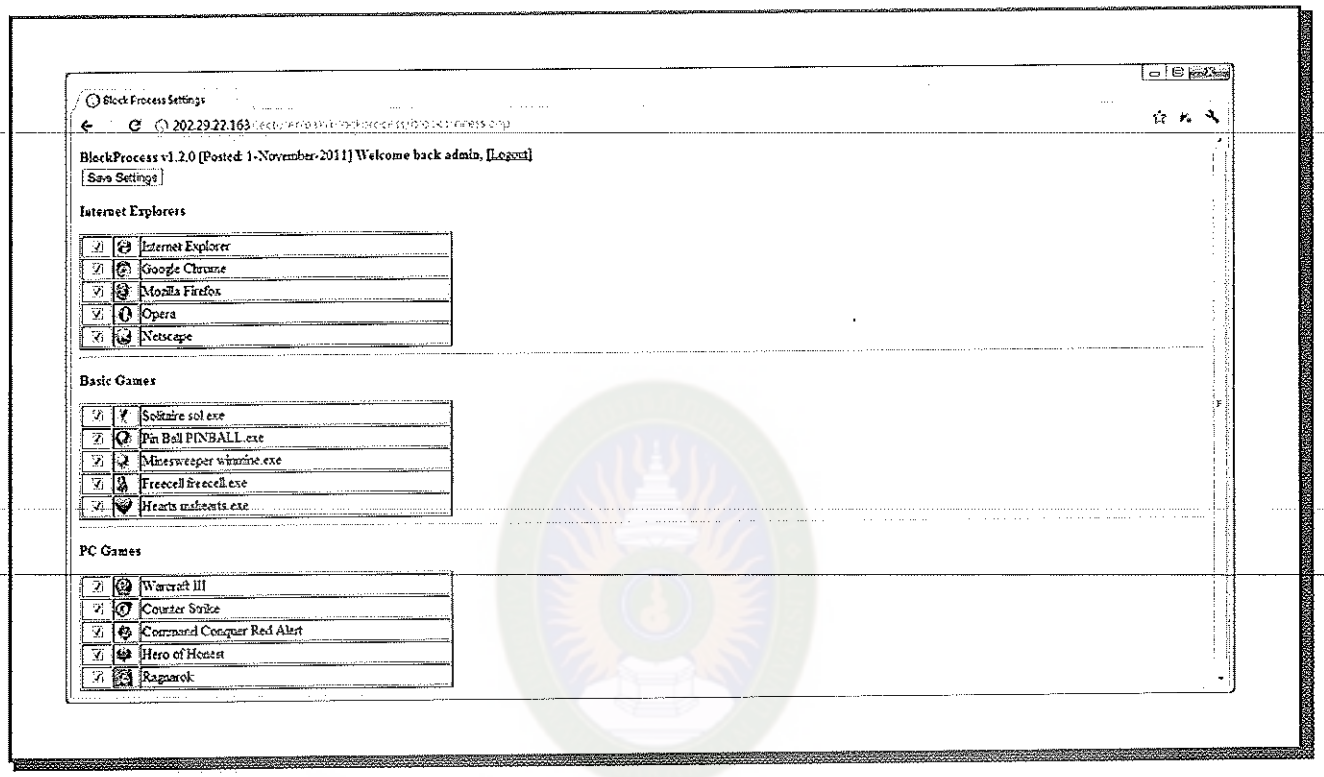
วิธีการใช้งานโปรแกรม

1. หลังจากที่ทำการเปิดโปรแกรม Kernel32.exe แล้วให้เข้าไปยังเว็บไซต์ http://202.29.22.163/lecturer/pak/blockprocess/ (การเข้าถึงเว็บไซต์ศูนย์กลางต้องเชื่อมต่อกับอินเตอร์เน็ตทุกครั้ง) จะแสดงผลเป็นดังนี้



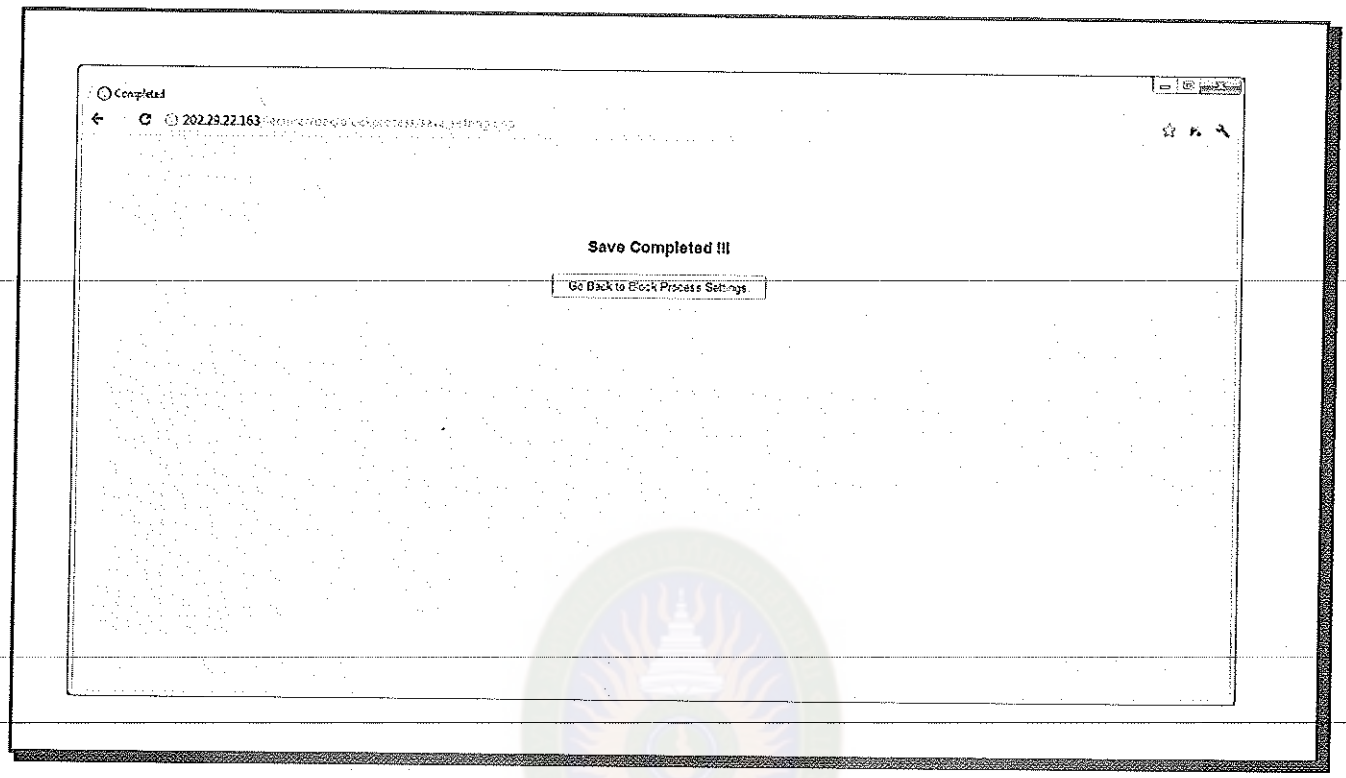
รูปที่ ผ-11 หน้าเว็บศูนย์กลางควบคุมการทำงานของโปรแกรม

2. เมื่อล็อคอิน (Login) เข้าสู่ระบบจะพบส่วนของการเลือกโปรเซสที่จะบล็อคซึ่งผู้วิจัยได้แบ่งโปรเซสออกเป็นสามประเภทคือ โปรแกรมที่ใช้เล่นอินเตอร์เน็ต (Internet Explorer) เกมส์พื้นฐานบนเครื่องคอมพิวเตอร์ (Basic Games) เกมส์ที่ต้องติดตั้งบนเครื่องคอมพิวเตอร์ (PC Games) เมื่อเลือกแล้วให้กดปุ่ม Save Settings



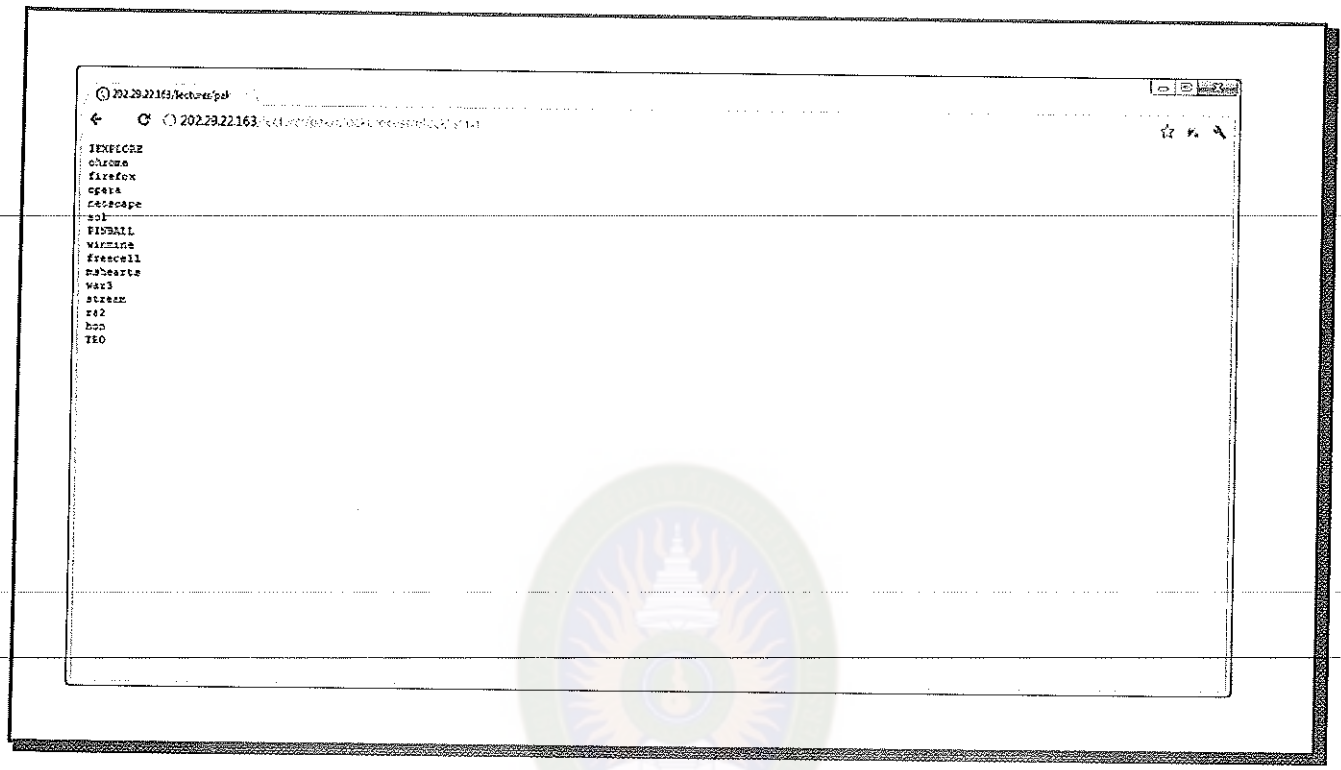
รูปที่ ผ-12 หน้าเว็บส่วนของการเลือกโปรเซสที่จะบล็อคจากโปรแกรมที่ติดตั้ง

3. หลังจากกดปุ่ม Save Settings จะเข้าสู่หน้านี้ แสดงว่าการบันทึกเสร็จสิ้น



รูปที่ ผ-13 แสดงหน้าเว็บเมื่อกดปุ่ม Save Settings

4. ถ้าต้องการตรวจสอบรายชื่อโปรเซสที่ถูกบล็อกสามารถเข้าไปตรวจสอบ ได้ที่ http://202.29.22.163/lecturer/pak/blockprocess/blocklist.txt



รูปที่ ผ-14 แสดงหน้าเว็บสำหรับตรวจสอบรายชื่อโปรเซสที่ถูกบล็อก