

웹캠을 이용한 노트북 도난 경보 시스템 (S.I.L.C)

팀명 : It's Real

담당 교수님 : 이병천 교수님

조장 : 최용현 (07)

조원 : 최철원 (07)

신준섭 (07)

김명진 (08)

2013. 5.
정보보호학과

목 차

I . 서론

- I - 1. 개발 목적
- I - 2. 관련 연구
 - 1) OpenCV 라이브러리
 - 2) DVR 시스템
 - 3) Winpcap 소켓프로그램

II . 본론

- II-1. 개발 환경
- II-2. 개발 일정
- II-3. 프로그램 구성
 - 1-1) Capture 프로그램
 - 1-2) Send 프로그램
 - 1-3) Receive 프로그램
 - 2) 프로그램 실행 구성

III . 결론

- III-1. 참고문헌
- III-2. 발표 PPT 자료

I. 서론

I-1 개발 목적

Notebook(Labtop), TabletPC 등의 발전으로 이 전자 장비를 휴대하는 사람들이 많습니다. 앞으로도 꾸준한 발전이 계속 돼서 많은 이용자가 계속 생길 것 입니다. 하지만 이에 반에 이 휴대용 전자장비등은 도난에 있어서는 매우 취약한 점을 가지고 있습니다. 그래서 저희 조는 이런 일의 사후처리와 예방에 도움을 줄 수 있는 도난방지 프로그램을 개발하기로 하였습니다.

* 관련 뉴스

NASA 노트북 도난에 정보 유출 사용자 1만 이를 듯

[AD] 데이터센터를 위한 프로세싱 파워, Windows Server 2012

미 항공우주국(NASA)이 노트북 도난에 따라 정보 유출 피해를 입을 수 있는 사용자가 1만명에 이른다고 추가 설명서를 발표했다.

15일(현지시각) 오전 로이터 등은 2주 전 NASA 임원의 노트북이 도난당하면서 노트북 내에 임직원 및 협력사 인력의 사회보장번호와 연락처 등 중요한 개인정보 파일이 저장되어 있다고 보도했다. 노트북은 비밀번호가 설정되어 있지만 하드드라이브는 암호화되어 있지 않은 상태다.



이후 NASA는 업무용 노트북에 대해 내달 21일까지 풀디스크 암호화를 적용하겠다고 보안 강화 조치를 발표했으며 다시 이날 오후 추가적인 공식 설명서를 통해 최소 1만명이 정보 유출 피해자가 될 수 있다고 시인했다.

NASA는 노트북이 도난당하고 2주 후에야 이 사실을 임직원들에 알렸는데 NASA측 대변인은 “도난 즉시 노트북 회수 및 데이터 보호를 위해 수사기관과 협력해 왔으며 NASA의 IT전문가와 보안 관계자들이 철저한 데이터 분석을 수행하고 있다”고 주장했다.

“etnews.com 박현선 기사 뉴스 인용”

출처 사이트 : http://www.etnews.com/news/computing/security/2677970_1477.html

I-2 관련 연구

1) OpenCV 라이브러리

Open Computer Vision 의 약자로 오픈 소스 컴퓨터 비전 C 라이브러리이다. 원래는 인텔에서 개발하였으며, Window, Linux 등의 여러 플랫폼에서 사용할 수 있다. 실시간 이미지 프로세싱에 중점을 둔 라이브러리이다. 인텔 CPU에서 사용하게 되면 속도의 향상을 볼 수 있는 Intel Performance Primitives(IPP)를 지원한다. 응용분야로는 인간과 컴퓨터 상호 작용(HCI), 물체 인식, 안면 인식, 모바일 로봇틱스, 제스처 인식 등 여러 방면으로 사용된다.

2) DVR 시스템

아날로그 영상 감시 장비인 CCTV를 대체하는 디지털 방식의 영상 감시 장비. CCTV에 비해 화질이 뛰어난 점 외에도 컴퓨터의 하드디스크를 저장 매체로 사용하기 때문에 녹화테이프를 교체할 필요가 없고 인터넷을 통한 실시간 영상 전송 및 원격지 감시 기능이 있어 네트워크로 통합화하고 있는 정부 및 기관, 기업체들에게 가장 적절한 영상 감시 시스템으로 평가받고 있다. DVR은 전 세계적으로 연간 3조5000억원 규모에 달하는 CCTV 시장을 향후 5년 내에 대체할 것으로 전망되고 있으며 장기적으로는 Home Security 등에 응용돼 개인시장으로도 확대될 가능성이 높아 성장 잠재력이 매우 높은 분야다.

3) winpcap 소켓프로그램

winpcap이란 리눅스/유닉스에서 이미 유명한 Libpcap을 윈도우 개발 환경에 맞게 포팅(반도체 디바이스에 플라스틱 수지를 씌워서 패키징하는 수법의 하나)한 것이다. win32 플랫폼에서 네트워크 패킷 캡처를 위한 아키텍처이다. 커널레벨에서 동작하는 저수준의 동적 라이브러리인 패킷 필터와 유저레벨에서 동작하는 고수준의 시스템 종속적인 라이브러리를 포함한다. 라이브러리라는 말을 대신하여 아키텍처라 부르는 이유는 패킷캡처는 네트워크 어댑터와 OS사이에서 엄격한 규칙이 적용하는 저수준의 매커니즘이다. 특별히 이러한 네트워크 작업은 단순한 라이브러리로 충분하지 않다. 처음으로 캡처 시스템은 네트워크상에서 전송되는 저수준의 데이터를 읽기 위해서 OS의 시스템 프로토콜을 스택을 우회하는 방법이 필요하다. 이것은 OS의 커널 레벨에서 네트워크 인터페이스 드라이버에서 직접적으로 동작하는 부분이 요구된다. 이러한 부분은 시스템에 종속이며 winpcap은 NPF라는 디바이스 드라이버를 통해 구현되었다. winpcap은 윈도우 버전별로 다른 드라이버 버전을 제공한다. 이 드라이버는 두가지 중요한 기능을 제공하는데 패킷캡처와 패킷 인젝션이다. 또한 보다 발전된 필터 시스템과 모니터링 엔진을 제공한다.

II. 본 론

II-1. 개발 환경

운영 체제 : Microsoft Windows 7 32bit, 64bit

개발 도구 : Microsoft Visual studio 2010

프로그래밍 언어 : C/C++, MFC

라이브러리 : cv (기본 컴퓨터 비전, 영상처리 주요 알고리즘 관련 라이브러리)

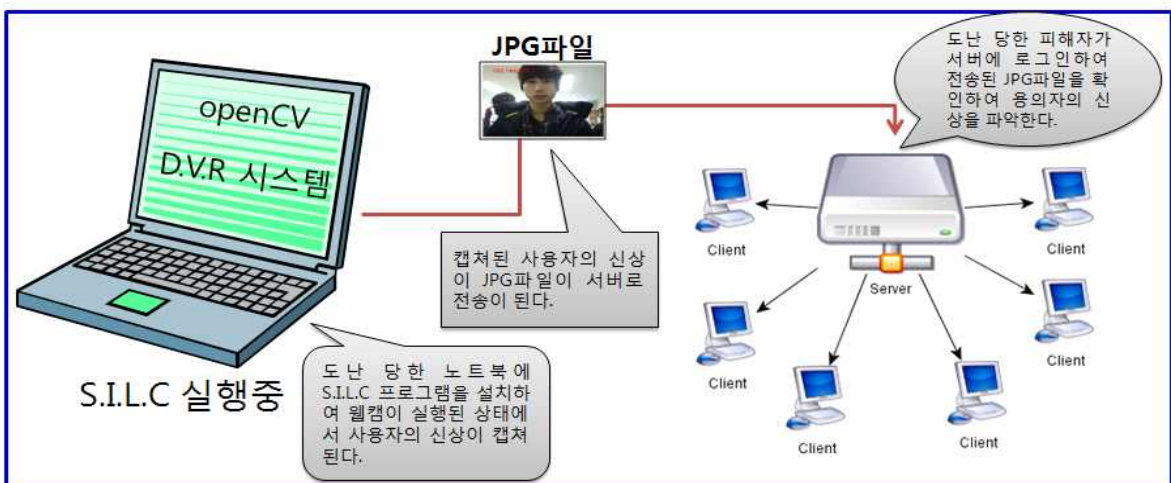
highgui (영상 읽기, 윈도우 관련, 카메라/동영상 제어)

cvcore (영상 행렬을 포함한 다양한 데이터 구조 지원 라이브러리)

II-2. 개발 일정

월별 내용	9월	10월	11월	12월	1월	2월	3월	4월	5월
조원 역할 분담 및 월별 계획 편성									
자료 수집 및 각자 취약한 점 연구									
중간 점검 및 오류 수정									
수정 완료 및 마무리 작업									
최종 보고서 작성									

II-3. 시스템 구상도



II-3. 프로그램 구성도

1-1) Capture 프로그램

```
#include "stdafx.h"
#include "MFCtest.h"
#include "MFCtestDlg.h"
#include "afxdialogex.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

    enum { IDD = IDD_ABOUTBOX };

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support

    // Implementation
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()
```

```

CMFCtestDlg::CMFCtestDlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(CMFCtestDlg::IDD, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    m_pImage = NULL;
}

void CMFCtestDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_OPENCV_PIG, m_pic);
}

BEGIN_MESSAGE_MAP(CMFCtestDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_WM_TIMER()
    ON_WM_DESTROY()
    ON_WM_WINDOWPOSCHANGING()
END_MESSAGE_MAP()

// CMFCtestDlg message handlers

BOOL CMFCtestDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    ASSERT((IDM_ABOUTBOX & 0xFFFF) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
    }
}

```

```

        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);         // Set small icon

    // TODO: Add extra initialization here

    capture = cvCreateCameraCapture(1); //1번캠으로부터 캡처를 실행
    m_Timer = FALSE;
    i = 0;

    CMFCtestDlg::start();    //스타트 정의를 실행
    AfxGetMainWnd()->PostMessage(WM_CLOSE, 0, 0); //다이얼로그를 백그라운드로 실행

    return true; // return TRUE unless you set the focus to a control
}

void CMFCtestDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CMFCtestDlg::OnPaint()
{

```



```

        if (IsIconic())
        {
            CPaintDC dc(this); // device context for painting

            S e n d M e s s a g e ( W M _ I C O N E R A S E B K G N D ,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

            // Center icon in client rectangle
            int cxIcon = GetSystemMetrics(SM_CXICON);
            int cyIcon = GetSystemMetrics(SM_CYICON);
            CRect rect;
            GetClientRect(&rect);
            int x = (rect.Width() - cxIcon + 1) / 2;
            int y = (rect.Height() - cyIcon + 1) / 2;

            // Draw the icon
            dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
            CDialogEx::OnPaint();
        }
    }

HCURSOR CMFCtestDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CMFCtestDlg::start()
{
    IplImage* image = 0; //매 프레임당 영상처리를 한다.
    CvCapture* capture = cvCaptureFromCAM(1); //1번 캠을 불러온다.
    cvNamedWindow( "Camera", 1 ); //그려줄 윈도우이름을 정의 한다.
    cvResizeWindow( "Camera", 320, 240 ); //그려줄 윈도우사이즈를 정의 한다.

    CTime time; //파일저장위한 시간변수

```

```

CString filename;          //파일명

CvFont* font = new CvFont;          // OpenCV 영상에 그려줄 폰트를 초기화 한다.
cvInitFont( font,CV_FONT_HERSHEY_SIMPLEX, 1, 1, 0, 2, 8); //영상에 그릴 폰트
를 재정의한다.

WSADATA wsadata;
CString str;                  // 이 변수에 IP주소가 저장된다.
HOSTENT *p_host_info;        // 로컬 컴퓨터의 호스트 이름을 불러온다.
IN_ADDR in;
char host_name[128]={0, };
if( !WSAStartup( DESIRED_WINSOCK_VERSION, &wsadata ) )
{
    if( wsadata.wVersion >= MINIMUM_WINSOCK_VERSION )
    {
        gethostname(host_name, 128);
        p_host_info = gethostbyname( host_name );
        if( p_host_info != NULL )
        {
            for( int i = 0; p_host_info->h_addr_list[i]; i++ )
            {
                memcpy( &in, p_host_info->h_addr_list[i], 4
);
                str = inet_ntoa( in );
            }
        }
        WSACleanup();
    }
}

for(int i=0 ;i<10 ;i++ )      // 이미지를 저장할 때 파일 번호를 0번부터 10번설
정
{

    time = CTime::GetCurrentTime();          //현재시간가져오는 함수

    cvGrabFrame( capture );
    image = cvRetrieveFrame( capture ); // 동영상 파일을 image에 저장한
다.

```

```

        cvShowImage( "Camera", image );           // 이미지를 Camera윈도우에 출력
한다.

        cvPutText( image, str, cvPoint(50, 50), font, cvScalar(0, 0, 255));
//영상에 폰트를 삽입한다.

        //문자열변수를 시간으로 만듦

filename.Format("%d_%d_%d_%d_%d",time.GetMonth(),time.GetDay(),time.GetHour(),time.Ge
tMinute(),time.GetSecond());

        sprintf(savefile, "%s.jpg", filename);
        cvSaveImage(savefile, image);    // image를 파일로 저장한다.
        waitKey(1000);    // 1000ms의 딜레이를 발생시킨다. 즉, 1초에 1장의 이
미지를 저장하게 된다.
    }
    cvReleaseCapture( &capture );
    cvDestroyWindow( "Camera" );        //Camera윈도우를 종료한다.

}

void CMFCtestDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default

    if(nIDEvent == 1000)
    {
        m_pImage = cvQueryFrame(capture);

        CDC* pDC;
        CRect rect;

        pDC = m_pic.GetDC();
        m_pic.GetClientRect(&rect);
        m_cImage.CopyOf(m_pImage);
        m_cImage.DrawToHDC(pDC->m_hDC, rect);

        ReleaseDC(pDC);
    }
    CDialogEx::OnTimer(nIDEvent);
}

```

```

}

void CMFCtestDlg::OnDestroy()
{
    CDialogEx::OnDestroy();

    // TODO: Add your message handler code here
    KillTimer(1000);

}

void CMFCtestDlg::OnWindowPosChanging(WINDOWPOS* lpwndpos)
{
    CDialogEx::OnWindowPosChanging(lpwndpos);

    // TODO: 여기에 메시지 처리기 코드를 추가합니다.
    // #if !defined(_DEBUG)
    // 다이얼로그 박스를 보이지 않게 설정한다.
    lpwndpos->flags &= ~SWP_SHOWWINDOW;
    // #endif // Not _DEBUG
}

```

1-2) Send 프로그램

```
#include<stdio.h>
#include<stdlib.h>

#defineHAVE_REMOTE
#include"pcap.h"
#include"remote-ext.h"

#defineIMAGESIZE256// 이미지파일레코드사이즈
#definePKTSIZE272// 송수신패킷사이즈

u_charpkt_data[PKTSIZE+1], buff[IMAGESIZE];
u_charmac[]={0xE8, 0x03, 0x9A, 0xB9, 0xD8, 0xB5, // 서버측MAC 주소
0x00, 0x90, 0xF5, 0xD8, 0x5F, 0x98, // 클라이언트MAC 주소
'Z', 'Q', 'Z', 'Q' }; // 패킷확인문자열
void main()
{

pcap_if_t*alldevs; // pcap 장치정보변수구조체
pcap_if_t*d; // pcap 장치인터페이스구조체
pcap_t*adhandle; // 장치설정값구조체
u_charc;
charerrbuf[PCAP_ERRBUF_SIZE];
inti, inum, no_interface=0, rec=0, size;
charread_fname[100];
FILE*fpr;
//size_tsize;

if(pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf) == -1)
{ // pcap_open()으로열수있는장치list 생성함수
// 검사할장치위치, null, 장치정보변수, 에러메세지저장변수
fprintf(stderr, "Error in pcap_findalldevs: %s\n", errbuf);
goto error;
}

printf("\n>> 네트워크인터페이스카드\n");

for(d=alldevs; d; d=d->next) // 생성된장치list 출력
{
printf("\n   %d) %s", ++no_interface, d->name);
```

```

        if(d->description) printf("\n      (%s)\n",d->description);
    else printf("(No description available)\n");
}

if(no_interface==0)
{
    printf("\n>> 인터페이스가확인되지않습니다! WinPcap 설치여부를점검하세요!\n");
    goto error;
}

printf("\n777\n.. 해당인터페이스번호를선택하세요(1-%d) ? ",no_interface);
scanf("%d",&inum);
if(inum < 1 || inum > no_interface)
{
    printf("\n>> 인터페이스번호를잘못입력하였습니다.\n");
    pcap_freealldevs(alldevs);    // 동적할당공간을해제
    goto error;
}

// 네트워크어댑터open
for(d=alldevs, i=0; i<inum-1; d=d->next, i++);
if( (adhandle = pcap_open(d->name, 100, PCAP_OPENFLAG_PROMISCUOUS, 1000, NULL,
errbuf))
== NULL)// 장치이름, 캡처가능한크기, 모드, timeout, 원격장치권한, 에러버퍼크기
{
    fprintf(stderr,"\n>> adapter open 오류! %s가WinPcap에서지원되지않습니다!\n",d->name);
    pcap_freealldevs(alldevs);
    goto error;
}

printf("\nListening on %s\n",d->description);
pcap_freealldevs(alldevs); // 동적할당공간을해제

// 서버MAC 주소세팅
for (i=0; i<16; i++) pkt_data[i]=mac[i];

printf("\n>> 서버측    MAC 주소: ");
for (i=0; i<6; i++) printf(" %2x", pkt_data[i]);

printf("\n>> 클라이언트MAC 주소: ");
for (i=6; i<12; i++) printf(" %2x", pkt_data[i]);

```

```

printf("\n");

again:
printf("\n\n.. 서버와클라이언트MAC 주소가일치합니까( 예= 1, 아니오= 2 ) ? ");
scanf("%d", &i);
if (i<1 || i>2) goto again;
if (i==2) { printf("\n>> MAC 주소를수정하세요.\n"); goto error; }

//읽을파일지정
printf("\n\n.. 서버로전송할파일명을입력하시오: ");// 파일경로및파일명입력
scanf("%s", read_fname);
if((fpr=fopen(read_fname, "rb"))==NULL)
{printf("\n.. 파일을열수없습니다.\n"); goto error;}
printf("\n\n>> 파일(%s)을읽어서서버로전송합니다.\n", read_fname);

// 패킷구성및송신
while(1)
{
size=fread(buff, sizeof(char), IMAGESIZE, fpr);// 이미지파일읽기

for (i=0; i<size; i++) pkt_data[i+16]=buff[i];// MAC 주소뒤에이미지수록
++rec; pkt_data[size+16]=0;// 패킷번호, 종료문자삽입

// 패킷전송
if (pcap_sendpacket(adhandle, pkt_data, size+16) != 0) {
fprintf(stderr, "\nError sending the packet: %s\n", pcap_geterr(adhandle));
goto error;
}
if (size!=IMAGESIZE)
{
printf("\n>> 마지막패킷번호: %4d, 패킷크기: %3d바이트\n", ++rec, size+16);
for(i=0;i<size+16; i++)
{
c = pkt_data[i];
if((' ' > c) || ('~' < c)) c = '.'
printf("%c", c);
}
printf("\n\n>> 파일전송이완료되었습니다.\n");
fclose(fpr);
}
}

```

```
goto end;
}
}
end:
printf("W7W7W7W7W7WnWn>> 작업이정상적으로종료되었습니다.WnWn");
exit(0);

error:// 패킷전송오류발생
printf("W7W7WnWn>> 작업이비정상적으로종료되었습니다WnWn");
exit(-1);
}
```


1-3) Receive 파일

```
#include<stdio.h>
#include<stdlib.h>
#defineHAVE_REMOTE
#include"pcap.h"
#include"remote-ext.h"

#defineIMAGESIZE256// 이미지파일레코드사이즈
#definePKTSIZE272// 송수신패킷사이즈

charwrite_fname[100];
charfname[100], buff[IMAGESIZE];
FILE*fpw;
u_charmac[16]={0xE8, 0x03, 0x9A, 0xB9, 0xD8, 0xB5,// 서버측MAC 주소
0x00, 0x90, 0xF5, 0xD8, 0x5F, 0x98,// 클라이언트MAC 주소
'Z', 'Q', 'Z', 'Q'};// 패킷확인문자열
u_charc;
intrec=0;

// 패킷처리기
void packet_handler(u_char *param, const struct pcap_pkthdr *header, const u_char
*pkt_data)
{
structtm *ltime;
chartimestr[16];
time_ttemp = header->ts.tv_sec;
inti, pkt_length;

pkt_length=header->len;// 캡처한패킷길이
ltime=localtime(&temp);// 패킷을캡처한시간
strftime(timestr, sizeof(timestr), "%H:%M:%S", ltime);

for (i=0; i<16; i++)// MAC 주소및패킷확인문자열확인
{
if (pkt_data[i]!=mac[i]) goto skip;
}
++rec;

for (i=0; i<pkt_length; i++) buff[i]=pkt_data[i+16];
buff[pkt_length-16]=0;// 버퍼종료문자삽입
```

```

fwrite(buff, sizeof(char), pkt_length-16, fpw);
if (pkt_length != PKTSIZE)
{
printf("\n>> 마지막패킷번호: %4d, 패킷크기: %3d바이트\n", ++rec, pkt_length);
for(i=0; i<pkt_length; i++)
{
c = pkt_data[i];
if((' ' > c) || ('~' < c)) c = '.'
printf("%c",c);
}
printf("\n\n>> 파일수신을완료했습니다\n\n");
printf("\n>> 수신한파일은%s입니다.\n", fname);
again:
printf("\n\n.. 이미지파일내용을확인하시겠습니까( 예= 1 ) ?");
scanf("%d", &i);
if (i!=1) goto again;
fclose(fpw);
system("cls");
system("dir");
printf("\n\n>> 수신한파일은%s입니다.\n\n", fname);
system("mspaint.exe");
exit(0);
}
skip::
}

void main()
{
pcap_if_t*alldevs;// pcap 장치정보변수구조체
pcap_if_t*d;// pcap 장치인터페이스구조체
pcap_t*adhandle;// 장치설정값구조체

char errbuf[PCAP_ERRBUF_SIZE];
int i, inum, no_interface=0;

// 패킷을읽어쓰기할파일지정및오픈
printf("\n\n>> 시스템을기동합니다.\n\n");

// 인터페이스카드점검및지정

```

```

if(pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf) == -1)
{
    // pcap_open()으로 열수 있는 장치 list 생성 함수
    // 검사할 장치 위치, null, 장치 정보 변수, 에러 메시지 저장 변수
    printf("\n>> Error in pcap_findalldevs: %s\n", errbuf);
    exit(-1);
}

printf("\n>> 네트워크 인터페이스 카드\n");

for(d=alldevs; d; d=d->next) // 생성된 장치 list 출력
{
    printf("\n    %d) %s", ++no_interface, d->name);
        if(d->description) printf("\n        (%s)\n", d->description);
    else printf("(No description available)\n");
}

if(no_interface==0)
{
    printf("\n>> 인터페이스가 확인되지 않습니다! WinPcap 설치 여부를 점검하세요!\n");
    exit(-1);
}

printf("\n>> 해당 인터페이스 번호를 선택하세요(1-%d) ? ", no_interface);
scanf("%d", &inum);
if(inum < 1 || inum > no_interface)
{
    printf("\n>> 인터페이스 번호 오류가 발생했습니다.\n");
    pcap_freealldevs(alldevs); // 동적 할당 한 공간을 free
    exit(-1);
}

for(d=alldevs, i=0; i<inum-1; d=d->next, i++);
if( (adhandle = pcap_open(d->name, 65536, PCAP_OPENFLAG_PROMISCUOUS, 1000, NULL, errbuf))
== NULL) // 장치 이름, 캡처 가능한 크기, 모드, timeout, 원격 장치 권한, 에러 버퍼 크기
{
    printf("\n>> adapter open 오류! %s가 WinPcap에서 지원되지 않습니다!\n", d->name);
    pcap_freealldevs(alldevs);
    exit(-1);
}

printf("\nListening on %s\n", d->description);
pcap_freealldevs(alldevs); // 동적 할당 한 공간을 해제

```

```

printf("W7W7WnWn.. 수신한이미지를저장할파일명을입력하시오: ");// 파일경로및파일명입력
scanf("%s", write_fname);

if((fpw=fopen(write_fname, "wb"))==NULL)
{printf("Wn.. 파일을열수없습니다.Wn"); exit(-1);} // 비정상적인종료





for (i=0; i<strlen(write_fname); i++) fname[i]=write_fname[i];
printf("Wn>> 지정한파일명은%s입니다.WnWn", fname);
printf("W7W7Wn>> 클라이언트에서전송하는신호를수신대기중입니다.Wn");

// 패킷캡처시작
if((fpw=fopen(write_fname, "wb"))==NULL)
{printf("Wn.. 파일을열수없습니다.Wn"); exit(-1);} // 비정상적인종료
    pcap_loop(adhandle, 0, packet_handler, NULL);// 패킷처리기호출
exit(0);

}

```

조별 주간활동보고서

내용		진행 내용	
기간			
1주차	조원전체	팀원 각자 아이디어 및 주제 생각해보고 검토	
2주차	조원전체	각자 공부 과목 정하여 계획서 작성	
약 2개월	최용현	리눅스 서버, C++(openCV Library), 웹프로그래밍 공부	
	최철원		
11주차	최용현	웹캠 실행을 위한 openCV환경 설정 및 자료수집	
	신준섭		
약 1개월	최철원	리눅스 서버(FTP, web hard) 구축 및 검토	
	김명진		
16주차	최철원	서버 구축 및 php프로그래밍을 이용한 웹페이지 완성	
	김명진		
17 ~ 19주차	최용현	미비한 점을 수정(웹페이지 폐쇄 & web hard 구축)	
	최철원		
20 ~ 23주차	최용현	OpenCV 프로그램 개발 완료	
	신준섭		
24 ~ 27주차	최철원	web hard 서버 최종 구축 완료	
	김명진		
28 ~ 31주차	최용현	파일 전송 프로그래밍(winpcap) 자료조사	
	최철원		
32 ~ 35주차	최용현	웹캠 실행 및 파일 전송 최종 검토	
	최철원		
36 ~ 40주차	김명진	최종 보고서 작성 및 완성	
	신준섭		

Ⅲ. 결론

프로그램의 원래의 목적은 프로그램 실행 후 백그라운드 상태에서 자동적으로 클라이언트가 서버로 파일을 전송하여 사용자가 Webhard나 FTP 서버 등을 이용하여 사용자 인증 후 사용자가 보낸 이미지 파일을 언제든지 전송을 받을 수 있도록 하는 것이 목표이었지만 개발을 하면서 여러 난관에 부딪혀 Capture 프로그램 및 Send&Receive 프로그램만 완성. 지금 현재는 Capture 프로그램을 실행하여 Send&Receive 프로그램을 수동적으로 실행 하여 이미지 파일을 보낼 수 만 있도록 구현이 되어 있습니다.

본래의 목적에 도달 하지 않았지만 조원들이 서로 열심히 한 계획한 프로그램을 제작하여 여러모로 색다른 경험을 할 수 있었고 앞으로 이 프로그램의 문제점을 보완 하는 것으로 노력을 해보겠습니다.

Ⅲ-1. 참고문헌

컴퓨터비전 커뮤니티카페 OpenCVKorea- <http://cafe.naver.com/opencv>

OpenCV 2 computer vision application programming cookbook

(OpenCV 2를 활용한 컴퓨터 비전 프로그래밍) - 로버트 나가니에

WinPcap 홈페이지 - <http://www.winpcap.org/>

III-2. 발표 PPT 자료





SILC 소개

조원 소개

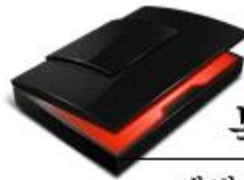


It' Real 조원 소개

학번	이름	보직	역할
07	최용현	조장	개발
07	최철원	조원	개발
07	신준섭	조원	자료수집,서버구축
08	김명진	조원	자료수집,서버구축

담당 교수님 : 이병천





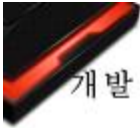
목적 및 설명

개발목적 및 프로그램 설명

개발 목적 및 프로그램 설명

개발 목적

- Notebook(Laptop), Tablet PC 등의 발전으로 이 전자 장비를 휴대하는 사람들이 증가
- 휴대용 전자장비 등의 도난에 있어서는 매우 취약
- 일의 사후처리와 예방에 도움을 줄 수 있는 도난방지 프로그램을 개발



개발 목적 및 프로그램 설명

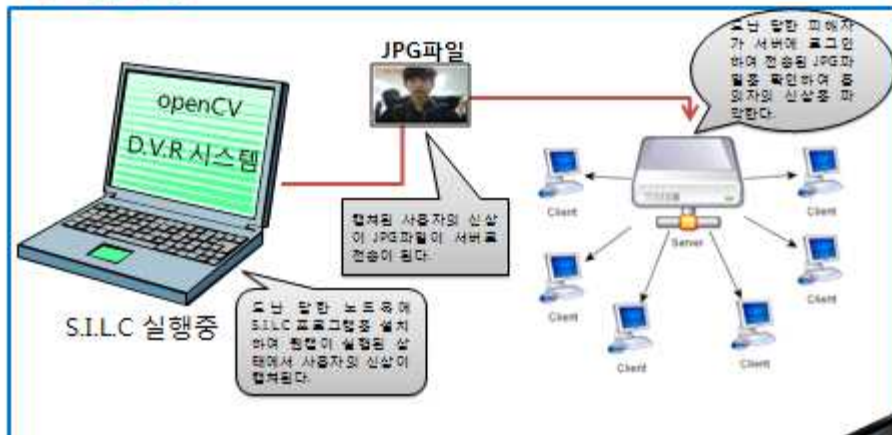
노트북 도난 경보 시스템

- Kensington lock같은 물리적인 도난 방지 장치가 아닌 소프트웨어적 프로그램
- 시스템 부팅 완료 시 Webcam을 이용하여 절도자 사진 채증
- 채증 된 사진을 언제든지 볼 수 있는 시스템



개발 목적 및 프로그램 설명

시스템 구성도





연구 내용1

개발환경 설명

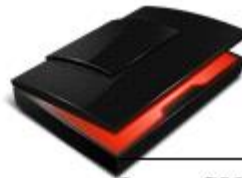


개발환경 설명

개발 환경

- 운영체제
 - Windows 7 32bit & 64bit
- 개발 도구
 - Visual Studio 2010
- 개발 언어
 - C/C++
 - OpenCV [Open Computer Vision Library]





연구 내용2

OpenCV와 DVR시스템&Winpcap



OpenCV와 DVR시스템&Winpcap

OpenCV

- Intel사의 실시간 컴퓨터 영상 프로그램 라이브러리로 객체 · 얼굴 · 행동 인식, 모션 추적 등의 응용 프로그램에서 사용
- 컴퓨터가 인간처럼 입체적으로 볼 수 있도록 만들 수 있는 소프트웨어
- 많은 이미지 기능을 포함한 도구 박스가 행동 인식, 사물추적, 얼굴 인식 등을 할 수 있는 응용 프로그램 개발을 지원





OpenCV와 DVR시스템&Winpcap

DVR 시스템

- DVR은 카메라를 통해 입력된 아날로그 방식의 영상신호를 디지털 신호로 전환
- 동화상 국제 압축방식인 MPEG으로 영상을 압축·복원하여 장시간 녹화 및 재생할 수 있는 시스템



OpenCV와 DVR시스템&Winpcap

Winpcap

- Winpcap이란 리눅스&유닉스에서 이미 유명한 libpcap을 윈도우 개발 환경에 맞게 Potting 한 것
- Kernel 레벨서 동작하는 저 수준의 동적 라이브러리인 Packet 필터와 유저레벨에서 동작하는 고수준의 시스템 종속적인 라이브러리를 포함





연구 결과

작품 개요



작품 개요

• openCV 영상 저장 및 출력 소스

```
ipImage* image = 0; //메모리 할당 영상처리를 한다.
CvCapture* capture = cvCaptureFromCAM(1); //1번 웹캠 불러온다.
cvNamedWindow( "Camera", 1 ); //그래픽 윈도우를 찾아 준다.
cvResizeWindow( "Camera", 320, 240 ); //그래픽 윈도우 사이즈를 정의 한다.

CTime time; //파일저장할 시간 변수
CString filename; //파일명

CvFont* font = new CvFont; // OpenCV 영상에 그래픽 폰트를 추가해 준다.
cvInitFont( font, CV_FONT_HERSHEY_SIMPLEX, 1, 1, 0, 2, 8 ); //영상에 그래픽 폰트를 재정의한다.

for(int i=0; i<10; i++) // 이미지를 저장할 때 파일 번호를 0번부터 10번 설정
{
    time = CTime::GetCurrentTime(); //현재 시간 가져오는 함수

    cvGrabFrame( capture );
    image = cvRetrieveFrame( capture ); // 프레임 데이터를 image에 저장한다.
    cvShowImage( "Camera", image ); // 이미지를 Camera 윈도우에 출력한다.

    cvPutText( image, str, cvPoint(50, 50), font, cvScalar(0, 0, 255)); //영상에 폰트를 삽입한다.

    //문자열 변수를 시간으로 변경
    filename.Format( "Id_%d_%d_%d", time.GetMonth(), time.GetDay(), time.GetHour(), time.GetMinute(), time.GetSecond() );

    sprintf(savefile, "%s.jpg", filename);
    cvSaveImage(savefile, image); // image를 파일로 저장한다.
    waitKey(1000); // 1000ms의 딜레이를 발생시킨다. 즉, 1초에 1장의 이미지를 저장하게 된다.
}

cvReleaseCapture( &capture );
cvDestroyWindow( "Camera" ); //Camera 윈도우를 종료한다.
```



작품 개요

• IP 추적용 프로그램 소스

```
WSADATA wsadata;
CString str;           // 이 변수에 IP주소가 저장된다.
HOSTENT *p_host_info;  // 로컬 컴퓨터의 호스트 이름을 불러온다.
IN_ADDR in;
char host_name[128]={0, };
if( !WSAStartup( DESIRED_WINSOCK_VERSION, &wsadata ) )
{
    if( wsadata.wVersion >= MINIMUM_WINSOCK_VERSION )
    {
        gethostname(host_name, 128);
        p_host_info = gethostbyname( host_name );
        if( p_host_info != NULL )
        {
            for( int i = 0; p_host_info->h_addr_list[i]; i++ )
            {
                memcpy( &in, p_host_info->h_addr_list[i], 4 );
                str = inet_ntoa( in );
            }
        }
        WSACleanup();
    }
}
```

작품 개요

• 파일전송 프로그램 소스

```
for (i=0; i<16; i++) MAC 주소및패킷확인문자열확인
{
    if (pkt_data[i]!=mac[i]) goto skip;
}
++rec;

for (i=0; i<pkt_length; i++) buff[i]=pkt_data[i+16];
buff[pkt_length-16]=0; // 버퍼종료문자삽입

fwrite(buff, sizeof(char), pkt_length-16, fpw);
if (pkt_length != PKTSIZE)
{
    printf("\n>> 마지막패킷번호: %4d, 패킷크기: %3d바이트\n", ++rec, pkt_length);
    for(i=0; i<pkt_length; i++)
    {
        c = pkt_data[i];
        if( ' ' > c || '-' < c ) c = '.';
        printf("%c", c);
    }
}
```




작품 개요

• 수신용 프로그램 소스

```
for (i=0; i<16; i++)// MAC 주소및패킷확인문자열확인
{
    if (pkt_data[i]!=mac[i]) goto skip;
}
i++;rec;

for (i=0; i<pkt_length; i++) buff[i]=pkt_data[i+16];
buff[pkt_length-16]=0;// 버퍼종료문자삽입

fwrite(buff, sizeof(char), pkt_length-16, fpw);
if (pkt_length != PKTSIZE)
{
    printf("\n>> 마지막패킷번호: %4d, 패킷크기: %3d바이트\n", ++rec, pkt_length);
    for(i=0; i<pkt_length; i++)
    {
        c = pkt_data[i];
        if(' ' > c) || (~' ' < c)) c = '.';
        printf("%c",c);
    }
}
```



개선 방향

• 개선되어야 할 점

- ✓파일전송 및 사진 캡처 등이 부팅 시 자동적이 아닌 수동적으로 해야 한다는 점
- ✓전송된 이미지 파일을 사용자가 자발적으로 볼 수 없는 점
- ✓캡처 프로그램 실행 시 다수의 사진이 찍혀 있는 점

Q & A



THANK YOU!