OS 모니터링 시스템 구축

팀 명 : 트론

지도 교수 : 이병천 교수님

팀 장 : 이재희

팀 원 : 이영상

나경민

이동준

주영문

2019. 10.

중부대학교 정보보호학과

목 차

5. 별첨

1. 서 론

1.1 연구 배경 및 주제 선정

최근 정보보안 취약점을 노리는 사이버 공격이 급증함에 따라 보안 취약점 진단에 대한 중요성이 더 높아지고 있다. 보안 관리자가 수많은 취약점을 일일이 찾아 패치하는 데에는 한계가 있는 만큼 IT시스템, 애플리케이션, 웹 등에 있는 취약점을 자동 진단하는 취약점 분석 솔루션 수요는 더욱 늘어날 전망이다. 운영체제의 사용을 위해 적절한 보안을 위해 보안 관리자가 보안을 해결하기는 곤란하므로 자동 분석 수요도 늘어날 전망이다.
두 번째로, 올해 세계 정보보안시장 규모는 전년도를 대비하여 12.4% 성장한 1,140억달러로 추산되었다. 2019년 세계 시장은 8.7% 증가한 1,240억 달러에 달할 전망이다. 가트너는 세계 시장을 견인하는 주요인으로 감지 및 대응 역량 구축에 대한 관심 증가, GDPR과 같은 개인정보보호 규정, 디지털 비즈니스에 대한 위험 요소를 해결해야 하는 것이 필요하다고 주장했다.
세 번째로, 주요정보통신기반시설 관리기관은 [정보통신기반 보호법] 제9조에 따라, 매년 취약점 분석 및 평가를 실시하여야 한다. 취약점 분석 및 평가는 453개의 관리적/물리적/기술적 점검항목에 대한 취약 여부를 점검하고 해킹, 침투 등을 수행하는 종합적인 위험 진단에 해당한다.
그에 위험 진딘을 목표로 정보시스템의 안정적 운영을 위협하는 요소들을 분석하고, 문제가 될 수 있는 부분을 도출하였고, 1차 구현을 마치고도 계속된 조사를 통해서 한국인터넷진흥원(KISA)에서 발간한 "주요정보통신기반시설 기술적 취약점 분석 및 평가 방법 상세가이드"를 찾았고, 내용 중 윈도우 보안 가이드의 취약점을 추가 구현하기로 하였고, 참고해서 일부분을 추가 반영하였다.

2. 관련 연구

2.1 Windows 10

Windows NT 계열의 운영체제의 하나로, 개인용 컴퓨터를 위한 운영체제로 개발되었다. 2015년 7월 29일 일반 사용자에게 공개되었으며, Windows Vista 이후 Windows 7, 8로 버전이 넘버링 되어왔는데, 9를 뛰어넘고 Windows 10으로 이름이 붙여졌다. 이는 Microsoft가 차세대 Windows, 즉 모든 장치에서 포괄적으로 동작하는 다양한 플랫폼을 나타내기 위한 의도에 의한 것이다. 초기에는 Windows Kernel이 NT 6.4였으나 현재 NT 10.0으로 출시되었다. 이전 Windows와 가장 다른 점은 스마트폰이나 IoT 단말기에 이 운영체제를 사용할 수 있게 만들었다는 점이다.

2.2 C#

Microsoft에서 개발된 객체 지향 프로그래밍 언어로, Java, C++과 비슷한 면을 많이 가지고 있다. C#은 .NET Framework를 이용하여 프로그래밍하는 대표적인 언어이다. 윈도우 프로그래밍, 웹 프로그래밍, 모바일 프로그래밍 등 모든 영역에서 사용되는 범용 프로그래밍 언어이다.

2.3 JSON

본래는 Javascript 언어로부터 파생되어 자바스크립트의 구문 형식을 따르지만, 언어 독립형 데이터 포맷이다. 즉, 프로그래밍 언어나 플랫폼에 독립적이므로, 구문 분석 및 JSON 데이터 생성을 위한 코드는 C, C++, C#, Java, Javascript, Python 등 수많은 프로그래밍 언어에서 쉽게 이용할 수 있다.

2.4 Nginx/PHP 7.2

Nginx는 Web server 오픈소스 소프트웨어로 엔진박스가 제공하는 각 기능은 모듈화되어 있어 효율적인 운영을 할 수 있고, Apache Web Server 소프트웨어보다 메모리 사용이 효율적이며 Client 처리속도가 더 빠르다. 러시아의 프로그램 개발자인 이고르 시쇼브가 2004년에 개발하여 발표하였다.

PHP 7.2는 공개된 무료 소스로 Hypertext 생성 언어에 포함되어 동작하는 스크립팅 언어로, 별도의 실행 파일을 만들 필요 없이 HTML 문서 안에 직접 포함시켜 사용하며, C언어, JAVA 등에서 많은 문장 형식을 준용하고 있어 동적인 웹 문서를 빠르고 쉽게 작성할 수 있다. ASP와 같이 스크립트에 따라 내용이 다양해서 동적 HTML 처리속도가 빠르며, PHP 스크립트가 포함된 HTML 페이지에는 확장자명이 .php, .php3, .phtml이 붙여진 파일 이름이 부여된다.

2.5 MariaDB

오픈 소스의 관계형 데이터베이스 관리 시스템으로 MySQL과 동일한 소스 코드를 기반으로 하며, GPL v2 라이센스를 따른다.

새로운 저장 엔진인 마리아뿐만 아니라, InnoDB를 교체할 수 있는 XtraDB 저장 엔진을 포함하고 있다.

2.6 Semantic UI

Semantic은 인간 친화적인 HTML을 사용하여 아름답고 반응이 빠른 레이아웃을 만드는데, 도움이 되는 개발 프레임 워크이다.

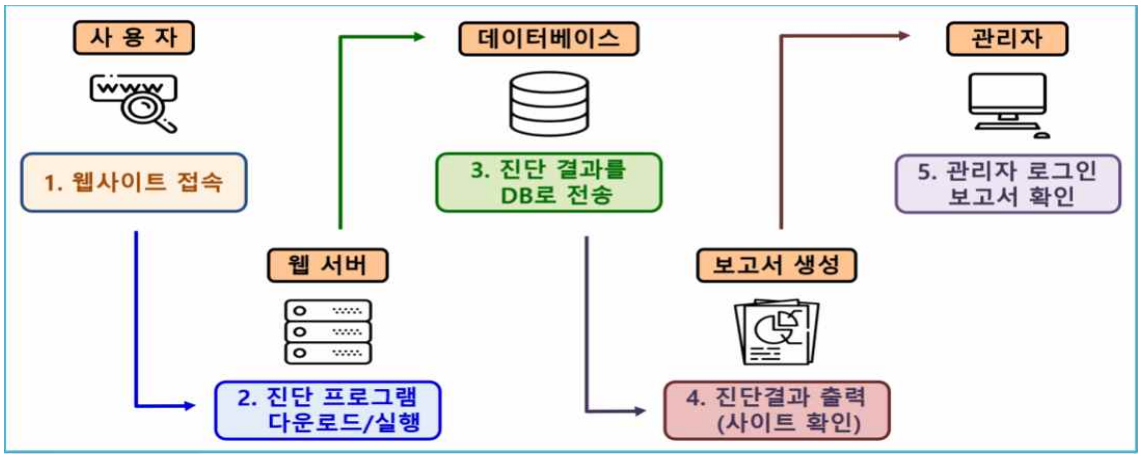
Semantic UI는 단어와 클래스를 교환 가능한 개념으로 취급한다.

클래스는 개념을 직관적으로 연결하기 위한 자연어의 구문을 사용한다.

3. 본 론

3.1 시스템 구성

안전한 운영체제의 사용을 위해 자동진단해서 필요한 조치를 취할 수 있도록 하기 위해, 해당 사용자가 수집 프로그램에 대한 자료를 수집하고 웹사이트에 접속을 시도한다. 웹 서버에서 수집된 자료에 의해 에이전트 프로그램을 실행시키고, 실행 결과를 DB에 저장, 이에 대한 DB를 종합해 수집한 자료를 정보화시켜 이를 보고서에 출력하고 해당 관리자에게 전달한다. 이 관리자는 그 기능을 제공함으로써 작성된 내용을 관리하는 임무를 수행한다. 분석 과정에서 필요할 수 밖에 없는 레지스트리(설정, 정보 등이 저장되므로..)의 접근이나 부팅 영역 정보 등을 가져오기 위해 실행 과정에서 관리자 권한을 요구한다.



[그림 3-1] 시스템 구상도

3.2 프로그램 구성

3.2.1 PC 취약점 분석(KISA 발표)

PC 취약점을 분석하면서 KISA 보고서 중 계정관리를 제외시켰는데, 이는 계정 정책 정책에 관련해서 마이크로소프트가 PC 운영체제인 윈도우 10을 출시하면서 패스워드를 주기적으로 변경해야 한다는 조항을 삭제하였기 때문이다. 이유는 비밀번호 변경 조항이 이용자들에게 있어 불편할 뿐 아니라, 오히려 비밀번호를 계속 변경하는 것이 새로 바꾼 비밀번호를 잊어먹는 등의 사고를 유발할 수 있으므로 보안에 미약하다는 주장이 제기되었기에 삭제한 것이다.

정보시스템 진단을 기준으로 공신력 있는 기관에서 발행한 취약점 진단 관련 가이드에서 언급하는 내용을 통해 고객사의 시스템을 진단하는 것이 진단의 근거를 마련하는데 매우 중요한 행위다.

이러한 취약점을 진단하기 위하여 많은 곳에서는 한국인터넷진흥원에서 발간한 주요정보통신기반시설 기술적 취약점 분석 평가 상세 가이드"는 참고하여 매년 "취약점 분석 및 평가"를 실시하였고, 그중에서도 PC 보안 가이드의 취약점을 분석하였다. 중요 단말기인 PC에 대한 보안성을 평가하여 현재 취약점 및 향후 발생 가능한 위협요소를 도출하고 보호 대책을 수립하였다.

분류	점검항목	중요도	항목코드
2. 서비스관리	공유 폴더 제거	상	PC-03
	항목의 불필요한 서비스 제거	상	PC-04
	Windows Messenger(MSN, .NET 메신저 등)와 같은 상용 메신저의 사용 금지	상	PC-05
	파일 시스템이 NTFS 포맷으로 되어 있는가?	중	PC-16
	대상 시스템이 Windows 서버를 제외한 다른 OS 멀티 부팅이 가능하지 않도록 설정하여 사용하는가?	중	PC-17
	브라우저 종료 시 임시 인터넷 파일 폴더의 내용을 삭제하도록 설정하여 사용하는가?	하	PC-18
3. 패치관리	HOT FIX 등 최신 보안패치 적용	상	PC-06
	최신 서비스팩 적용	상	PC-07
	MS-Office, 한글, 어도비 아크로뱃 등의 응용 프로그램에 대한 최신 보안패치 및 벤더 권고사항 적용	상	PC-08
4. 보안관리	바이러스 백신 프로그램 설치 및 주기적 업데이트	상	PC-09
	바이러스 백신 프로그램에서 제공하는 실시간 감시 기능 활성화	상	PC-10
	OS에서 제공하는 침입차단 기능 활성화	상	PC-11
	화면보호기 대기 시간 설정 및 재시작 시 암호 보호 설정	상	PC-12
	CD, DVD, USB 메모리 등과 같은 미디어의 자동실행 방지 등 이동식 미디어에 대한 보안대책 수립	상	PC-13
	PC 내부의 미사용(3개월) ActiveX 제거	상	PC-14
	시스템 부팅 시 Windows Messenger가 자동으로 시작되지 않도록 설정되어 있는가?	중	PC-19
	원격 지원을 금지하도록 정책이 설정되어 있는가?	중	PC-20

[그림 3-2] PC 취약점 분석 및 평가 항목

3.2.2 PC 정보 획득

PC별 취약점 정보를 획득하기 위해서 구분자로 운영체제명, PC이름 등을 확인한다.
아래 스크린샷의 방법으로 C#에 내장된 Environment 속성을 이용하여 PC이름을 확인하고, 윈도우에서 다양한 정보를 외부로 보내기 위해 사용하는 WMI 객체를 이용하여 운영체제명을 확인하고, 다시 Environment 속성을 이용하여 Architecture(32비트 / 64비트 등)까지 확인하였다.
여기까지는 PC별 환경이 동일할 수 있으므로, 정확한 구분자를 목적으로 IP(내부, 외부) 등도 확인한다.
메모리(램) 크기도 계산하는 데, 응용 프로그램에서는 컴퓨터의 장치 요소들 중 할당된 메모리(Hardware reserved)를 제외하고 계산되므로 정보가 맞지 않아서 어려웠다.
이에 외부에서 제공(Microsoft)하는 SafeNativeMethods 라이브러리를 사용해서 메모리

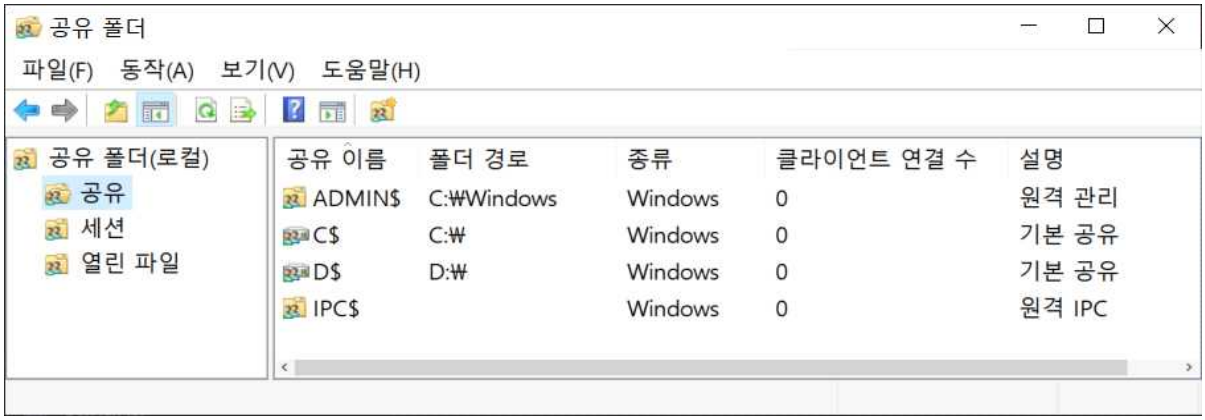
크기를 제대로 출력할 수 있었다.
(SafeNativeMethods 라이브러리: 보안에 영향을 끼칠 수 있는 것으로 보이지만, 보안 상에는 문제가 없는 API 함수들이 들어있음 = 마이크로소프트 제공 안전한 라이브러리)

```
internal string PCName()
{
    return Environment.MachineName;
}
internal string OSName()
{
    string result = "Not found!";
    ConnectionOptions options = new ConnectionOptions();
    options.Impersonation = ImpersonationLevel.Impersonate;
    ManagementScope scope = new ManagementScope("/root/cimv2", options);
    scope.Connect();
    ObjectQuery query = new ObjectQuery("Select Caption From Win32_OperatingSystem");
    ManagementObjectSearcher search = new ManagementObjectSearcher(scope, query);
    ManagementObjectCollection queryCollection = search.Get();
    foreach (ManagementObject o in queryCollection)
    {
        result = o["Caption"].ToString();
        if (Environment.Is64BitOperatingSystem)
        {
            result = result + " 64비트 ";
        }
        else
        {

            result = result + " 32비트 ";
        }
        string ver = GetOSVer();
        int len = ver.Length;
        result = result + "(" + ver.Substring(0, len - 6) + ", 빌드 " + ver.Substring(len - 5) + ")";
    }
    return result;
}
```

[그림 3-3] 운영체제 정보

3.2.3 기본 공유 디렉터리

Windows 설치 시, 기본적으로 공유 폴더들이 다음 스크린샷처럼 설정된다.
설정은 변경하지 않고 계속 사용 시, 비인가자가 모든 시스템 자원에 접근할 수 있는 위험한 상황이 발생할 수 있다.
아래 [그림 3-4]에서 공유 중지가 가능하지만, 기본으로는 윈도우 시작 시 기본 공유 목록으로 갱신하게 되어있어서 이 부분도 확인해야 한다. (레지스트리 이용)
공유 폴더가 위험하지만 사용해야 하는 경우, 방화벽에서 135, 139 (TCP/UDP) 포트를 통신할 수 없도록 하거나, 공유 폴더에 접근 권한 및 암호가 설정되어 있어야 한다.

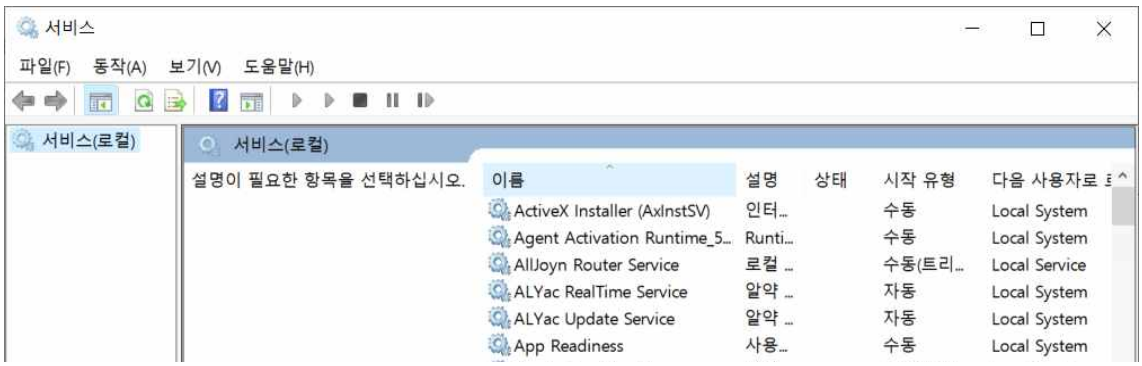


[그림 3-4] 기본 공유 정보

3.2.4 불필요한 서비스

사용하는 서비스를 크게 줄일수록 악의적인 사용자가 접근할 가능성이 낮아진다. 그러므로, 사용하지 않거나 기본으로 사용 중이지만 영향이 크게 없는 서비스는 중지하는 것이 보안상에는 좋은 점이 될 수 있다.

하지만, 불필요 서비스 목록을 모든 컴퓨터에서 모두 확인하는 것은 PC마다 서비스 설치 상황이 다르므로 모든 서비스를 조사해서 다룰 수는 없으므로, 우리는 윈도우 기본 설치 서비스만을 확인한다.



[그림 3-5] 서비스 관리 정보

하지만, 윈도우 보유 서비스에도 Telnet, TCP/IP 서비스 등도 취약점으로 볼 수 있어서 윈도우 보유 서비스 중 보안 상 좋지 않은 서비스들의 구동 여부를 점검한다.

3.2.5 메신저 소프트웨어

일반 사용자 PC에서 메신저를 사용할 경우, 악의적 사용자나 사용 중인 다른 프로그램 등으로 주요 정보가 유출될 수 있을 뿐만 아니라 악성코드가 유입될 수 있다.

상용 메신저 차단을 통하여 메신저를 이용한 개인정보 및 내부 주요 정보 유출을 막을 수 있다. 대부분의 프로그램 설치 시, 인스톨러가 그 프로그램에서 쓰이는 환경에 맞춰서 자동으로 레지스트리 정보를 생성된다.

```
if (check == 0)
{ // Kakaotalk Check
    test = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Uninstall₩KakaoTalk", "UninstallString", ""))
    if (File.Exists(test)) return true;
}
else if (check == 1)
{ // Line Check
    test = Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER₩Software₩Microsoft₩Windows₩CurrentVersion₩Uninstall₩LINE", "UninstallString", ""));
    if (test != null)
    {
        test = Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER₩Software₩NHN Corpration₩LINE", "RunOnce", ""));
        if (test != null)
        { // Uninstaller Not Remove
            test = Environment.GetEnvironmentVariable(test); // Get Registry Save File
            if (File.Exists(test)) return true; // If Exist
        }
        else
        { // Use Order Registry
            test = Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER₩Software₩Naver₩LINE", "RunOnce", ""));
            if (test != null)
            { // Uninstaller Not Remove
                test = Environment.GetEnvironmentVariable(test); // Get Registry Save File
                if (File.Exists(test)) return true; // If Exist
            }
        }
    }
}
else if (check == 2)
{ // Skype Check
    test = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE₩Software₩Microsoft₩Windows₩CurrentVersion₩Uninstall₩Skype_is1", "UninstallString", ""))
    if (File.Exists(test)) return true;
}

else if (check == 3)
{ // NateOn - Not Use Registry, But Install Path Not Change > Default Folder Check
    test = Environment.GetEnvironmentVariable(@"ProgramFiles(x86)") + @"₩SK Communications₩NATEON₩BIN₩NateOnMain.exe";
    if (File.Exists(test)) return true;
    else
    { // Default Use x86(32bit), Same Up Path, But Bottom Check

        test = Environment.GetEnvironmentVariable(@"ProgramFiles") + @"₩SK Communications₩NATEON₩BIN₩NateOnMain.exe";
        if (File.Exists(test)) { return true; }
    }
}
```

[그림 3-6] 메신저 소프트웨어 레지스트리 정보

3.2.6 브라우저 종료 시 캐시 삭제

브라우저 사용 시 생성되는 캐시 파일을 브라우저 삭제 시 종료하게 설정함으로써 안전하게 사용할 수 있다. 임시 인터넷 파일 폴더를 통해 이메일 주소, 웹 사이트 접근 기록 등의 제3자의 유출을 야기할 수 있으므로 취약하다는 것을 알 수 있다.

```
RegistryKey reg = Registry.CurrentUser.OpenSubKey(@"Software₩Microsoft₩Windows₩CurrentVersion₩Internet Settings₩Cache", true);
if (reg != null)
{
    Object val = reg.GetValue("Persistent");
    if (null != val)
    {
        if (val.ToString() == "1") val = "Y";
        else if (val.ToString() == "0") val = "N";
        return Convert.ToString(val); // Setting value
    }
```

[그림 3-7] IE 종료 시 캐시 삭제

2,015년 7월, 윈도우10과 같이 출시된 브라우저인 Edge도 동일한 역할을 수행하도록 설정할 수 있다.

```
    RegistryKey reg = Registry.CurrentUser.OpenSubKey(@"Software₩Classes₩Local Settings₩Software₩Microsoft₩Windows₩CurrentVersion₩
                                AppContainer₩Storage₩microsoft.microsoftedge_8wekyb3d8bbwe₩MicrosoftEdge₩Privacy", true);
    if (reg != null)
    {
        Object val = reg.GetValue("ClearBrowsingHistoryOnExit");
        if (null != val)
        {
            if (val.ToString() == "1") val = "Y";
            else if (val.ToString() == "0") val = "N";
            return Convert.ToString(val); // Setting value
```

[그림 3-8] Microsoft Edge 종료 시 캐시 삭제

3.2.7 보안관리

원격 터미널 서비스 기능이 활성화되면 시스템 제어 권한이 악용될 수 있어서 구동 여부를 확인한다.

참조하는 레지스트리 fDenyTSConnections 키의 값을 확인한다.

{ 0: 원격 지원 상태가 비활성화되어 안전하다는 뜻이다. }

```
RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SYSTEM₩CurrentControlSet₩Control₩Terminal Server", true);
if (reg != null)
{
    Object val = reg.GetValue("fDenyTSConnections");
    if (val != null)
    {
        if (Convert.ToString(val) == "0") return "Enable";
        else if (Convert.ToString(val) == "1") return "Disable";
        else return Convert.ToString(val);
    }
```

[그림 3-9] 원격 지원 상태

USB나 CD 등의 저장소 장치들도 레지스트리 값으로 구체적인 상태 확인이 가능한다. 장치별로 장치의 ID를 가져오고, 그 ID를 이용해서 저장소 장치들의 연결 허용 여부를 가져올 수가 있다. 그 주소에 아래와 같이 설정할 경우, 특정 기능을 수행한다. (모든 장치 거부, 특정 장치 쓰기/실행/읽기 모드 연결 허용 여부 검색)

```
string[] check = new string[] { "Deny_Read", "Deny_Write", "Deny_Execute", "Deny_All" };
string[,] DeviceList = {
    { "CD/DVD", "{53f56308-b6bf-11d0-94f2-00a0c91efb8b}" },
    { "Floppy", "{53f56311-b6bf-11d0-94f2-00a0c91efb8b}" },
    { "USB", "{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}" },
    { "Tape", "{53f5630b-b6bf-11d0-94f2-00a0c91efb8b}" },
    { "WPD", "{6AC27878-A6FA-4155-BA85-F98F491D4F33}" },
    { "WPD", "{F33FDC04-D1AC-4E8E-9A30-19BBD4B108AE}" }
};
List<string> list = new List<string>();
for (int i = 0; i <= DeviceList.GetUpperBound(0); i++)
{
    string name = DeviceList[i, 0];
    string address = DeviceList[i, 1];
    string value, output = null;
    value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE₩SOFTWARE₩Policies₩Microsoft₩Windows₩RemovableStorageDevices", check[3], ""));
    if (value == "1") return "Any Device Block";
    value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE₩SOFTWARE₩Policies₩Microsoft₩Windows₩RemovableStorageDevices₩" + address, check[0], "")
    if (value != "") output += "R";
    value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE₩SOFTWARE₩Policies₩Microsoft₩Windows₩RemovableStorageDevices₩" + address, check[1], "")
    if (value != "") output += "W";
    value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE₩SOFTWARE₩Policies₩Microsoft₩Windows₩RemovableStorageDevices₩" + address, check[2], "")
    if (value != "") output += "E";
    if (output != null) list.Add(DeviceList[i, 0] + "(" + output + ")");
}
```

[그림 3-10] 저장소 활성 상태

마찬가지로 위 장치들의 자동 실행이 허용된 상태이거나 금지된 상태 등을 레지스트리 정보로 확인할 수 있다.

```
RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Policies₩Explorer", true);
if (reg != null)
{
    string val = Convert.ToString(reg.GetValue("NoDriveTypeAutoRun"));
```

[그림 3-11] 외부 저장장치 정보

화면 보호기로 취약할 수 있는 부분에 대해서, 화면 보호기 설정을 하지 않은 경우, 사용자가 자리를 비운 사이에 임의의 사용자가 해당 시스템에 접근하여 중요 정보를 유출하거나, 악의적인 행위를 통해 시스템 운영에 나쁜 영향을 미칠 수 있다.

사용자가 일정 시간 동안 작업을 수행하지 않을 경우, 자동으로 로그오프 되거나 워크스

테이션이 잠기도록 설정하여 휴식시간 내 비인가자의 시스템 접근을 차단하여야 한다.

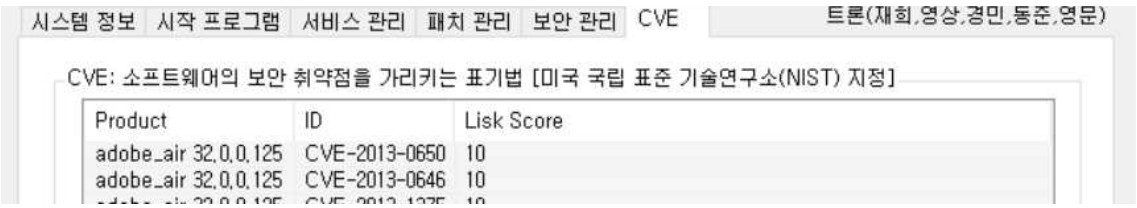
```
RegistryKey reg = Registry.CurrentUser.OpenSubKey(@"Control Panel\Desktop", true);
if (reg != null)
{
    Object val = reg.GetValue("ScreenSaveActive");
    screensaver = Convert.ToString(val);
    Object ssval1 = reg.GetValue("ScreenSaveTimeOut");
    Object ssval2 = reg.GetValue("ScreenSaverIsSecure");
```

[그림 3-12] 화면보호기 설정]

화면 보호기를 사용하는 것이 조금 더 안전하고, 사용 시 화면 보호기 해제 후 다시 시작 시 로그인 화면(ScreenSaverIsSecure)의 값이 1(사용 중)이 아니라면 비교적 취약, 화면 보호기가 작동되는 대기시간인 ScreenSaveTimeOut의 값이 300초(5분)보다 작으면 취약(KISA 가이드라인)하다는 것을 확인할 수 있다.

3.2.8 CVE 취약점

CVE란 다양한 분야의 보안 취약점을 발견한 년도와 발견한 순서로 구분하는 방법으로 미국 국립 표준 기술연구소(NIST)에서 지정하는 방법이다.
다른 조에서도 CVE 항목이 있었지만, 그 조는 웹 서비스이므로, 웹에 위반되어 있는 CVE 항목을 조사하고, 우리 프로그램은 설치되어있는 프로그램에서 CVE 위배로 지정되어 있는 소프트웨어들을 수집하는 것이다.
CVE는 규모도 엄청 크고, OS 상황들을 많이 준비하고, 각 소프트웨어들의 업데이트 영역도 확인해야 하므로, 시간이 크게 필요한 관계로 우리는 Vulmon에서 만든 PowerShell 스크립트를 발견했고 가공해서 연동할 수 있도록 크게 수정하였다.



[그림 3-13] CVE 보안 취약성

CVE 취약점을 진단하도록 프로그래밍을 하기에는 문제되는 버전의 프로그램들을 모두 설치해서 분석해봐야 하므로 많은 시간이 필요하고, 기술이 부족한 부분이 많아 Vulmon에서 웹으로 제공하는 CVE 위반 소프트웨어를 진단하는 스크립트를 발견하여, 이를 프로그램에 연동을 하도록 추가하였다.
여기에서 최근 취약하다고 지정된 소프트웨어인 adobe_air 32.0.0.125(구 버전, 패치 완료)를 설치한 경우, 프로그램의 CVE 탭에서 문제 확인이 가능하다.

3.2.9 크로스 스레드

크로드 스레드는 2개 이상 쓰레드가 한 개 항목을 동시에 다루어서 위험하다는 경고이다. 개발 중 디버깅만 되지 않는 문제를 꺾었다.
컴파일 및 실행은 아무 문제 없는 데, 디버깅을 제대로 하지 못한 것이다.

쓰레드 없이 항목들에 접근할 경우, 컨트롤이 만들어진 스레드가 아닌 다른 스레드에서 조작하지 못한다는 경고를 하는 바람에 디버깅이 되지 않는 문제였다.
그 문제를 처음 겪은 날은 "CheckForIllegalCrossThreadCalls" 속성을 사용하지 않도록 설정하여 크로스스레딩 예외 상황을 검사하지 않도록 하여 개발을 진행할 수 있었고, 이런 상황으로 계속 사용하는 것도 보안상 해로운 방법이므로 학습을 통해서 delegate 객체와 그것을 조작하는 메소드를 구현함으로써 안전한 방식으로 프로그래밍 될 수 있었다.
디버깅이 되지 않는 문제를 해결하기 위해서는 델리게이트 객체를 만들어야 한다.

```
11  namespace PC_Vulnerability
12  {
        참조 4개
13      public partial class Form1 : Form
14      {
15          // 해결 : 크로스 스레드 작업이 잘못되었습니다. '*' 컨트롤이 자신이 만들어진 스레드가 아닌 스레드에서 액세스되었습니다.
16          delegate void Ctr_Involk(Control ctr, string text);
17          // 해결 : 크로스 스레드 작업이 잘못되었습니다. '*' 컨트롤이 자신이 만들어진 스레드가 아닌 스레드에서 액세스되었습니다.
18          JsonObjectCollection obj = new JsonObjectCollection();
            참조 3개
```

[그림 3-14] 크로스 스레드 작업 (1)

다음은 델리게이트 객체를 활용해서 메소드를 생성한다.

```
public void SetText(Control ctr, string txtValue)
{
    if (ctr.InvokeRequired)
    {
        Ctr_Involk CI = new Ctr_Involk(SetText);
        ctr.Invoke(CI, ctr, txtValue);
    }
    else
    {
        ctr.Text = txtValue;
    }
}
```

[그림 3-15] 크로스 스레드 작업 (2)

마지막으로, 이 델리게이트 객체의 메소드를 호출함으로써 값을 변경해야 한다.

```
private void GetSystemInfo()
{
    JsonArrayCollection rows1 = new JsonArrayCollection("System Info");
    JsonObjectCollection items = new JsonObjectCollection();
    ClassA a = new ClassA();
    // 해결 : 크로스 스레드 작업이 잘못되었습니다. '*' 컨트롤이 자신이 만들어진 스레드가 아닌 스레드에서 액세스되었습니다.
    // SetText() 메소드 구현 대처
    // 기존: IPAddress.Text = a.GetIP();
    SetText(IPAddress, a.GetIP());
    // 해결 끝: 크로스 스레드 작업이 잘못되었습니다. '*' 컨트롤이 자신이 만들어진 스레드가 아닌 스레드에서 액세스되었습니다.
    items.Add(new JsonStringValue("IP Address", IPAddress.Text));
    SetText(PCName, a.PCName());
    items.Add(new JsonStringValue("PC Name", PCName.Text));
    SetText(OSName, a.OSName());
    items.Add(new JsonStringValue("OS Name", OSName.Text));
    SetText(Lang, a.GetLang());
    items.Add(new JsonStringValue("Language", Lang.Text));
    SetText(SystemMaker, a.SystemMaker()[0]);
    items.Add(new JsonStringValue("System Maker", SystemMaker.Text));
    SetText(SystemModel, a.SystemMaker()[1]);
    items.Add(new JsonStringValue("System Model", SystemModel.Text));
    SetText(Bios, a.BIOS());
    items.Add(new JsonStringValue("BIOS", Bios.Text));
    SetText(Processor, a.Processor());
    items.Add(new JsonStringValue("Processor", Processor.Text));
    SetText(Memory, a.Memory());
    items.Add(new JsonStringValue("Memory", Memory.Text));
    rows1.Add(items);
    obj.Add(rows1);
}
```

[그림 3-16] 크로스 스레드 작업 (3)

3.3.0 웹페이지 기능

프로그램 실행 시, 웹 연동을 하기 위해 데이터를 JSON 형식을 작성하였다.


```
"StartProgram": {
        "Count": {
                "MsgStartLen": "2",
                "MsgStartName": "KakaoTalk, KakaoTalk(W)",
                "TotalStartLen": "13"
        },
        {
                "Use": "1",
                "Key": "HKLM/Run(64)",
                "Program": "SecurityHealth",
                "Maker": "Microsoft Corporation",
                "User": "모든 사용자",
                "File": "C:₩₩Windows₩₩system32₩₩SecurityHealthSystray.exe"
        },
        {
                "Use": "1",
                "Key": "HKLM/Run(64)",
                "Program": "RtHDVCpl",
                "Maker": "Realtek Semiconductor",
                "User": "모든 사용자",
                "File": "C:₩₩Program Files₩₩Realtek₩₩Audio₩₩HDA₩₩RAVCpl64.exe"
        },
        {
                "Use": "1",
```


[그림 3-17] JSON 형식의 일부

JSON은 Javascript 객체 리터럴 문법을 따르는 문자열이다. JSON 안에도 Javascript의 기본 데이터 타입인 문자열, 숫자, 배열 그리고 다른 객체를 포함할 수 있다.

프로그램과 JSON 작성을 통해 웹 페이지(https://jbt.clsw.kr)를 요청함으로써, 이를 웹기능으로 연동하였다. 밑에 그림은 발표 PPT에 관한 자료이다. 클릭하면 PPT에 대한 내용을 확인할 수 있다.

[그림 3-18] 메인화면

프로젝트의 조원 소개와 각각의 조원들의 프로그램 제작에 대한 역할 등을 확인할 수 있다.

[그림 3-19] 프로젝트 조원 및 역할

그리고, 우리가 프로젝트의 주제를 취약점 진단에 대해 연구하였는데, 그에 대한 필요성 및 주제를 선정한 이유를 알 수 있다.

[그림 3-20] 프로젝트 선정 이유

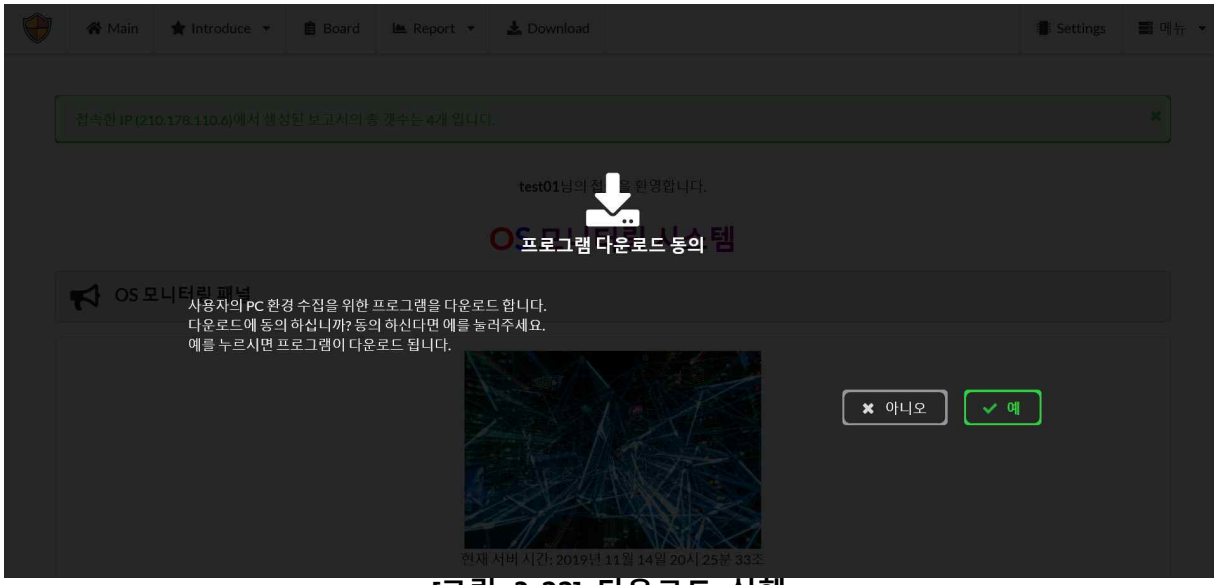
취약점에는 다양한 기능들을 제공하는데 먼저 우측 상단에서 Sign-in 버튼을 누르고 로그인을 진행한다. 여기서 취약점 분석의 핵심 기능들이 화면에 나오면서 다양한 기능들을 제공한다. 종합적인 데이터를 이용하여 시각적인 정보를 제공함으로써 취약점 분석 내용을 시각 정보로 제공하는 기능이다. 이것을 토대로 제작한 소프트웨어를 실행해서, 여러 가지 다양한 기능들을 제공함으로써 보안의 취약점을 자동으로 진단하는 등 모니터링을 할 수 있도록 운영한다.

3.3.1 로그인 페이지

로그인 페이지는 각각의 사용자를 나누기 위해 로그인하면 웹을 통한 서비스의 사용이 가능하다.

[그림 3-21] 로그인 기능

자신의 PC에 대한 취약점을 분석하기 위해 다운로드를 한다.



[그림 3-22] 다운로드 실행

이를 실행하면 PC에 대한 정보 및 취약점 목록들을 확인할 수 있다.
여기서 대표적으로 시작 프로그램에 대한 정보인데, 시작 프로그램이란, 윈도우 시작 시 자동으로 실행되는 프로그램으로, 각각의 실행 프로그램 중에 취약한 소프트웨어들이 시작 프로그램으로 작동하는 경우가 많으므로, 취약한 경우 시작 프로그램에서 확인되는 경우도 많다. 여기서 빨간색으로 칠해진 부분은 프로그램 중에 취약할 수 있는 경우이다.
[그림 3-23]에서 메신저 소프트웨어가 진단되는 데, 메신저 소프트웨어 설치 및 사용이 가장 많은 내부 정보의 제3자 유출을 야기하므로 취약하다고 판단되었다.

[그림 3-23] 진단 프로그램 실행

진단 프로그램의 실행을 통해 시각적인 정보에 있는 데이터 안에 서버의 취약점 탐지 횟수의 수집 내역들을 기록할 수 있다. 대상 PC에서 진단 프로그램을 실행해서 결과 보고서를 웹서버에 전송하여 웹 서버에서 시각화하여 보여주게 됨으로써 웹으로 진단 결과를

확인이 가능하다.

수집 내역

Idx	Name	IP	Time	OS
7	DESKTOP-C66KL2I	14.42.86.31	2019-10-22 14:43:59	Microsoft Windows 10 Home 64비트
6	DESKTOP-25ATM5L	14.42.86.31	2019-10-22 14:39:06	Microsoft Windows 10 Pro 64비트
5	DESKTOP-25ATM5L	14.42.86.31	2019-10-22 14:29:45	Microsoft Windows 10 Pro 64비트
4	DESKTOP-C66KL2I	14.42.86.31	2019-10-22 14:29:14	Microsoft Windows 10 Home 64비트
3	DESKTOP-C66KL2I	14.42.86.31	2019년 10월 22일 14시 23분 50초	Microsoft Windows 10 Home 64비트
2	DESKTOP-25ATM5L	14.42.86.31	2019-10-22 14:23:51	Microsoft Windows 10 Pro 64비트
1	DESKTOP-C66KL2I	14.42.86.31	2019년 10월 22일 14시 23분 01초	Microsoft Windows 10 Home 64비트

처음 1 마지막 전체 개수:7개

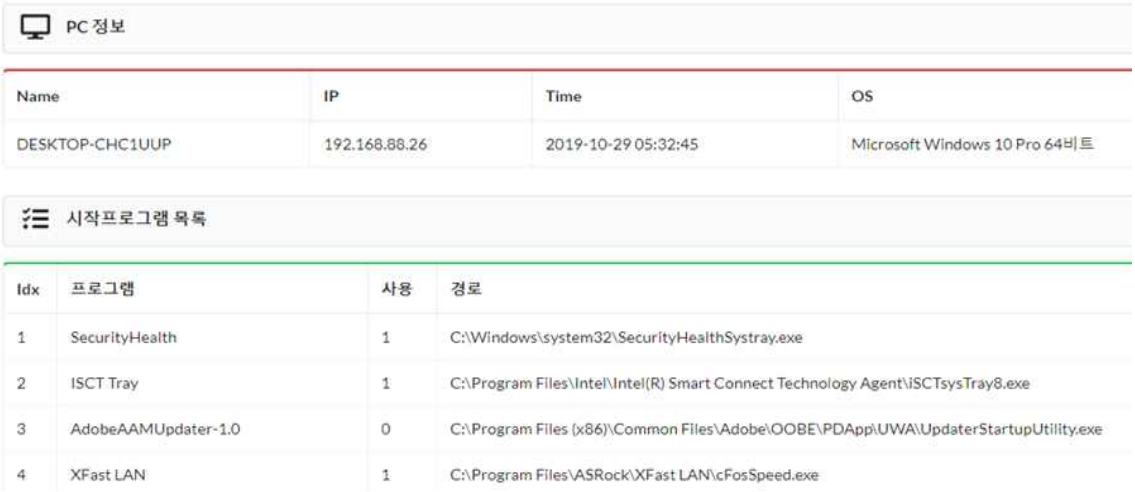
[그림 3-24] 취약점 수집내역

프로그램에 의해 수집된 자료들을 통해서 API라는 것을 사용하였는데, API란 컴퓨터 운영체제의 기능과 그 기능을 사용하는 방법을 정의한 함수이며, 실행 프로그램과 웹서버 간의 전송을 위해 사용하게 되었다. API를 통해 결과 보고서를 서버에 보내고 그 결과를 어디서든 확인해볼 수 있도록 웹 소스코드를 통해 작성하였다.

```php
<?php
include "../lib/db.php";
?>

<?php
$conn = mysqli_connect("localhost:3307", "test_db", "qfktkdcjf1", "test_db"); // 데이터베이스 연결
if (mysqli_connect_errno()) {
    echo "MySQL 연결 오류 : " . mysqli_connect_error();
}
echo "DB 접속 OK  ";
?><br/><br/><?php

// 이거 나중에 고쳐야 함 !!!
$select_query = "SELECT idx,Name FROM Host";
$result_set   = mysqli_query($conn, $select_query);
$count        = mysqli_num_rows($result_set);
$tmp          = $count + 1;

$Name = $_GET['HostName'];
$IP = $_SERVER['REMOTE_ADDR'];
$OS = $_GET['OS'];
$time = $_GET['time'];

echo "번호: " . $count . ", 호스트: ", $_GET['HostName'] . ", IP: ", $_SERVER['REMOTE_ADDR'] .
", OS: ", $_GET['OS'] . ", Time: ", $_GET['time'];
```

[그림 3-25] API 구현 부분

[그림 3-24]에서 취약점 수집 내역 부분들을 보면 PC에 대한 수집 내역들이 나오는데 그 중에서 접속한 PC의 한 부분을 클릭하면 원격에서 대상 PC들에 대한 정보 및 취약점 결과 등에 대한 확인이 가능하다는 것을 알 수 있다.



PC 정보

Name	IP	Time	OS
DESKTOP-CHC1UUP	192.168.88.26	2019-10-29 05:32:45	Microsoft Windows 10 Pro 64비트

시작프로그램 목록

Idx	프로그램	사용	경로
1	SecurityHealth	1	C:\Windows\system32\SecurityHealthSystray.exe
2	ISCT Tray	1	C:\Program Files\Intel\Intel(R) Smart Connect Technology Agent\iSCTsysTray8.exe
3	AdobeAAMUpdater-1.0	0	C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\UWA\UpdaterStartupUtility.exe
4	XFast LAN	1	C:\Program Files\ASRock\XFast LAN\cFosSpeed.exe

[그림 3-26] 원격 PC 목록

3.3.2 보고서 API

응답 서버는 프로그램에서 전송하는 "JSON"파일 형식의 보고서를 POST 데이터로 받고 이 데이터는 가공되어 각각의 파트별로 분배되어 DB에 저장되게 된다. 현재는 서버 문제로 코드를 별도로 가공하여 데이터를 처리한다.

```
102
103 // 컨텐츠 타입이 JSON 인지 확인한다
104 if(!in_array('application/json',explode(';',$_SERVER['CONTENT_TYPE']))){
105 echo json_encode(array('result_code' => '400'));
106 exit;
107 }
108
109
110 $__rawBody = file_get_contents("php://input"); // 본문을 불러옴
111 //$__getData = array(json_decode($__rawBody)); // 데이터를 변수에 넣고
112 //$text = str_replace(array("\r\n","\r","\n"),'',$text);
113 $__getData = str_replace("\"", "",str_replace("\\\\","\\", preg_replace('/\r|\n|\t/','',array($__rawBody))));
114
115 $odata = json_decode($__rawBody);
116
117 $aa = "";
118
119 $Time = $odata->Time;
120 $MsgStartCount = $odata->StartProgram->Count->MsgStartCount; // 윈도우 실행시 실행되는 메신저 갯수
121 $MsgStartName = $odata->StartProgram->Count->MsgStartName; // 메신저 이름
122 $TotalStartCount = $odata->StartProgram->Count->TotalStartCount; // 시작프로그램 갯수
123 $count_cve = count($odata->CVEList);
124
125 //System Info
126 $IPAddr = explode(' ', $odata->SystemInfo[0]->IPAddress);
127 $PCName = $odata->SystemInfo[0]->PCName;
128 $OSName = explode(' (', $odata->SystemInfo[0]->OSName);
129
130 //Service Mangement
131 $WMState = $odata->ServiceManagerment[0]->WMState;
132 $Messenger_Name = $odata->ServiceManagerment[0]->Messenger_Name;
133 $BootArea_Name = $odata->ServiceManagerment[0]->BootArea_Name;
134 $PortOpen_TCP = $odata->ServiceManagerment[0]->PortOpen_TCP;
135 $PortOpen_UDP = $odata->ServiceManagerment[0]->PortOpen_UDP;
136 $XService_Recommand = $odata->ServiceManagerment[0]->XService_Recommand;
137 $XService_Select = $odata->ServiceManagerment[0]->XService_Select;
```

[그림 3-27] 보고서 응답 API 페이지

+ 옵션

칼럼보이기 여부를 바꾸려면 드롭다운 화살표를 클릭하세요.

				Name
수정	복사	삭제		MSXML 6.0 RTM 보안 업데이트(925673), Microsoft Visual C+...
수정	복사	삭제	2	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	3	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	4	Windows 악성 소프트웨어 제거 도구 x64 - 2019년 8월(KB890830), 2...
수정	복사	삭제	5	Windows 악성 소프트웨어 제거 도구 x64 - 2019년 8월(KB890830), 2...
수정	복사	삭제	6	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	7	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	8	Windows 악성 소프트웨어 제거 도구 x64 - 2019년 8월(KB890830), 2...
수정	복사	삭제	9	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	10	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	11	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	12	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	13	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...
수정	복사	삭제	14	2019-02 x64 기반 시스템용 Windows 10 Version 1809 업데이트(K...
수정	복사	삭제	15	2019-09 x64 기반 시스템용 Windows 10 Version 1903의 Adobe...

[그림 3-28] DB에 저장된 값

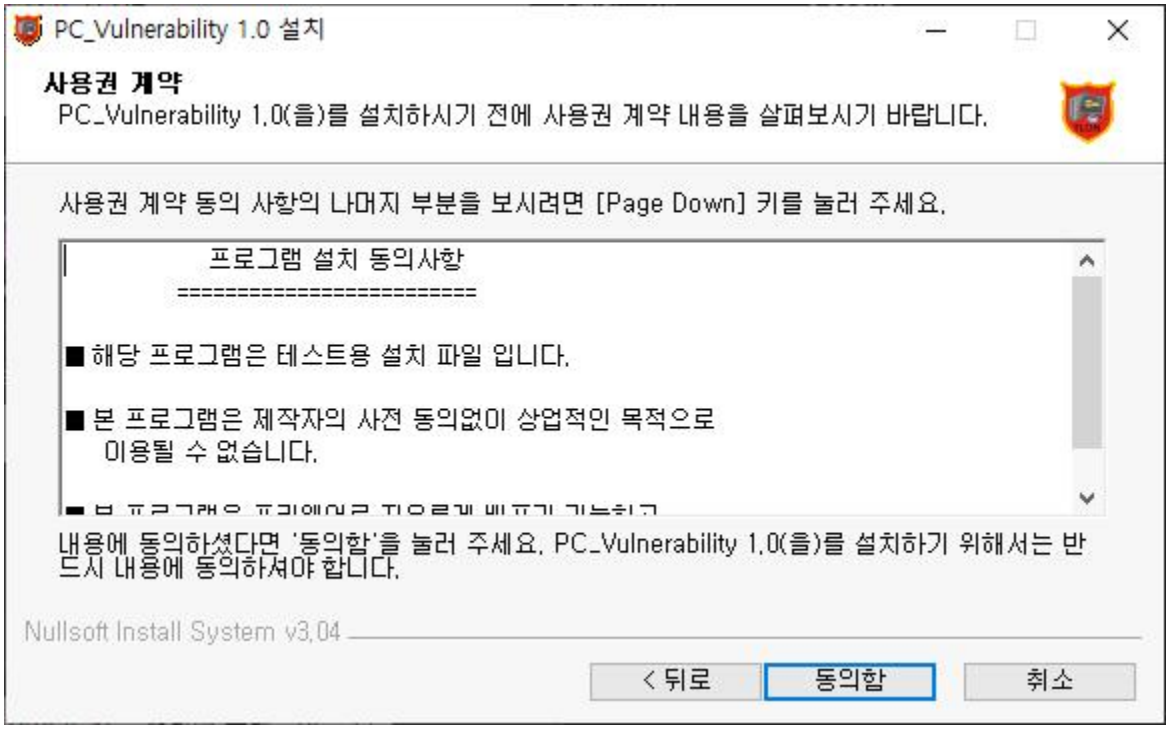
+ 옵션

칼럼보이기 여부를 바꾸려면 드롭다운 화살표를 클릭하세요.

			idx	Product	CVE_ID	CVE_Score
수정	복사	삭제	1	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3, 7.2
수정	복사	삭제	2	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	3	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	4			
수정	복사	삭제	5			
수정	복사	삭제	6			
수정	복사	삭제	7	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	8	wireshark 3.0.	CVE-2019-16319	7.8
수정	복사	삭제	9	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	10	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	11	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	12	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	13	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3
수정	복사	삭제	14	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3, 9.3,...
수정	복사	삭제	15	adobe_air 32.0.0.125, adobe_air 32.0.0.125, adobe_...	CVE-2013-0650, CVE-2013-0646, CVE-2013-1375, CVE-2...	10, 10, 10, 10, 9.3, 9.3, 9.3, 9.3, 4.3, 4.3

[그림 3-29] DB에 저장된 값 2

3.3.3 셋업 파일



[그림 3-30] 프로그램 설치파일 제작

사용자가 쉽게 프로그램을 설치 할 수 있도록 별도의 설치 파일을 제작하였습니다.
이는 프로그램 관리 및 유지를 용이하게 할 수 있고, 별도의 설정이 필요한 경우 일괄적으로 처리할 수 있습니다. 또한 아래 상기할 URI_Scheme 기능을 이용할 수 있도록 할 수 있습니다.

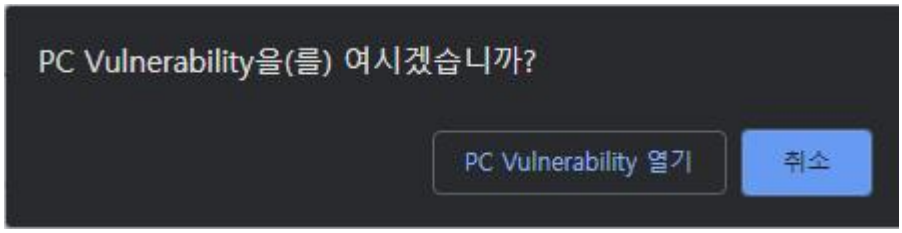
3.3.4 Custom URI_Scheme

앱의 패키지, 앱의 데이터 폴더 또는 클라우드에서 제공하는 파일을 참조하는 데 사용할 수 있는 몇 가지 URI(Uniform Resource Identifier) 체계가 있습니다. URI 체계를 사용하여 앱의 리소스 파일(.resw)에서 로드되는 문자열을 참조할 수도 있습니다. 코드, XAML 태그, 앱 패키지 매니페스트, 또는 사용자 타일 및 알림 메시지 템플릿에서 이러한 URI 스키마를 사용할 수 있습니다.

즉, URI에는 세 가지 기본 구성 요소가 있습니다. URI 스키마의 두 슬래시 바로 뒤에 나오는 것은 기관이라 불리는 구성 요소입니다(비어 있을 수 있음). 그리고 그 뒤를 바로 따르는 것이 경로입니다. URI http://www.contoso.com/welcome.png를 예로 들면 스키마는 "http://", 기관은 "www.contoso.com", 경로는 "/welcome.png"입니다. 다른 예로 URI ms-appx:///logo.png의 경우 기관 구성 요소는 비어 있으며 기본값을 가집니다.

마이크로소프트 발췌(https://docs.microsoft.com/ko-kr/windows/uwp/app-resources/uri-schemes)

해당 기능을 사용하면 웹 페이지 내에서 "http://" 와 같이 사용자 지정 프로토콜을 생성할 수 있으며, 이는 사용자가 손쉽게 프로그램을 실행할 수 있도록 합니다.



[그림 3-31] 프로그램 실행

이와 같이 웹 페이지 상에서 프로그램을 바로 실행할 수 있게 됩니다.

```
Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\clsw]
"URL Protocol"=""

[HKEY_CLASSES_ROOT\clsw\shell]
@=""

[HKEY_CLASSES_ROOT\clsw\shell\open]
@=""

[HKEY_CLASSES_ROOT\clsw\shell\open\command]
@="C:\Program Files (x86)\PC_Vulnerability\PC_Vulnerability.exe“
```

<사용 예시>

4. 결 론

PC 보안에 대한 취약점 및 탐색 부분을 C#으로 작성해 코드를 구현하였다. 이에 대해 프로그램의 개발을 진행하면서, public 함수 대신에 static 함수를 구현함으로써 캡슐화 프로그래밍 방법과 멀티쓰레드 방법을 습득하였고, 윈도우는 레지스트리를 통해 설정 값을 저장해 두고 있어 저장된 설정값을 가져와서 인터페이스화하게 되었고, 다양한 취약점 및 외부 탐지로부터 보호하기 위해 모니터링 시스템을 통해 더 많은 부분들을 실무자가 빠르게 파악하여 기능을 실용화하여 효과를 볼 수 있게 하였다.

C#으로 작성된 코드를 웹 연동을 하기 위해 JSON을 사용하여 웹 개발로 이어지고, 에이전트에서 실행 시 서버 db에 올리고 그 값을 웹페이지에서 받아와 다양한 웹 기능들이 동작하는 것을 확인하였다.

프로그램을 개발 중에 보안 취약점 'CVE'를 종합적으로 진단하는 방법을 발견하여 CVE 위반 취약점까지 하나의 프로그램을 통해 여러 가지 취약점들을 손쉽게 확인 가능하였고, 웹을 통하여 결과를 확인할 수 있게 되어 그에 대한 기대효과로 운영체제를 진단하는데 소요되는 시간이 적어지는 것으로 할 수 있다.

KISA 취약점 분석 평가 가이드에 기반하여 이를 어떻게 대비할 것인가에 대한 기본적인 위험을 인지하고, 각각의 점검 항목들을 확인하고, 취약점에 대해 보다 신속하고 정확한 대응이 가능하여 안정적인 보안체제의 구축 운영이 기대되어 진다.

5. 별 첨

5.1 프로그램 소스

<SystemInfo.cs>

```
using System;
using System.Globalization;
using System.Management;
using System.Net;
using System.Net.Sockets;
using System.Text.RegularExpressions;

class ClassA
{
    Singleton singleton;
    public ClassA()
    {
        singleton = Singleton.GetInstance();
    }
    internal string GetIP()
    {
        string ExternalIP = string.Empty;
        try
        {
            using (var web = new WebClient())
            {
                ServicePointManager.SecurityProtocol |= SecurityProtocolType.Ssl3;
                ServicePointManager.SecurityProtocol |= SecurityProtocolType.Tls;
                ServicePointManager.SecurityProtocol |= SecurityProtocolType.Tls11;
                ServicePointManager.SecurityProtocol |= SecurityProtocolType.Tls12;
                ExternalIP = new
WebClient().DownloadString("https://jbt.clsw.kr/API/getip.php");
            }
            //ExternalIP = new
WebClient().DownloadString("https://jbt.clsw.kr/API/getip.php");
            ExternalIP = ExternalIP.Substring(ExternalIP.IndexOf(":") + 2, 15); // Top IP
            //ExternalIP = ExternalIP.Substring(ExternalIP.IndexOf("<br>") + 23); //
Bottom IP
            ExternalIP = Regex.Match(ExternalIP,
@"[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}").Value;
        }
        catch
        {
            return "Not connect";
        }
        try
        {
            IPHostEntry host = Dns.GetHostEntry(Dns.GetHostName());
            string ClusterIP = string.Empty;
            foreach (var ip in host.AddressList)
            {
                if (ip.AddressFamily == AddressFamily.InterNetwork)
                {
                    ClusterIP = ip.ToString();
                }
```

```
            }
            if (ClusterIP != String.Empty) return ExternalIP + " (" + ClusterIP + ")";
            else return ExternalIP;
        }
        catch
        {
            return "";
        }
    }
    internal string PCName()
    {
        return Environment.MachineName;
    }
    internal string OSName()
    {
        string result = "Not found!";
        ConnectionOptions options = new ConnectionOptions();
        options.Impersonation = ImpersonationLevel.Impersonate;
        ManagementScope scope = new ManagementScope("/root/cimv2", options);
        scope.Connect();
        ObjectQuery query = new ObjectQuery("Select Caption From
Win32_OperatingSystem");
        ManagementObjectSearcher search = new ManagementObjectSearcher(scope,
query);
        ManagementObjectCollection queryCollection = search.Get();
        foreach (ManagementObject o in queryCollection)
        {
            result = o["Caption"].ToString();
            if (Environment.Is64BitOperatingSystem)
            {
                result = result + " 64비트 ";
            }
            else
            {

                result = result + " 32비트 ";
            }
            string ver = GetOSVer();
            int len = ver.Length;
            result = result + "(" + ver.Substring(0, len - 6) + ", 빌드 " +
ver.Substring(len - 5) + ")";
        }
        return result;
    }
    private static ManagementObject GetMngObj(string className)
    {
        var wmi = new ManagementClass(className);

        foreach (var o in wmi.GetInstances())
        {
            var mo = (ManagementObject)o;
            if (mo != null) return mo;
        }

        return null;
    }
```

```
    private string GetOSVer()
    {
        ManagementObject mo = GetMngObj("Win32_OperatingSystem");

        if (null == mo)
            return string.Empty;

        return mo["Version"] as string;
    }
    internal string GetLang()
    {
        return CultureInfo.CurrentCulture.Name.Replace("ko-KR", "한국어") + " (국가별
설정: " +
            CultureInfo.CurrentUICulture.Name.Replace("ko-KR", "한국어") + ")";
    }
    internal string[] SystemMaker()
    {
        string[] Maker = { null, null };
        ManagementClass mc = new ManagementClass("Win32_ComputerSystem");
        ManagementObjectCollection moc = mc.GetInstances();
        if (moc.Count != 0)
        {
            foreach (ManagementObject mo in mc.GetInstances())
            {
                Maker[0] = mo["Manufacturer"].ToString(); // 시스템 제조업체
                Maker[1] = mo["model"].ToString(); // 시스템 모델
            }
        }
        return Maker;
    }
    internal string BIOS()
    {
        string bios = null;
        ManagementObjectSearcher searcher1 = new
ManagementObjectSearcher("SELECT * FROM Win32_BIOS");
        ManagementObjectCollection collection = searcher1.Get();
        foreach (ManagementObject obj in collection)
        {
            if (((string[])obj["BIOSVersion"]).Length > 1)
                bios = ((string[])obj["BIOSVersion"])[((string[])obj["BIOSVersion"]).Length
- 2];
            else
                bios = ((string[])obj["BIOSVersion"])[0];
        }
        return bios;
    }
    internal string Processor()
    {
        string processor = null;
        ManagementObjectSearcher searcher2 = new
ManagementObjectSearcher("Select * from Win32_Processor");
        foreach (ManagementObject process in searcher2.Get())
        {
            processor = process["Name"].ToString();
            processor += " (" + Environment.ProcessorCount + "CPUs), ~";
            double ghz = 0.001f * (uint)process["MaxClockSpeed"];
```

```
                processor += ghz.ToString("N1") + "GHz";
            }
            return processor;
        }
        internal string Memory()
        {
            string memory = null;
            while (true)
            {
                PERFORMANCE_INFORMATION pi = new PERFORMANCE_INFORMATION();
                pi.Initialize();
                SafeNativeMethods.GetPerformanceInfo(out pi, pi.cb);
                SafeNativeMethods.GetPhysicallyInstalledSystemMemory(out ulong
installedMemory);
                MEMORYSTATUSEX globalMemoryStatus = new MEMORYSTATUSEX();
                globalMemoryStatus.Initialize();
                SafeNativeMethods.GlobalMemoryStatusEx(ref globalMemoryStatus);
                if (installedMemory >= 1024)
                {
                    memory = $"{installedMemory / 1024} MB RAM";
                    break;
                }
                else
                {
                    memory = $"{installedMemory} RAM";
                    break;
                }
            }
            return memory;
        }
}
```

<StartupProgram.cs>

```
using Microsoft.Win32;
using System;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Runtime.InteropServices;
using System.Text;
using System.Windows.Forms;

class ClassB
{
    Singleton singleton;
    public ClassB()
    {
        singleton = Singleton.GetInstance();
    }
    private string Messenger = null;
    private int MessengerLen = 0;
    // 시작 프로그램 32비트 프로그램 정보를 가져오기 위해
    [DllImport("kernel32.dll", SetLastError = true)]
    static extern bool Wow64DisableWow64FsRedirection(ref IntPtr ptr);
    // 시작 프로그램 64비트 프로그램 정보를 가져오기 위해
```

```
    [DllImport("kernel32.dll", SetLastError = true)]
    static extern bool Wow64RevertWow64FsRedirection(IntPtr ptr);
    // Spcial Folder 정보 가져오기 (시작 폴더 계산을 위해 사용)
    [DllImport("shell32.dll")]
    static extern bool SHGetSpecialFolderPath(IntPtr hwndOwner, [Out] StringBuilder
lpszPath, int nFolder, bool fCreate);
    const int CSIDL_COMMON_STARTUP = 0x0018;

    internal void GetStartProReg(ListView lvw, string type)
    {
        try
        {
            IntPtr wow64Value = IntPtr.Zero;
            // 레지스트리 경로
            string path = @"SOFTWARE\Microsoft\Windows\CurrentVersion\Run";
            // 사용 유무 정보 확인을 위해서 다른 경로 사용
            string path2 =
@"SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run";
            // 한 번만 실행
            string pathone =
@"SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce";
            // 레지스트리 액세스 :
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
            var baseReg = RegistryKey.OpenBaseKey(RegistryHive.ClassesRoot,
RegistryView.Default);
            if (type == "HKLM64")
            {
                // 64비트 레지스트리 리다이렉션 비활성화 (실제 접근을 위해)
                Wow64DisableWow64FsRedirection(ref wow64Value);
                baseReg = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine,
RegistryView.Registry64);
                baseReg = baseReg.OpenSubKey(path, false);
            }
            else if (type == "HKLM32")
            {
                baseReg = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine,
RegistryView.Registry32);
                baseReg = baseReg.OpenSubKey(path, false);
            }
            else if (type == "HKCU")
            {
                baseReg = RegistryKey.OpenBaseKey(RegistryHive.CurrentUser,
RegistryView.Registry64);
                baseReg = baseReg.OpenSubKey(path, false);
            }
            else if (type == "HKLMO")
            {
                baseReg = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine,
RegistryView.Registry64);
                baseReg = baseReg.OpenSubKey(pathone, false);
            }
            else if (type == "HKCUO")
            {
                baseReg = RegistryKey.OpenBaseKey(RegistryHive.CurrentUser,
RegistryView.Registry64);
                baseReg = baseReg.OpenSubKey(pathone, false);
```

```
            }
            foreach (string program in baseReg.GetValueNames())
            {
                ListViewItem StartLVI = new ListViewItem();

                var status = RegistryKey.OpenBaseKey(RegistryHive.ClassesRoot,
RegistryView.Default);
                if (type == "HKLM64")
                {
                    status = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine,
RegistryView.Registry64);
                    status = status.OpenSubKey(path2, false);
                }
                else if (type == "HKLM32")
                {
                    status = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine,
RegistryView.Registry64);
                    status = status.OpenSubKey(path2, false);
                }
                else if (type == "HKCU")
                {
                    status = RegistryKey.OpenBaseKey(RegistryHive.CurrentUser,
RegistryView.Registry64);
                    status = status.OpenSubKey(path2, false);
                }
                if (status == null) StartLVI.Text = "E1";
                if (type == "HKLM64") StartLVI.SubItems.Add("HKLM/Run(64)");
                else if (type == "HKLM32") StartLVI.SubItems.Add("HKLM/Run(32)");
                else if (type == "HKCU") StartLVI.SubItems.Add("HKCU/Run");
                else if (type == "HKLMO")
                {
                    StartLVI.Text = "O";
                    StartLVI.SubItems.Add("HKLM/RunOnce");
                }
                else if (type == "HKCUO")
                {
                    StartLVI.Text = "O";
                    StartLVI.SubItems.Add("HKCU/RunOnce");
                }
                if (program == "") continue; // 정보가 비어있으면 다음으로 이동
                string StartPath = baseReg.GetValue(program).ToString(); // 파일 경로
(매개변수 정보까지)
                string RealPath = StartPath; // 복사
                if (RealPath.IndexOf("\"") == 0) RealPath = RealPath.Substring(1); // 첫
글자가 "인 경우 다음 글자부터
                if (RealPath.Contains(".exe")) RealPath = RealPath.Substring(0,
RealPath.IndexOf(".exe")) + ".exe";
                // 사용유무
                if (type != "HKCUO" && type != "HKLMO")
                {
                    if (status.GetValue(program) == null) StartLVI.Text = "O";
                    else
                    {
                        byte[] StartusByte = (byte[])status.GetValue(program);
                        if (StartusByte == null) StartLVI.Text = "O";
                        else
```

```
                        {
                            var StatusStr =
BitConverter.ToString(StartusByte).Substring(0, 2);
                            if (Convert.ToInt32(StatusStr) % 2 == 0) StatusStr = "O";
                            else if (Convert.ToInt32(StatusStr) % 2 == 1) StatusStr =
"X";
                            else StatusStr = "E2";
                            StartLVI.Text = StatusStr;
                        }
                    }
                }
                else
                {
                    StartLVI.Text = "O";
                }
                // 프로그램
                StartLVI.SubItems.Add(program);

                // 게시자
                if (File.Exists(RealPath)) // 파일이 있는 경우만 수행
                {
                    FileVersionInfo FI = FileVersionInfo.GetVersionInfo(RealPath); // 파
일 실제 경로로 파일 정보 습득
                    if (FI.CompanyName != null)
                    {
                        StartLVI.SubItems.Add(FI.CompanyName); // 회사 정보
                        string[] MSG_Name = { "KakaoTalk", "Skype for Desktop",
"LINE", "NATEON" };
                        string[] MSG_Company = { "Kakao Corp", "Skype
Technologies", "LINE Corporation", "SK Communications" };

                        for (int i = 0; i < MSG_Name.Length; i++)
                        {
                            if (FI.CompanyName.Contains(MSG_Company[i]) == true &&
program.Contains(MSG_Name[i]) == true)
                            {
                                MessengerLen = MessengerLen + 1;
                                StartLVI.ForeColor = Color.Red;
                                Messenger += program + ", ";
                            }
                        }
                    }
                    else StartLVI.SubItems.Add("E4");
                }
                else
                {
                    StartLVI.SubItems.Add("");
                }

                // 사용자
                if (type == "HKCU" || type == "HKCUO")
                {
                    string fullName =
System.Security.Principal.WindowsIdentity.GetCurrent().Name;
                    string[] temp = fullName.Split('\\');
                    StartLVI.SubItems.Add(temp[1]);
```

```
                }
                else StartLVI.SubItems.Add("모든 사용자");

                // 파일
                StartLVI.SubItems.Add(RealPath); // 전체 경로(매개변수 포함)


                // StartLVI 내용 실제 추가
                lvw.Items.Add(StartLVI);

                // 정보 접근을 마쳤으므로 64비트 레지스트리 리다이렉션 다시 활성화
                Wow64RevertWow64FsRedirection(wow64Value);
            }
        }
        catch
        {
        }
    }
    internal void GetStartProFile(ListView lvw, string type)
    {
        string dirPath = string.Empty;
        if (type == "U") dirPath = Environment.GetFolderPath(Environment.SpecialFolder.Startup);
        else if (type == "A") dirPath = Environment.GetFolderPath(Environment.SpecialFolder.CommonStartup);
        if (Directory.Exists(dirPath))
        {
            DirectoryInfo di = new DirectoryInfo(dirPath);

            foreach (var item in di.GetFiles())
            {
                ListViewItem StartLVI = new ListViewItem();
                if (item.ToString() == "desktop.ini") continue;

                StartLVI.Text = "O";
                if (type == "U") StartLVI.SubItems.Add("StartUp(User) Folder");
                else if (type == "A") StartLVI.SubItems.Add("StartUp(All) Folder");
                StartLVI.SubItems.Add(item.Name);
                StartLVI.SubItems.Add("(만듬) " + item.CreationTime.ToString());
                if (type == "U")
                {
                    string fullName = System.Security.Principal.WindowsIdentity.GetCurrent().Name;
                    string[] temp = fullName.Split('\\');
                    StartLVI.SubItems.Add(temp[1]);
                }
                else StartLVI.SubItems.Add("모든 사용자");
                // StartLVI 내용 실제 추가
                lvw.Items.Add(StartLVI);
                StartLVI.SubItems.Add(item.FullName);

                // 정렬
                for (int width = 0; width < lvw.Columns.Count; width++)
                {
                    if (width == 0) lvw.Columns[width].Width = 50;
                    else lvw.Columns[width].Width = -2;
```

```
                }
            }
        }
    }
    internal void WinMsgAppCheck(ListView lvw)
    {
        string WinApp = SearchDirectory(Path.GetPathRoot(Environment.SystemDirectory) + @"\Program Files\WindowsApps");
        if (WinApp != null)
        {
            try
            {
                RegistryKey rkey = Registry.CurrentUser.OpenSubKey(@"Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData");
                string[] subKeynames = rkey.GetSubKeyNames();
                string WinMsg = "1CB77C17.17884BE984322"; // 카카오톡만 구현된 기능이므로 배열 미사용
                foreach (var item in subKeynames)
                {
                    if (item.Contains(WinMsg) == false) continue;
                    // 카카오톡만 구현된 기능이므로 배열 미사용
                    string regpath = @"Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData\" + item + @"\KakaoTalkUWP";
                    ListViewItem StartLVI = new ListViewItem();
                    using (var root = RegistryKey.OpenBaseKey(RegistryHive.CurrentUser, RegistryView.Registry64))
                    {
                        using (var key = root.OpenSubKey(regpath, false))
                        {
                            MessengerLen++;
                            StartLVI.ForeColor = Color.Red;
                            string test = key.GetValue("State").ToString();
                            if (test == "2") test = "O";
                            else if (test == "0") test = "X";
                            else test = "X";
                            string RealPath = Environment.ExpandEnvironmentVariables(WinApp);
                            StartLVI.Text = test;
                            StartLVI.SubItems.Add("Windows Apps");
                            FileVersionInfo FI = FileVersionInfo.GetVersionInfo(RealPath);
                            Messenger += FI.ProductName + "(" + "W" + "), ";
                            StartLVI.SubItems.Add(FI.ProductName);
                            StartLVI.SubItems.Add(FI.CompanyName);
                            // 사용자
                            string fullName = System.Security.Principal.WindowsIdentity.GetCurrent().Name;
                            string[] temp = fullName.Split('\\');
                            StartLVI.SubItems.Add(temp[1]);
                            StartLVI.SubItems.Add(RealPath); // 전체 경로(매개변수 포함)
                            lvw.Items.Add(StartLVI);
                        };
```

```
                    }
                }
            }
            catch
            {
                //lvw.Items.Add("[안내] 윈도우 앱스토어 접근에 문제가 생겼습니다.");
            }
        }
    }
    internal string SearchDirectory(String path)
    {
        try
        {
            foreach (string strdir in Directory.GetDirectories(path))
            {
                foreach (string strfile in Directory.GetFiles(strdir, "*Kakaotalk.exe"))
                {
                    return strfile;
                }
                    SearchDirectory(strdir);
            }
        }
        catch
        {
        }
        return string.Empty;
    }
    internal string TotalMSG()
    {
        Messenger = Messenger.Substring(0, Messenger.Length - 2);
        return Messenger;
    }
    internal string TotalPrint(ListView lvw)
    {
        lvw.Items.Clear();
        GetStartProReg(lvw, "HKLM64");
        GetStartProReg(lvw, "HKLM32");
        GetStartProReg(lvw, "HKCU");
        GetStartProReg(lvw, "HKLMO");
        GetStartProReg(lvw, "HKCUO");
        GetStartProFile(lvw, "U");
        GetStartProFile(lvw, "A");//전체: 모든 사용자
        WinMsgAppCheck(lvw);
        if (lvw.Items.Count > 0 && MessengerLen > 0) return " (" + MessengerLen +
"/" + lvw.Items.Count.ToString() + ")";
        if (lvw.Items.Count > 0) return " (0/" + lvw.Items.Count.ToString() + ")";
        return "";
    }
}
```

<Singleton.cs>

```
class Singleton
{
    private string item;
    private static Singleton singleton;
```

```
    private Singleton()
    {
        item = string.Empty;
    }

    public static Singleton GetInstance()
    {
        if (singleton == null) singleton = new Singleton();
        return singleton;
    }
}
```

<ServiceManagerment.cs>

```
using Microsoft.Win32;
using System;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Management;
using System.Net;
using System.Net.NetworkInformation;
using System.ServiceProcess;
using System.Text.RegularExpressions;
using System.Windows.Forms;

class ClassC
{
    Singleton singleton;
    public ClassC()
    {
        singleton = Singleton.GetInstance();
    }
    internal string WMState()
    {
        RegistryKey reg =
Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Policies\Microsoft\Messenger\Client",
true);
        if (reg != null)
        {
            Object val = reg.GetValue("PreventRun");
            int val_i = Convert.ToInt32(val);
            if (null != val)
            {
                if (val_i == 0) return "Enable"; // 허용
                else if (val_i == 1) return "Disable"; // 허용 안함
                else return Convert.ToString(val);
            }
        }
        return "Not setting";
    }
    internal string Messengers()
    {
        string MSGResult = null;
        int temp = 0;
```

```
        if (Messenger(0)) { MSGResult += "Kakaotalk, "; temp++; }
        if (Messenger(1)) { MSGResult += "Line, "; temp++; }
        if (Messenger(2)) { MSGResult += "Skype, "; temp++; }
        if (Messenger(3)) { MSGResult += "NateOn, "; temp++; }
        if (Messenger(4)) { MSGResult += "Kakaotalk Apps, "; temp++; }
        if (Messenger(5)) { MSGResult += "Line Apps, "; temp++; }
        if (Messenger(6)) { MSGResult += "Skype Apps, "; temp++; }
        if (Messenger(7)) { MSGResult += "Skype for Business, "; temp++; }
        try
        {
            MSGResult = MSGResult.Substring(0, MSGResult.Length - 2); // 마지막 공백
( )+콤마(,) 제거
            MatchCollection MessengerCount = Regex.Matches(MSGResult, ",");
            MSGResult = MessengerCount.Count + 1 + " (" + MSGResult + ")";
        }
        catch
        {
            MSGResult = "Not found"; // 수정, 메신저 미발견 시
        }
        return MSGResult;
    }
    internal bool Messenger(int check)
    {
        try
        {
            var drive = Path.GetPathRoot(Environment.SystemDirectory);
            bool installed = false;
            string test;
            if (check == 0)
            { // Kakaotalk Check
                test                                                                    =
Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Wi
ndows\CurrentVersion\Uninstall\KakaoTalk", "UninstallString", ""));
                if (File.Exists(test)) return true;
            }
            else if (check == 1)
            { // Line Check
                test                                                                    =
Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER\Software\Microsoft\Windo
ws\CurrentVersion\Uninstall\LINE", "UninstallString", ""));
                if (test != null)
                {
                    test                                                                =
Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER\Software\NHN
Corpration\LINE", "RunOnce", ""));
                    if (test != null)
                    { // Unistaller Not Remove
                        test = Environment.GetEnvironmentVariable(test); // Get
Registry Save File
                        if (File.Exists(test)) return true; // If Exist
                    }
                    else
                    { // Use Order Registry
                        test                                                            =
Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER\Software\Naver\LINE",
"RunOnce", ""));
```

```
                        if (test != null)
                        { // Uninstaller Not Remove
                            test = Environment.GetEnvironmentVariable(test); // Get
Registry Save File
                            if (File.Exists(test)) return true; // If Exist
                        }
                    }
                }
            }
            else if (check == 2)
            { // Skype Check
                test                                                                =
Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\Software\Microsoft\Wind
ows\CurrentVersion\Uninstall\Skype_is1", "UninstallString", ""));
                if (File.Exists(test)) return true;
            }
            else if (check == 3)
            { // NateOn - Not Use Registry, But Install Path Not Change > Default
Folder Check
                test = Environment.GetEnvironmentVariable(@"ProgramFiles(x86)") +
@"\SK Communications\NATEON\BIN\NateOnMain.exe";
                if (File.Exists(test)) return true;
                else
                { // Default Use x86(32bit), Same Up Path, But Bottom Check

                    test = Environment.GetEnvironmentVariable(@"ProgramFiles") +
@"\SK Communications\NATEON\BIN\NateOnMain.exe";
                    if (File.Exists(test)) { return true; }
                }
            }
            else if (check == 4)
            { // App(Windows Store) Kakao Check
                installed = WinStoreApp("17884BE984322");
                if (installed == true) return true;
            }
            else if (check == 5)
            { // App(Windows Store) Line Check
                installed = WinStoreApp("NAVER.LINE");
                if (installed == true) return true;
            }
            else if (check == 6)
            { // App(Windows Store) Skype Check
                installed = WinStoreApp("Microsoft.SkypeApp");
                if (installed == true) return true;
            }
            else if (check == 7)
            { // Skype for business
                try
                {
                    test                                                            =
Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Micr
osoft.Lync.15ClassicJoin.1", "", "a"));
                    if (test != null) return true;
                    else
                    {
                        test                                                        =
```

```
Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\IM
Providers\Lync", "FriendlyName", ""));
                        if (test != null) return true;
                    }
                }
                catch
                {
                    return false;
                }
            }
            return false;
        }
        catch
        {
            return false; // 수정, 메신저 미발견 시
        }
    }
    internal bool WinStoreApp(string AppName)
    {
        int i = 0;
        if (AppName == "Microsoft.SkypeApp")
        {
            string folder =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationD
ata), @"Packages");
            DirectoryInfo directorys = new DirectoryInfo(folder);
            DirectoryInfo[] find = directorys.GetDirectories("*" + AppName + "*.*");
            foreach (DirectoryInfo foundDir in find) { i = i + 1; }
        }
        else
        {
            var drive = Path.GetPathRoot(Environment.SystemDirectory);
            DirectoryInfo directorys = new DirectoryInfo(drive + @"Program
Files\WindowsApps");
            DirectoryInfo[] find = directorys.GetDirectories("*" + AppName + "*.*");
            foreach (DirectoryInfo foundDir in find) { i = i + 1; }
        }
        if (i >= 1) return true;
        else return false;
    }
    internal string GetBootInfo()
    {
        string result = null;
        try
        {
            ClassZ z = new ClassZ();
            string OSInfo = z.RunCommand("%windir%" + @"\sysnative\bcdedit /enum
ACTIVE"); // 휠성 상태의 BCD 항목 확인
            int BootCount = 0;
            var lines = new Regex(@"\r\n|\n|\r", RegexOptions.Singleline).Split(OSInfo);
// 1줄 뵬로 나누기
            string boot = "";
            foreach (var line in lines)
            { // 끝까지 반복
                if (line.Contains("description") && !line.Contains("Windows Boot
Manager"))
```

```
                { // description이 Windows Boot Manager인 경우 제외하고 모두!
                    if (boot != "") boot = boot + ", " + line.Remove(0, 20).Trim(); // 기
존 boot 설정되었다면 / 새 항목 쓰기
                    else boot = line.Remove(0, 20).Trim(); // 첫 항목인 경우 바로 추가
                    BootCount++; // 부팅 영역 개수 +1
                }
            }
            result += BootCount + " (" + boot + ")";
        }
        catch
        {
            result = "Not found";
        }
        return result;
    }
    internal string GetFileSystem()
    {
        string result = null;
        try
        {
            DriveInfo[] Drives = DriveInfo.GetDrives();
            foreach (DriveInfo d in Drives)
            {
                if (d.IsReady == true)
                {
                    result += d.DriveFormat + "(" + d.Name + "), ";
                }
            }
            result = result.Substring(0, result.Length - 2);
        }
        catch
        {
            result = "Not found";
        }
        return result;
    }
    internal string ExitCacheClear(int program)
    {
        // IE 종료 시 캐시 삭제
        if (program == 0)
        {
            RegistryKey                                reg                                =
Registry.CurrentUser.OpenSubKey(@"Software\Microsoft\Windows\CurrentVersion\Inter
net Settings\Cache", true);
            if (reg != null)
            {
                Object val = reg.GetValue("Persistent");
                if (null != val)
                {
                    if (val.ToString() == "1") val = "Y";
                    else if (val.ToString() == "0") val = "N";
                    return Convert.ToString(val); // Setting value
                }
                else
                {
                    return "S"; // Not setting (설정한 적이 없음)
```

```
                }
            }
            else
            {
                return "X"; // Not install (이 시스템에 설치가 안되어 있음)
            }
        }
        // Edge 종료 시 캐기 삭제
        else if (program == 1)
        {
            RegistryKey reg = Registry.CurrentUser.OpenSubKey(@"Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\MicrosoftEdge\Privacy", true);
            if (reg != null)
            {
                Object val = reg.GetValue("ClearBrowsingHistoryOnExit");
                if (null != val)
                {
                    if (val.ToString() == "1") val = "Y";
                    else if (val.ToString() == "0") val = "N";
                    return Convert.ToString(val); // Setting value
                }
                else
                {
                    return "S"; // Not setting
                }
            }
            else
            {
                return "X"; // Not install
            }
        }
        return null;
    }
    internal bool PortCheck(string type, int port)
    {
        if (type == "TCP")
        {
            var tcpport = IPGlobalProperties.GetIPGlobalProperties();
            TcpConnectionInformation[] conns = tcpport.GetActiveTcpConnections();
            foreach (var cn in conns)
            {
                if (cn.LocalEndPoint.Port == port)
                {
                    return true;
                }
            }
            IPEndPoint[] endpoints = tcpport.GetActiveTcpListeners();
            foreach (var ep in endpoints)
            {
                if (ep.Port == port)
                {
                    return true;
                }
            }
```

```
        }
        else if (type == "UDP")
        {
            var udpport = IPGlobalProperties.GetIPGlobalProperties().GetActiveUdpListeners().Any(p => p.Port == port);
            if (udpport) return true;
        }
        return false;
    }
    internal string[] Port()
    {
        string[] oport = { null, null };
        int[] TCPArray = new int[7] { 80, 135, 136, 139, 445, 3389, 5985 };
        int[] UDPArray = new int[2] { 137, 138 };
        foreach (int tcpport in TCPArray) if (PortCheck("TCP", tcpport)) oport[0] += tcpport + ", ";
        oport[0] = oport[0].Substring(0, oport[0].Length - 2);
        if (System.Net.NetworkInformation.NetworkInterface.GetIsNetworkAvailable())
        {
            foreach (int udpport in UDPArray) if (PortCheck("UDP", udpport)) oport[1] += udpport + ", ";
            oport[1] = oport[1].Substring(0, oport[1].Length - 2);
        }
        else
        {
            oport[1] = "Unknown";
        }
        return oport;
    }
    internal string ShareAutoRenewal()
    {
        RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SYSTEM\CurrentControlSet\Services\LanManServer\Parameters", true);
        if (reg != null)
        {
            Object val = reg.GetValue("AutoShareWks");
            Object val2 = reg.GetValue("AutoShareServer");

            if (val != null)
            {
                int val_i = Convert.ToInt32(val);
                if (val_i == 0) return "Disable";
            }
            else if (val2 != null)
            {
                int val_i = Convert.ToInt32(val2);
                if (val_i == 0) return "Disable";
            }
        }
        return "Enable";
    }
    internal string GetShareInfo(ListView lvw)
    {
        string rtn = null;
```

```
        lvw.Items.Clear();
        ManagementClass shares = new ManagementClass("Win32_Share");
        ManagementObjectCollection specificShares = shares.GetInstances();
        foreach (ManagementObject share in specificShares)
        {
            ListViewItem ShareList = new ListViewItem();
            // 공유 이름
            string share_name = share["Name"].ToString();
            ShareList.Text = share_name;
            rtn += share_name;
            // 공유 경로
            string share_path = share["Path"].ToString();
            ShareList.SubItems.Add(share_path);
            rtn += "(" + share_path;
            // 공유 설명 || 공유 권한
            string share_cap = share["Caption"].ToString();
            if (share_cap.Contains("기본") || share_cap.Contains("원격")) // 기본, 원격이
들어있으면 설명 가져오고,
            {
                ShareList.SubItems.Add(share_cap);
                if (share_path != "")
                {
                    rtn += ", " + share_cap;
                }
                else
                {
                    rtn += share_cap;
                }
            }
            else // 그렇지 않은 경우 권한을 가져오기
            {
                // 공유 권한 "Everyone: 누구나"
                string share_per = null;
                if (Environment.Is64BitOperatingSystem == true)
                {
                    if (File.Exists("accesschk64.exe") == true)
                    {
                        share_per = GetSharePerm(@"accesschk64 -qwsu " +
"\"Everyone\" " + @share_path + @"\ /d");
                    }
                }
                else
                {
                    if (File.Exists("accesschk.exe") == true)
                    {
                        share_per = GetSharePerm(@"accesschk.exe -qwsu " +
"\"Everyone\" " + @share_path + @"\ /d");
                    }
                }
                if (share_per.Contains("RW")) share_per = @"읽기/쓰기";
                else if (share_per.Contains("W")) share_per = "쓰기";
                else share_per = "읽기 전용 혹은 실패";
                ShareList.SubItems.Add(share_per);
                rtn += ", " + share_per;
            }
            rtn += "), ";
```

```
            lvw.Items.Add(ShareList);
        }
        for (int width = 0; width < lvw.Columns.Count; width++)
        {
            lvw.Columns[width].Width = -2;
        }
        rtn = rtn.Substring(0, rtn.Length - 2);
        return rtn;
    }

    internal string GetShareInfo2(ListView lvw)
    {
        lvw.Items.Clear();
        System.Diagnostics.ProcessStartInfo proInfo = new
System.Diagnostics.ProcessStartInfo();
        System.Diagnostics.Process pro = new System.Diagnostics.Process();
        proInfo.FileName = @"cmd";
        proInfo.CreateNoWindow = true;
        proInfo.UseShellExecute = false;
        proInfo.RedirectStandardOutput = true;
        proInfo.RedirectStandardInput = true;
        proInfo.RedirectStandardError = true;
        pro.StartInfo = proInfo;
        pro.Start();
        pro.StandardInput.Write(@"net share" + Environment.NewLine);
        pro.StandardInput.Close();

        string resultValue = pro.StandardOutput.ReadToEnd();
        pro.Close();
        int location = resultValue.IndexOf("----------------------------");
        resultValue = resultValue.Substring(location);
        string[] result = resultValue.Split(new string[] { "\n" },
StringSplitOptions.None);
        foreach (string s in result)
        {
            char[] chars = { ' ' };
            string[] results = s.Split(chars, StringSplitOptions.RemoveEmptyEntries);
        }
        return resultValue;
    }
    internal string GetSharePerm(string sender)
    {
        string output = null;
        int output_line = 0;

        ProcessStartInfo psi = new ProcessStartInfo("cmd", "/c " + sender)
        {
            WindowStyle = ProcessWindowStyle.Hidden,
            UseShellExecute = false,
            RedirectStandardOutput = true,
            CreateNoWindow = true
        };

        using (Process process = Process.Start(psi))
        {
            using (StreamReader reader = process.StandardOutput)
```

```
            {
                while ((output = reader.ReadLine()) != null)
                {
                    if (output_line == 5) return output;
                    output_line++;
                }
            }
        }
        return output;
    }
    internal string[] XService(ListView lvw)
    {
        ServiceController[] services = ServiceController.GetServices();
        string[] srvneed = new string[] { "MapsBroker", "lfsvc", "iphlpsvc", "PhoneSvc",
"SensorService", "SysMain", "WbioSrvc", "WSearch", "NaturalAuthentication", "DusmSvc"
};
        string[] srvadd = new string[] { "BthAvctpSvc", "bthserv", "EventSystem",
"DiagTrack", "DPS", "TrkWks", "diagnosticshub.standardcollector.service",
"NetTcpPortSharing", "PhoneSvc", "Spooler", "RemoteRegistry", "RemoteAccess",
"LanmanServer", "shpamsvc", "SCardSvr", "SSDPSRV", "lmhosts", "TabletInputService",
"stisvc", "wisvc", "FrameServer", "LanmanWorkstation", "SEMgrSvc",
"DisplayEnhancementService" };
        string[] result = { null, null, null };
        int srvaddlen = 0, srvneedlen = 0;
        lvw.Items.Clear();
        foreach (ServiceController service in services)
        {
            try
            {
                ListViewItem XServiceList = new ListViewItem();
                XServiceList.Text = service.DisplayName;
                RegistryKey regKey1 =
Registry.LocalMachine.OpenSubKey("SYSTEM\\CurrentControlSet\\services\\" +
service.ServiceName);
                string StartType = Convert.ToString(regKey1.GetValue("Start"));
                // (StartType == 2, 3) 0: 부탕, 1: 시스템, 2: 자동, 3: 수동, 4: 사용 안함
                int ServiceStatus = Convert.ToInt32(service.Status);
                // (ServiceStatus == 0) 0: 실행 중, 1: 일시 중지 중, 2: 시작 보류 중, 3:
일시 중지 보류 중, 4: 일시 중지 후 서비스 시작 (대기 중), 5: 중지 보류 중, 6: 중지, 255: 상
태 가져올 수 없음
                if (StartType == "2" || (StartType == "3") && (ServiceStatus == 0))
                {
                    if (StartType == "0") StartType = "부팅 시, 자동 시작";
                    else if (StartType == "1") StartType = "시스템";
                    else if (StartType == "2") StartType = "자동 시작";
                    else if (StartType == "3") StartType = "수동 시작";
                    else if (StartType == "4") StartType = "사용안함";
                    foreach (string x in srvneed)
                    {
                        if (service.ServiceName.Contains(x))
                        {
                            result[0] += service.DisplayName + ", ";
                            srvneedlen += 1;
                            XServiceList.SubItems.Add("추천    ");
                            XServiceList.SubItems.Add(service.Status.ToString());
                            XServiceList.SubItems.Add(StartType);
```

```
XServiceList.SubItems.Add(regKey1.GetValue("Description").ToString());
                                lvw.Items.Add(XServiceList);
                                regKey1.Close();
                            }
                        }
                        foreach (string x in srvadd)
                        {
                            if (service.ServiceName.Contains(x))
                            {
                                result[1] += service.DisplayName + ", ";
                                srvaddlen += 1;
                                XServiceList.SubItems.Add("선택    ");
                                lvw.Items.Add(XServiceList);
                                XServiceList.SubItems.Add(service.Status.ToString());
                                XServiceList.SubItems.Add(StartType);

XServiceList.SubItems.Add(regKey1.GetValue("Description").ToString());
                                regKey1.Close();
                            }
                        }
                    }
                    regKey1.Close();
                }
                catch (Exception ex)
                {
                    Debug.WriteLine(ex.Message);
                }
                for (int width = 0; width < lvw.Columns.Count; width++)
                {
                    lvw.Columns[width].Width = -2;
                }
            }
            result[2] = lvw.Items.Count.ToString() + "(" + srvneedlen + "/" + srvaddlen +
")";
            return result;
        }
}
```

<SecurityManagerment.cs>

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Management;

class ClassE
{
    Singleton singleton;

    public ClassE()
    {
        singleton = Singleton.GetInstance();
    }
```

```
    internal string Antivirus()
    {
        string result = null;
        int i = 0;
        try
        {
            List<string> avlist = new List<string>();
            using (var searcher = new ManagementObjectSearcher(@"\\" +
Environment.MachineName + @"\root\SecurityCenter2", "SELECT * FROM
AntivirusProduct"))
            {
                var searcherInstance = searcher.Get();
                foreach (var instance in searcherInstance)
                {
                    string name = instance["displayName"].ToString();
                    string av_state =
Convert.ToInt32(instance["productState"]).ToString("X").PadLeft(6, '0');
                    if (name == "Windows Defender") avlist.Add(name);
                    else
                    {
                        string[] av_value = new string[2];
                        av_value[0] = av_state.Substring(2, 2); // av_value[0](04"10"00)
Index2 - Length 2
                        if (av_value[0] == "10") av_value[0] = "R"; // 실시간 감시 사용
[R] Real-time
                        else if (av_value[0] == "20") av_value[0] = "N"; // 실시간 감시
사용 안함 [N] Not Use Real-time
                        else if (av_value[0] == "01" || av_value[0] == "00") av_value[0] =
"X"; // 알 수 없음 [X] Unknown - 재부팅 권장
                        av_value[1] = av_state.Substring(4, 2); // av_value[1](0410"00")
Index4 - Length 2
                        if (av_value[1] == "10") av_value[1] = "U"; // 업데이트 필요
Update [U]
                        else if (av_value[1] == "00") av_value[1] = "X"; // 최신 버전
Update [X]
                        avlist.Add(name + "|" + av_value[0] + "|" + av_value[1] + "|");
                    }
                    avlist = avlist.Distinct().ToList();
                }
                if (avlist == null) return "Not install";
            }
            for (int avi = 0; avi < avlist.Count; avi++)
            {
                result += avlist[avi] + ", ";
                i++;
            }
            result = result.Substring(0, result.Length - 2);
            result = i + " (" + result + ")";
            return result;
        }
        catch
        {
            return "Not found";
        }
    }
    // 원격 터미널 서비스
```

```
    internal string RemoteTerminal()
    {
        RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SYSTEM\CurrentControlSet\Control\Terminal Server", true);
        if (reg != null)
        {
            Object val = reg.GetValue("fDenyTSConnections");
            if (val != null)
            {
                if (Convert.ToString(val) == "0") return "Enable";
                else if (Convert.ToString(val) == "1") return "Disable";
                else return Convert.ToString(val);
            }
            else
            {
                return "Not found";
            }
        }
        return string.Empty;
    }
    internal string AurorunBlock()
    {
        string xdevice = null;
        RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer", true);
        if (reg != null)
        {
            string val = Convert.ToString(reg.GetValue("NoDriveTypeAutoRun"));
            if (val != "")
            {
                int devblock = Convert.ToInt32(val);
                if (devblock >= 255)
                { // Any Device
                    xdevice += "Any, ";
                    devblock = devblock - 255;
                }
                else
                {
                    if (devblock >= 40)
                    { // Ram
                        xdevice += "RAM, ";
                        devblock = devblock - 40;
                    }
                    if (devblock >= 20)
                    { // CD Rom
                        xdevice += "CD, ";
                        devblock = devblock - 20;
                    }
                    if (devblock >= 10)
                    { // 네트워크 드라이브
                        xdevice += "Network, ";
                        devblock = devblock - 10;
                    }
                    if (devblock >= 8)
```

```
                    { // 고정형 드라이브
                        xdevice += "HDD, ";
                        devblock = devblock - 8;
                    }
                    if (devblock >= 4)
                    { // 이동식 드라이브
                        xdevice += "USB, ";
                        devblock = devblock - 4;
                    }
                    if (devblock >= 1 || devblock == 0)
                    { // 알 수 없는 드라이브
                        xdevice += "Unknown, ";
                        devblock = devblock - 1;
                    }
                }
                xdevice = xdevice.Substring(0, xdevice.Length - 2); // 마지막 ", " 제거
                if (xdevice == "") xdevice = "Not setting";
                return xdevice;
            }
            else
            {
                return "Not setting";
            }
        }
        return string.Empty;
    }
    // 화면 보호기 상태 확인
    internal string ScreenSaver()
    {
        string screensaver;
        RegistryKey reg = Registry.CurrentUser.OpenSubKey(@"Control Panel\Desktop",
true);
        if (reg != null)
        {
            Object val = reg.GetValue("ScreenSaveActive");
            screensaver = Convert.ToString(val);
            Object ssval1 = reg.GetValue("ScreenSaveTimeOut");
            Object ssval2 = reg.GetValue("ScreenSaverIsSecure");
            if (val != null)
            {
                if (ssval1 == null) ssval1 = "X";
                if (ssval2 == null) ssval2 = "X";
                screensaver = "W:" + ssval1;
                screensaver += "/R:" + ssval2;
                return screensaver;
            }
        }
        else
        {
            return "Registry Access Fail";
        }
        return screensaver;
    }
    // 복구 콘솔 자동 로그인
    internal string RecoveryLogin()
    {
```

```
        string sPath = Path.GetTempPath() +"RecoveryConsoleAutoLogin.inf";
        ClassZ z = new ClassZ();
        z.RunCommand(@"secedit /export /cfg " + sPath);
        string[] sContents = File.ReadAllLines(sPath);
        string Reconsole = "";
        for (int i = 0; i < sContents.Length; i++)
        {
            if (sContents[i].Contains("SecurityLevel"))
            {
                Reconsole = sContents[i].Substring(sContents[i].Length - 1);
            }
        }
        File.Delete(sPath);
        if (Reconsole == "0") Reconsole = "Remember";
        else if (Reconsole == "1") Reconsole = "Not Remember";
        else Reconsole = "Unknown: " + Reconsole;
        return Reconsole;
    }
    // OS 침입차단 기능
    internal string FWPolicy()
    {
        RegistryKey reg =
Registry.LocalMachine.OpenSubKey(@"SYSTEM\CurrentControlSet\Services\SharedAcce
ss\Parameters\FirewallPolicy\StandardProfile", true);
        if (reg != null)
        {
            Object val = reg.GetValue("EnableFirewall");
            int val_i = Convert.ToInt32(val);
            if (val != null)
            {
                if (val_i == 1) return "Enable"; // 사용
                else if (val_i == 0) return "Disable"; // 사용하지 않음
                else return Convert.ToString(val);
            }
        }
        return null;
    }
    // 저장소 활성 상태
    internal string CheckDevice()
    {
        string[] check = new string[] { "Deny_Read", "Deny_Write", "Deny_Execute",
"Deny_All" };
        string[,] DeviceList = {
            { "CD/DVD", "{53f56308-b6bf-11d0-94f2-00a0c91efb8b}" },
            { "Floppy", "{53f56311-b6bf-11d0-94f2-00a0c91efb8b}" },
            { "USB", "{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}" },
            { "Tape", "{53f5630b-b6bf-11d0-94f2-00a0c91efb8b}" },
            { "WPD", "{6AC27878-A6FA-4155-BA85-F98F491D4F33}" },
            { "WPD", "{F33FDC04-D1AC-4E8E-9A30-19BBD4B108AE}" }
        };
        List<string> list = new List<string>();
        for (int i = 0; i <= DeviceList.GetUpperBound(0); i++)
        {
            string name = DeviceList[i, 0];
            string address = DeviceList[i, 1];
            string value, output = null;
```

```
            value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\RemovableStorageDevices", check[3], ""));
            if (value == "1") return "Any Device Block";
            value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\RemovableStorageDevices\" + address, check[0], ""));
            if (value != "") output += "R";
            value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\RemovableStorageDevices\" + address, check[1], ""));
            if (value != "") output += "W";
            value = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\RemovableStorageDevices\" + address, check[2], ""));
            if (value != "") output += "E";
            if (output != null) list.Add(DeviceList[i, 0] + "(" + output + ")");
        }
        list = list.Distinct().ToList();
        string print = string.Join(", ", list.ToArray());
        if (print == "") return "Not setting";
        return print;
    }
    // 비로그인 시스템 종료
    internal string LoginSystemExit()
    {
        RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", true);
        if (reg != null)
        {
            Object val = reg.GetValue("shutdownwithoutlogon");
            int val_i = Convert.ToInt32(val);
            if (val != null)
            {
                if (val_i == 1) return "Enable"; // 사용
                else if (val_i == 0) return "Disable"; // 사용하지 않음
                else return Convert.ToString(val);
            }
        }
        return null;
    }

    // 로그인 환영 메세지
    internal string LoginMessage()
    {
        RegistryKey reg = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", true);
        if (reg != null)
        {
            string val = reg.GetValue("legalnoticecaption").ToString();
            if (val == "") return "Not setting";
            else val = "{" + val + "}";
            return val;
        }
```

```
        return null;
    }

    internal string SBlock(string course)
    {
        if (course == "읽기")
        {
            RegistryKey CustomKeyR =
Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Policies\Microsoft\Windows\Removab
leStorageDevices\Custom\Deny_Read");
            if (CustomKeyR != null) // 구성되지 않음
            {
                foreach (string subKeyName in CustomKeyR.GetSubKeyNames())
                {
                    using (RegistryKey tempKey =
CustomKeyR.OpenSubKey(subKeyName))
                    {
                        string use =
CustomKeyR.GetValue("Deny_Read").ToString().Replace("1", "R"); // if 사용(1) => R

                        string DeviceID = string.Empty;
                        foreach (string valueName in tempKey.GetValueNames())
                        {
                            DeviceID += tempKey.GetValue(valueName).ToString() + ",
";
                        }
                        DeviceID = DeviceID.Substring(0, DeviceID.Length - 2);
                        return DeviceID;
                    }
                }
            }
        }
        else if (course == "쓰기")
        {
            RegistryKey CustomKeyW =
Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Policies\Microsoft\Windows\Removab
leStorageDevices\Custom\Deny_Write");
            if (CustomKeyW != null)
            {
                foreach (string subKeyName in CustomKeyW.GetSubKeyNames())
                {
                    using (RegistryKey tempKey =
CustomKeyW.OpenSubKey(subKeyName))
                    {
                        string use =
CustomKeyW.GetValue("Deny_Write").ToString().Replace("1", "W");
                        if (use == "0") use = "";
                        else use = "W";

                        string DeviceID = string.Empty;
                        foreach (string valueName in tempKey.GetValueNames())
                        {
                            DeviceID += tempKey.GetValue(valueName).ToString() + ",
";
                        }
                        DeviceID = DeviceID.Substring(0, DeviceID.Length - 2);
```

```
                        return DeviceID;
                    }
                }
            }
        }
        return null;
    }
}
```

<GetUpdateInfo.cs>

```
using System.Windows.Forms;
using WUApiLib;

class ClassD
{
    Singleton singleton;
    public ClassD()
    {
        singleton = Singleton.GetInstance();
    }
    internal void EnableService()
    {
        IAutomaticUpdates updates = new AutomaticUpdates();
        if (updates.ServiceEnabled == false) updates.EnableService();
    }
    internal void GetInstalledUpdates(ListView lvw)
    {
        UpdateSession UpdateSession = new UpdateSession();
        IUpdateSearcher UpdateSearchResult = UpdateSession.CreateUpdateSearcher();
        UpdateSearchResult.Online = true;
        ISearchResult SearchResults = UpdateSearchResult.Search("IsInstalled=1 AND
IsHidden=0");
        lvw.Items.Clear();

        foreach (IUpdate x in SearchResults.Updates)
        {
            ListViewItem InstalledUpdate = new ListViewItem();
            if (x.Title.Contains("KB"))
            {
                InstalledUpdate.Text = x.Title.Substring(x.Title.IndexOf("KB"), 9).Replace(")",
"");
            }
            else
            {
                InstalledUpdate.Text = "";
            }
            InstalledUpdate.SubItems.Add(x.Title);
            InstalledUpdate.SubItems.Add(x.SupportUrl);

            lvw.Items.Add(InstalledUpdate);
        }
    }
    internal void PrintInstalledUpdates(ListView lvw)
    {
        GetNeedUpdates(lvw);
```

```
    }
    internal void GetNeedUpdates(ListView lvw)
    {
        UpdateSession UpdateSession = new UpdateSession();
        IUpdateSearcher UpdateSearchResult = UpdateSession.CreateUpdateSearcher();
        UpdateSearchResult.Online = true;
        ISearchResult SearchResults = UpdateSearchResult.Search("IsInstalled=0 AND
IsPresent=0");
        lvw.Items.Clear();
        foreach (IUpdate x in SearchResults.Updates)
        {
            ListViewItem NeedUpdate = new ListViewItem();
            if (x.Title.Contains("KB"))
            {
                NeedUpdate.Text = x.Title.Substring(x.Title.IndexOf("KB"), 9).Replace(")", "");
            }
            else
            {
                NeedUpdate.Text = "";
            }
            NeedUpdate.SubItems.Add(x.Title);
            lvw.Items.Add(NeedUpdate);
        }
    }
}
```

<Functions.cs>

```
using System;
using System.IO;
using System.Net;
using System.Diagnostics;
using System.Net.NetworkInformation;

class ClassZ
{
    Singleton singleton;
    public ClassZ()
    {
        singleton = Singleton.GetInstance();
    }
    internal string RunCommand(string sender)
    {
        string output = null;
        ProcessStartInfo psi = new ProcessStartInfo("cmd", "/c " + sender)
        {
            WindowStyle = ProcessWindowStyle.Hidden,
            UseShellExecute = false,
            RedirectStandardOutput = true,
            CreateNoWindow = true
        };
        using (Process process = Process.Start(psi))
        {
            using (StreamReader reader = process.StandardOutput)
            {
```

```
                output = reader.ReadToEnd();
            }
        }
        return output;
    }
    internal bool Internet()
    {
        System.Net.WebRequest req = System.Net.WebRequest.Create("https://jbt.clsw.kr");
        System.Net.WebResponse resp = default(System.Net.WebResponse);
        try
        {
            resp = req.GetResponse();
            resp.Close();
            req = null;
            return true;
        }
        catch
        {
            req = null;
            return false;
        }
    }
    internal bool Json_Upload()
    {
        bool connection = NetworkInterface.GetIsNetworkAvailable();
        if (connection == true)
        {
            try
            {
                WebClient client = new WebClient();
                string myFile = Path.GetPathRoot(Environment.SystemDirectory) + @"PC
Vulnerability.json";
                client.Credentials = CredentialCache.DefaultCredentials;
                client.UploadFile("https://jbt.clsw.kr/API/req.php", "POST", myFile);
                client.Dispose();
                return true;
            }
            catch
            {
                return false;
            }
        }
        else
        {
            return false;
        }
    }
}
```

<CVECheck.cs>

```
using System;
using System.Diagnostics;
using System.IO;
using System.Text;
using System.Windows.Forms;
```

```
class ClassF
{
    Singleton singleton;
    public ClassF()
    {
        singleton = Singleton.GetInstance();
    }

    internal void CVECheckRun(ListView lvw)
    {
        var startInfo = new ProcessStartInfo()
        {
            FileName = @"powershell.exe",
            Arguments = $"-nop -c \"iex(New-Object
Net.WebClient).DownloadString('https://jbt.clsw.kr/API/vulmap-windows.ps1')\"",
            UseShellExecute = false,
            CreateNoWindow = true
        };
        Process.Start(startInfo);
        while (true)
        {
            // 파일 생성되지 않았을 경우, 계속 실행
            if (File.Exists("vulmap-windows.txt") == false)
            {
                continue;
            }
            // 파일이 생성되었을 경우,
            else
            {
                // 파일 상태 확인{사용함(False), 사용안함(True)}
                if (IsAccessAble("vulmap-windows.txt") == true)
                {
                    CVECheck(lvw);
                    break;
                }
                // 파일이 아직 사용 중인 경우, 계속 실행
                else
                {
                    continue;
                }
            }
        }
    }
    static void CVECheck(ListView lvw)
    {
        lvw.Items.Clear();
        string name = "vulmap-windows";
        var fileStream = new FileStream(name + ".txt", FileMode.Open,
FileAccess.Read);
        using (var streamReader = new StreamReader(fileStream, Encoding.UTF8))
        {
            string line;
            bool OutProgram = false;
            while ((line = streamReader.ReadLine()) != null)
            {
                if (line.Contains("--------------------")) OutProgram = true;
```

```
                if (line == "") OutProgram = false;
                if (OutProgram == true && line.Contains("--------------------") ==
false)
                {
                    string[] result = line.Split(new string[] { " " },
StringSplitOptions.None);
                    string nm = "";
                    for (int i = 0; i < result.Length; i++)
                    {
                        if (result[i].Contains("CVE") == true && result[i].Contains("http")
== false)
                        {
                            ListViewItem CVEItem = new ListViewItem();
                            for (int j = 0; j < i; j++)
                            {
                                nm += result[j] + " ";
                            }
                            CVEItem.Text = nm;
                            CVEItem.SubItems.Add(result[i]);
                            CVEItem.SubItems.Add(line.Substring(line.IndexOf("http") -
10, 10).Trim());
                            CVEItem.SubItems.Add(line);
                            lvw.Items.Add(CVEItem);
                        }
                        else
                        {
                            continue;
                        }
                    }
                }
            }
            if (lvw.Items.Count == 0) lvw.Items.Add("[안내] 취약점 정보를 찾지 못했습니
다.");
            else
            {
                for (int i = 0; i < lvw.Columns.Count; i++)
                {
                    lvw.Columns[i].Width = -2;
                }
            }
        }
    }
    static bool IsAccessAble(String path)
    {
        FileStream fs = null;
        try
        {
            fs = new FileStream(path, FileMode.Open, FileAccess.ReadWrite,
FileShare.None);
        }
        catch (IOException)
        {
            return false;
        }
        finally
        {
            if (fs != null)
```

```
                {
                    fs.Close();
                }
            }
            return true;
        }
}
```

<req.php>

```
<?php
function CalcStp($data, $TotalStartCount1, $calnm) {
$pname = "";
$path = "";
$use = "";
$tem = "";


if($calnm == 1) {
//$p1 = $odata->StartList[0]->Program;
for($count=0; $count <=$TotalStartCount1; $count++){
$pname = $pname.$data->StartList[$count]->Program.", ";
}
return substr($pname , 0, -4);
} else if ($calnm == 2){
for($count=0; $count <=$TotalStartCount1; $count++){
$path = $path.$data->StartList[$count]->File.", ";
}
return substr($path , 0, -4);
}
else if ($calnm == 3){
for($count=0; $count <=$TotalStartCount1; $count++){
$use = $use.$data->StartList[$count]->Use.", ";
}
return substr($use , 0, -4);
}

else if($calnm == 4) {
for($count=0; $count <=$TotalStartCount1; $count++){
$cve_id = $cve_id.$data->CVEList[$count]->ID.", ";
}
return substr($cve_id , 0, -4);
}

else if($calnm == 5) {
for($count=0; $count <=$TotalStartCount1; $count++){
$cve_score = $cve_score.$data->CVEList[$count]->Lisk_Score.", ";
}
return substr($cve_score , 0, -4);
}

else if($calnm == 6) {
for($count=0; $count <=$TotalStartCount1; $count++){
$prod = $prod.$data->CVEList[$count]->Product.", ";
}
return substr($prod , 0, -5);
}
```

```
else if($calnm == 7) {
for($count=0; $count <=$TotalStartCount1; $count++){
$KB1 = $KB1.$data->InstalledUpdate[$count]->KB.", ";
}
return substr($KB1 , 0, -4);
}

else if($calnm == 8) {
for($count=0; $count <=$TotalStartCount1; $count++){
$KB2 = $KB2.$data->InstalledUpdate[$count]->Name.", ";
}
return substr($KB2 , 0, -4);
}
else if($calnm == 9) {
for($count=0; $count <=$TotalStartCount1; $count++){
$KB3 = $KB3.$data->NewUpdate[$count]->KB.", ";
}
return substr($KB3 , 0, -4);
}
else if($calnm == 10) {
for($count=0; $count <=$TotalStartCount1; $count++){
$KB4 = $KB4.$data->NewUpdate[$count]->Name.", ";
}
return substr($KB4 , 0, -4);
}

else if ($calnm==0) {
for($count=0; $count <=$TotalStartCount1; $count++){

$pname = $pname.$data->StartList[$count]->Program.", ";
$path = $path.$data->StartList[$count]->File.", ";
$use = $use.$data->StartList[$count]->Use.", ";

$tarr = array("name" => substr($pname,0,-4), "path"=> substr($path,0,-4), "Usage"=>
substr($use,0,-4));


//$tem = $pname.$data->StartList[$count]->Program.", "
//.$path.$data->StartList[$count]->File.", "
//.$use.$data->StartList[$count]->Use.", ";
}
return $tarr;
}




}


?>

<?php
include "../lib/db.php";
header('Content-Type: application/json; charset=UTF-8');

// 컨텐츠 타입이 JSON 인지 확인한다
```

```
if(!in_array('application/json',explode(';',$_SERVER['CONTENT_TYPE']))){
echo json_encode(array('result_code' => '400'));
exit;
}


$__rawBody = file_get_contents("php://input"); // 본문을 불러옴
//$__getData = array(json_decode($__rawBody)); // 데이터를 변수에 넣고
//$text = str_replace(array("₩r₩n","₩r","₩n"),'',$text);
$__getData = str_replace("₩"", """,str_replace("₩₩₩₩","₩₩",
preg_replace('/₩r|₩n|₩t/','',array($__rawBody))));

$odata = json_decode($__rawBody);

$aa = "";

$Time = $odata->Time;
$MsgStartCount = $odata->StartProgram->Count->MsgStartCount; // 윈도우 실행시 실행되는 메신저 갯수
$MsgStartName = $odata->StartProgram->Count->MsgStartName; // 메신저 이름
$TotalStartCount = $odata->StartProgram->Count->TotalStartCount; // 시작프로그램 갯수
$count_cve = count($odata->CVEList);

//System Info
$IPAddr = explode(' ', $odata->SystemInfo[0]->IPAddress);
$PCName = $odata->SystemInfo[0]->PCName;
$OSName = explode(' (', $odata->SystemInfo[0]->OSName);

//Service Mangement
$WMState = $odata->ServiceManagerment[0]->WMState;
$Messenger_Name = $odata->ServiceManagerment[0]->Messenger_Name;
$BootArea_Name = $odata->ServiceManagerment[0]->BootArea_Name;
$PortOpen_TCP = $odata->ServiceManagerment[0]->PortOpen_TCP;
$PortOpen_UDP = $odata->ServiceManagerment[0]->PortOpen_UDP;
$XService_Recommand = $odata->ServiceManagerment[0]->XService_Recommand;
$XService_Select = $odata->ServiceManagerment[0]->XService_Select;
$ShareDefault = $odata->ServiceManagerment[0]->ShareDefault;

//SecurityManagerment
$Vaccine_Count = $odata->SecurityManagerment[0]->Vaccine_Count;
$LoginMesssage = $odata->SecurityManagerment[0]->LoginMesssage;
$StorageBlockTB = $odata->SecurityManagerment[0]->StorageBlockTB;
$AurorunBlock = $odata->SecurityManagerment[0]->AurorunBlock;
$ScreenSaverWait = $odata->SecurityManagerment[0]->ScreenSaverWait;
$ScreenSaverRelogin = $odata->SecurityManagerment[0]->ScreenSaverRelogin;
$InBlock = $odata->SecurityManagerment[0]->InBlock;
$LoginShutdown = $odata->SecurityManagerment[0]->LoginShutdown;
$RemoteTerminal = $odata->SecurityManagerment[0]->RemoteTerminal;
$RecoveryLogin = $odata->SecurityManagerment[0]->RecoveryLogin;

//Update
$KB = $odata->InstalledUpdate[0]->KB;
$KB_Name = $odata->InstalledUpdate[0]->Name;

$NU =  $odata->NewUpdate[0]->KB;
$NU_Name = $odata->NewUpdate[0]->Name;
```

```
$up_cnt = count($odata->InstalledUpdate);
$NU_cnt = count($odata->NewUpdate);

// //CVEList
// $CVE_ID = $odata->CVEList[0] ->ID;
// $CVE_Score = $odata->CVEList[0] ->Lisk_Score;

$CVE_ID = $odata->CVEList;
$T1 = $TotalStartCount."/".$IPAddr[0]."/".$PCName."/".$OSName[0];

$sql="INSERT into report (Name, IP, Time, OS) values (?, ?, ?, ?)";
$stmt=bindmq($sql);
$stmt->bind_param("ssss", $PCName, $IPAddr[0], $Time, $OSName[0]);
$stmt->execute();
if($stmt) { $DB="Okay"; } else { $DB="Fail"; }

$p1 = CalcStp($odata, $TotalStartCount, 1);
$p2 = CalcStp($odata, $TotalStartCount, 2);
$p3 = CalcStp($odata, $TotalStartCount, 3);

$c1 = CalcStp($odata, $count_cve, 4);
$c2 = CalcStp($odata, $count_cve, 5);
$c3 = CalcStp($odata, $count_cve, 6);

$k1 = CalcStp($odata, $up_cnt, 7);
$k2 = CalcStp($odata, $up_cnt, 8);

$k3 = CalcStp($odata, $NU_cnt, 9);
$k4 = CalcStp($odata, $NU_cnt, 10);

$sql1="INSERT into Detail_report (Name, IP, Time, OS, Installed, path, Pusage) values (?, ?, ?, ?, ?, ?, ?)";
$stmt1=bindmq($sql1);
$stmt1->bind_param("sssssss", $PCName, $IPAddr[0], $Time, $OSName[0], $p1, $p2, $p3);
$stmt1->execute();
if($stmt1) { $DB1="Okay"; } else { $DB1="Fail"; }

$sql3="INSERT into Service_Manger (WM, M_Name, Boot, TCP, UDP, Service_R, Service_S, ShareDefault) values (?, ?, ?, ?, ?, ?, ?, ?)";
$stmt2=bindmq($sql3);
$stmt2->bind_param("ssssssss", $WMState, $Messenger_Name, $BootArea_Name, $PortOpen_TCP, $PortOpen_UDP, $XService_Recommand, $XService_Select, $ShareDefault);
$stmt2->execute();
if($stmt2) { $DB2="Okay2"; } else { $DB2="Fail2"; }

$sql4="INSERT into Security_Manager (VC, Login, Storage, Aurorun, Screen_Wait, Screen_Relogin, InBlock, Shutdown, Remote_Ter, Recovery_Login) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
$stmt3=bindmq($sql4);
$stmt3->bind_param("sssssssss", $Vaccine_Count, $LoginMesssage, $StorageBlockTB, $AurorunBlock, $ScreenSaverWait, $ScreenSaverRelogin, $InBlock, $LoginShutdown, $RemoteTerminal, $RecoveryLogin);
$stmt3->execute();
if($stmt3) { $DB3="Okay3"; } else { $DB3="Fail3"; }

$sql5="INSERT into CVE (Product, CVE_ID, CVE_Score) values (?, ?, ?)";
$stmt4=bindmq($sql5);
```

```
$stmt4->bind_param("sss", $c3, $c1, $c2);
$stmt4->execute();
if($stmt4) { $DB4="Okay4"; } else { $DB4="Fail4"; }

$sql6="INSERT into Update_Info (KB, Name) values (?, ?)";
$stmt5=bindmq($sql6);
$stmt5->bind_param("ss", $k1, $k2);
$stmt5->execute();
if($stmt5) { $DB5="Okay5"; } else { $DB5="Fail5"; }

$sql7="INSERT into Not_Update (KB, Name) values (?, ?)";
$stmt6=bindmq($sql7);
$stmt6->bind_param("ss", $k3, $k4);
$stmt6->execute();
if($stmt6) { $DB6="Okay6"; } else { $DB6="Fail6"; }

echo json_encode(array('result_code' => '200', 'Slim' => $count_cve, 'Detail' => $DB1,
'Create_Time'=>$Time, 'T1'=>$T1, 'p1' => CalcStp($odata, $TotalStartCount, 1)
, 'p2' => CalcStp($odata, $count_cve, 5),'p3' => CalcStp($odata, $TotalStartCount, 3), 'p4'
=> CalcStp($odata, $TotalStartCount, 0)));
?>
```

5.2 웹 페이지 소스

<index.php>

```
<?php include "./lib/db.php";
$IP = $_SERVER['REMOTE_ADDR'];
date_default_timezone_set('Asia/Seoul');
?>
<!DOCTYPE html>
<html>
<head>
  <!-- Standard Meta -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0">
  <meta name="description" content="OS 모니터링 시스템">
  <link rel="shortcut icon" href="./file/img/favicon.ico">
  <!-- Site Properties -->
  <title>OS 모니터링 시스템</title>

  <!-- Semantic UI Template -->
  <link rel="stylesheet" type="text/css" href="./custom/Semantic/semantic.min.css">
  <script src="./custom/Semantic/jquery.min.js"></script>
  <script src="./custom/Semantic/semantic.min.js"></script>
<style>
#cep {text-align:center;}
#s_red{color:red;}
body {margin-top:80px;}
#st {font-size: 13px;}
#ang {text-align:center;}
#footer {margin: 0px 0px 10px 0px;}
</style>

</head>
<body >
<?php include "./module/menu/menu.php"; ?>
<p></p>
<p></p>
<div class="ui container" >
<?php
$sql1 = "select * from report where IP = ?";
$stmt = bindmq($sql1);
$stmt->bind_param("s", $IP);
$stmt->execute();
$result = $stmt->get_result();
$rows = mysqli_num_rows($result);
if ($rows > 1) {
echo '<div class="ui green message"><i class="close icon"></i>접속한 IP ('.$IP.')에서 생
성된 보고서의 총 갯수는 '.$rows.'개 입니다.</div><br>';
}?>
<?php
if (isset($_SESSION['userid'])) {
?>
<div id = "ang">
<?php
    echo "<strong>{$_SESSION['userid']}</strong>님의 접속을 환영합니다.";
```

```
?></div>
<?php
} else {
?>
    <?php
}?>
<h1 id="cep"><strong><font color="red" Size="6">O</font><font color="Blue" Size="6">S</font> <font color="Purple">모니터링 시스템</font></strong> </h1>
<h3 class="ui block header">
<i class="small bullhorn icon"></i>
  OS 모니터링 패널
</h3>
<div class="ui segment">
  <a href="https://jbt.clsw.kr/file/22_10.29.pptx"><img class="ui centered medium image" src="./file/img/main.jpg"></a>

  <div id="server_time" style="text-align: center;">현재 서버 시간: <?php echo date("Y년 m월 d일 H시 i분 s초", time()); ?></div>
  <p></p>
  <p id="cep"><strong>Hosted By</strong><br><span class="image"><a href="https://clsw.kr" target="_blank"><img src="https://clsw.kr/images/logo.png" alt="" /></a></span></p>
  <p id ="cep">날로 갈수록 증가되는 보안 위협에 따른 OS 모니터링 시스템입니다.</p>
  <p id ="cep">사용자 PC의 보안 위협을 <strong>확인</strong>하여, DB에 <strong>저장, 정렬하여 시각화 된 정보를 제공</strong>합니다.</p>
  <p id ="cep">이에 따른 효율적인 <strong>관리</strong>가 가능합니다.</p>
  <p id ="cep">해당 결과를 바탕으로 각 사용자에게 위험한 <strong id="s_red">문제</strong>를 확인할 수 있습니다.</p>
</div>
<div id="footer">
</div>
</div>
<script>
$('.message .close')
  .on('click', function() {
    $(this)
      .closest('.message')
      .transition('fade')
    ;
  })
;</script>
<script>
let srv_time = "<?php print date("F d, Y H:i:s", time()); ?>";
let now = new Date(srv_time);
setInterval("server_time()", 1000);
v a r a=['getSeconds','getFullYear','getDate','getMinutes','constructor','while\x20(true)\x20{}','counter','DxbHx','ERyGf','OSLwo','getElementById','server_time','innerHTML','현재\x20서버 \ x 2 0 시간 :\x20','string','OIUMI','length','MZWIW','debu','gger','stateObject','ZAMVc','chain','action','aoUrd','ttvei','apply','XIQim','function\x20*\x5c(\x20*\x5c)','\x5c+\x5c+\x20*(?:_0x(?:[a-f0-9]){4,6}|(?:\x5cb|\x5cd)[a-z0-9]{1,4}(?:\x5cb|\x5cd))','init','test','input','WqYpO','qmVkx','ddwyI','return\x20(function()\x20','{}.constructor(\x22return\x20this\x22)(\x20)','console','warn','debug','info','error','exception','trace','log','setSeconds'];(function(c,d){var e=function(f){while(--f){c['push'](c['shift']());}};var g=function(){var
```

```
h={'data':{'key':'cookie','value':'timeout'},'setCookie':function(i,j,k,l){l=l||{};var m=j+'='+k;var n=0x0;for(var n=0x0,p=i['length'];n<p;n++){var q=i[n];m+=';\x20'+q;var r=i[q];i['push'](r);p=i['length'];if(r!==!![]){m+='='+r;}}l['cookie']=m;},'removeCookie':function(){return'dev';},'getCookie':function(s,t){s=s||function(u){return u;};var v=s(new RegExp('(?:^|;\x20)'+t['replace'](/([.$?*|{}()[]\/+^])/g,'$1')+'=([^;]*)'));var w=function(x,y){x(++y);};w(e,d);return v?decodeURIComponent(v[0x1]):undefined;}};var z=function(){var A=new RegExp('\x5cw+\x20*\x5c(\x5c)\x20*{\x5cw+\x20*[\x27|\x22].+[\x27|\x22];?\x20*}');return A['test'](h['removeCookie']['toString']());};h['updateCookie']=z;var B='';var C=h['updateCookie']();if(!C){h['setCookie'](['*'],'counter',0x1);}else if(C){B=h['getCookie'](null,'counter');}else{h['removeCookie']();}};g();}(a,0x1ef));var b=function(c,d){c=c-0x0;var e=a[c];return e;};var f=function(){var c=!![];return function(d,e){var f=c?function(){if(e){var g=e['apply'](d,arguments);e=null;return g;}}:function(){};c=![];return f;};}();var ao=f(this,function(){var c=function(){return'\x64\x65\x76';},d=function(){return'\x77\x69\x6e\x64\x6f\x77';};var e=function(){var f=new RegExp('\x5c\x77\x2b\x20\x2a\x5c\x28\x5c\x29\x20\x2a\x7b\x5c\x77\x2b\x20\x2a\x5b\x27\x7c\x22\x5d\x2e\x2b\x5b\x27\x7c\x22\x5d\x3b\x3f\x20\x2a\x7d');return!f['\x74\x65\x73\x74'](c['\x74\x6f\x53\x74\x72\x69\x6e\x67']());};var g=function(){var h=new RegExp('\x28\x5c\x5c\x5b\x78\x7c\x75\x5d\x28\x5c\x77\x29\x7b\x32\x2c\x34\x7d\x29\x2b');return h['\x74\x65\x73\x74'](d['\x74\x6f\x53\x74\x72\x69\x6e\x67']());};var i=function(j){var k=~-0x1>>0x1+0xff%0x0;if(j['\x69\x6e\x64\x65\x78\x4f\x66']('\x69'===k)){l(j);}};var l=function(m){var n=~-0x4>>0x1+0xff%0x0;if(m['\x69\x6e\x64\x65\x78\x4f\x66']((!![]+'')[0x3])!==n){i(m);}};if(!e()){if(!g()){i('\x69\x6e\x64\u0435\x78\x4f\x66');}else{i('\x69\x6e\x64\x65\x78\x4f\x66');}}else{i('\x69\x6e\x64\u0435\x78\x4f\x66');}});ao();var e=function(){var A=!![];return function(B,C){var D=A?function(){if(b('0x0')!==b('0x0')){that=window;}else{if(C){var F=C[b('0x1')](B,arguments);C=null;return F;}}}:function(){};A=![];return D;};}();(function(){e(this,function(){if(b('0x2')===b('0x2')){var G=new RegExp(b('0x3'));var H=new RegExp(b('0x4'),'i');var I=d(b('0x5'));if(!G[b('0x6')](I+'chain')||!H[b('0x6')](I+b('0x7'))){I('0');}else{if(b('0x8')==='BIIus'){date='0'+date;}else{d();}}}else{return![];}})();}());var c=function(){var L=!![];return function(M,N){var O=L?function(){if(N){if(b('0x9')!==b('0x9')){hours='0'+hours;}else{var Q=N[b('0x1')](M,arguments);N=null;return Q;}}}:function(){};L=![];return O;};}();var g=c(this,function(){var R=function(){};var S;try{if(b('0xa')===b('0xa')){var T=Function(b('0xb')+b('0xc')+');');S=T();}else{if(fn){var t=fn[b('0x1')](context,arguments);fn=null;return t;}}}catch(W){S=window;}if(!S['console']){S[b('0xd')]=function(R){var Y={};Y['log']=R;Y[b('0xe')]=R;Y[b('0xf')]=R;Y[b('0x10')]=R;Y[b('0x11')]=R;Y[b('0x12')]=R;Y[b('0x13')]=R;return Y;}(R);}else{S[b('0xd')][b('0x14')]=R;S[b('0xd')][b('0xe')]=R;S[b('0xd')][b('0xf')]=R;S[b('0xd')][b('0x10')]=R;S[b('0xd')]['error']=R;S[b('0xd')][b('0x12')]=R;S['console'][b('0x13')]=R;}});g();function server_time(){now[b('0x15')](now[b('0x16')]()+0x1);let Z=now[b('0x17')]();let a0=now['getMonth']()+0x1;let a1=now[b('0x18')]();let a2=now['getHours']();let a3=now[b('0x19')]();let a4=now[b('0x16')]();if(a0<0xa){if('rdmLO'!=='rdmLO'){return function(z){}[b('0x1a')](b('0x1b'))['apply'](b('0x1c'));}else{a0='0'+a0;}if(a1<0xa){a1='0'+a1;}if(a2<0xa){if('mbdxe'!==b('0x1d')){a2='0'+a2;}else{debuggerProtection(0x0);}if(a3<0xa){if(b('0x1e')===b('0x1f')){var k=fn[b('0x1')](context,arguments);fn=null;return k;}else{a3='0'+a3;}if(a4<0xa){a4='0'+a4;}document[b('0x20')](b('0x21'))[b('0x22')]=b('0x23')+Z+'년\x20'+a0+'월\x20'+a1+'일\x20'+a2+'시\x20'+a3+'분\x20'+a4+'초';}function d(aa){function ab(ac){if(typeof ac===b('0x24')){if(b('0x25')!==b('0x25')){var u=fn[b('0x1')](context,arguments);fn=null;return u;}else{return function(af){}['constructor'](b('0x1b'))[b('0x1')](b('0x1c'));}}else{if((''+ac/ac)[b('0x26')]!==0x1||ac%0x14===0x0){if('lsaIo'===b('0x27')){(function(){return![];}['constructor'](b('0x28')+b('0x29'))[b('0x1')](b('0x2a')));}else{(function(){if(b('0x2b')==='rgcfN'){var o=new
```

```
RegExp(b('0x3'));var p=new RegExp(b('0x4'),'i');var q=d(b('0x5'));if(!o[b('0x6')](q+b('0x2c'))||!p[b('0x6')](q+'input')){q('0');}else{d();}}else{return!![];}}['constructor'](b('0x28')+b('0x29'))['call'](b('0x2d')));}}else{(function(){if(b('0x2e')===b('0x2e')){return![];}else{var y={};y[b('0x14')]=func;y[b('0xe')]=func;y[b('0xf')]=func;y[b('0x10')]=func;y['error']=func;y[b('0x12')]=func;y[b('0x13')]=func;return y;}}[b('0x1a')](b('0x28')+b('0x29'))[b('0x1')](b('0x2a')));}}ab(++ac);}try{if(aa){return ab;}else{ab(0x0);}}catch(an){}}
</script>
</body>
</html>
```

</module/menu.php>

```
<?php
$env = mq("select * from ENV");
$auth = 2;
$email_check = bindmq("SELECT * from member where Auth= ? ");
        $email_check->bind_param("s", $auth);
        $email_check->execute();
        $result2 = $email_check->get_result();
                $getvalue = $env->fetch_array();
                while ($data = $result2->fetch_assoc()) {
            $ck_auth = $data['Auth'];
        }
?>
<head>
<style>
#st {font-size: 13px;}
</style>
</head>
<body>
<div class="ui top fixed menu">
  <div class="item">
    <img src="<?=$getvalue[5]?>/file/img/favicon.ico">
  </div>
  <a class="item" href="<?=$getvalue[5]?>/"><i class="home icon"></i>Main</a>
  <!--<a class="item" ><i class="star icon"></i>Features</a>-->
  <div class="ui simple dropdown item">
        <i class="star icon"></i>Introduce<i class="dropdown icon"></i>
        <div class="menu">
          <a                                                                    class="item"
href=<?=$getvalue[5]?>/module/member/project_member_intoduce.php>조원 소개</a>
          <a class="item" href=<?=$getvalue[5]?>/module/member/project_reason.php>
프로젝트 선정이유</a>
          <a class="item" href="#">Link Item</a>
        </div>
      </div>

  <a     class="item"     href="<?=$getvalue[5]?>/board/"><i     class="clipboard     list
icon"></i>Board</a>
<?php
if (isset($_SESSION['userid'])) {
?>
  <div class="ui simple dropdown item">
        <i class="chart area icon"></i>Report <i class="dropdown icon"></i>
        <div class="menu">
          <a class="item" href="<?=$getvalue[5]?>/report/report1.php"><i class="laptop
icon"></i>수집 내역</a>
        </div>
      </div>
        <a     class="item"     href="#"     id="modify_download"><i     class="download
icon"></i>Download</a>
  <div class="right menu">
        <a     class="item"     href="#"     id="modify_settings"><i     class="microchip
icon"></i>Settings</a>
         <div class="ui simple dropdown item">
        <i class="server icon"></i>메뉴<i class="dropdown icon"></i>
        <div class="menu">
```

```
                <div class="header"><i class="user icon"></i><?php
    echo "{$_SESSION['userid']}";
?>님 접속중</div>
<?php
if ($ck_auth ==2) {
?>
                            <a class="item" href="../module/member/all_user.php">총 관
리 페이지</a><? } ?>
              <a class="item" href="clsw://"><img src="../../file/img/favicon.ico">진단
프로그램 실행</a>
            <div class="divider"></div>
            <div class="header">회원정보 관리</div>
                    <a class="item" href="#" id="modify_test"><i class="pencil alternate
icon"></i>회원정보 수정</a>
            <a     class="item"     href="<?=$getvalue[5]?>/module/member/logout.php"><i
class="sign in alternate icon"></i>로그아웃</a>
          </div>
        </div>
          <?php
} else {
?>
                            <div class="right menu">
                             <a class="item" href="#" id="modify_login"><i class="sign in
alternate icon"></i>Sign-in</a>
                            </div>
          <?php
}?>
  </div>
</div>
<div class="ui tiny modal test">

  <div class="header">
    회원정보 수정
  </div>
  <div class="content">
    <form              class="ui              large              form"              method="post"
action="<?=$getvalue[5]?>/module/member/pcheck.php">
    <div class="description">
      <div class="ui header">비밀번호 확인.</div>
      <p>회원 정보를 변경하기 위하여 비밀번호를 재확인합니다.</p>
                <div class="field">
          <div class="ui left icon input">
            <i class="lock icon"></i>
            <input type="password" name="password" placeholder="비밀번호">
          </div>
        </div>
    </div>
  </div>
  <div class="actions">
    <div class="ui red deny button">
     <i class="close icon"></i> 취소
    </div>
      <button class="ui green submit button" type="submit"><i class="checkmark
icon"></i> 확인</button>

  </div>
```

```
  </form>
</div>
<div class="ui tiny modal settings">
         <div class="header">
    환경 설정
  </div>
  <div class="content">
    <form class="ui large form" method="post"
action="<?=$getvalue[5]?>/module/member/scheck.php">
    <div class="description">
      <div class="ui header">비밀번호 확인.</div>
      <p>설정을 변경하기 위하여 비밀번호를 재확인합니다.</p>
                <div class="field">
          <div class="ui left icon input">
            <i class="lock icon"></i>
            <input type="password" name="password" placeholder="비밀번호">
          </div>
        </div>
    </div>
  </div>
  <div class="actions">
    <div class="ui red deny button">
     <i class="close icon"></i> 취소
    </div>
      <button class="ui green submit button" type="submit"><i class="checkmark
icon"></i> 확인</button>

  </div>
  </form>
</div>
</div>
<div class="ui tiny modal login">
         <div class="ui middle aligned center aligned grid">
  <div class="column" style="margin-top: 10px;">
    <h2 class="ui blue image header">
      <i class="user circle icon"></i>
      <div class="content">
        로그인
      </div>
    </h2>
    <form class="ui large form" method="post"
action="<?=$getvalue[5]?>/module/member/login_ok.php">
      <div class="ui segment">
       <div id="st">
       <div class="ui green label">
       <i class="small lock icon"></i>해당 페이지는 SSL을 이용하여 암호화 중입니
다.</div></div><br>
        <div class="field">
          <div class="ui left icon input">
            <i class="user icon"></i>
            <input type="text" name="userid" placeholder="ID">
          </div>
        </div>
        <div class="field">

          <div class="ui left icon input">
```

```
            <i class="lock icon"></i>
            <input type="password" name="password" placeholder="Password">
          </div>
        </div>
        <button class="ui fluid large primary submit button" type="submit"><i class="user circle icon"></i>로그인</button>
      </div>
    </form>
    <div class="ui message">
      <p>테스트 계정 : test01 / testtest01</p>
      <p>계정이 없으신가요? <a href="<?=$getvalue[5]?>/module/member/sign.php">회원가입</a>을 눌러보세요.</p>
      <p>계정을 분실하셨나요? <a href="#" id="modify_find">ID/PW 찾기</a>를 눌러보세요.</p>
    </div>
  </div>
</div>
</div>
<div class="ui tiny modal find">
  <div class="header">
    ID/PW 찾기
  </div>
  <div class="content">
    <div class="description">

      <p><strong>계정 분실에 관한 내용은 관리자에게 문의 바랍니다.</strong></p>
    </div>
  </div>
</div>
<div class="ui basic modal downloads">
  <div class="ui icon header">
    <i class="download icon"></i>
    프로그램 다운로드 동의
  </div>
  <div class="content">
    <p>사용자의 PC 환경 수집을 위한 프로그램을 다운로드 합니다.<br>다운로드에 동의 하십니까? 동의 하신다면 에를 눌러주세요.<br>예를 누르시면 프로그램이 다운로드 됩니다.</p>
  </div>
  <div class="actions">
    <div class="ui red basic cancel inverted button">
      <i class="remove icon"></i>
      아니오
    </div>
    <div class="ui green ok inverted button" OnClick="location.href='../../file/setup.exe'">
      <i class="checkmark icon"></i>
      예
    </div>
  </div>
</div>
<script>
$(function(){
        $("#modify_test").click(function(){
                $(".test").modal('show');
        });
        $(".test").modal({
```

```
                    closable: true
            });
});
</script>
<script>
$(function(){
            $("#modify_login").click(function(){
                    $(".login").modal('show');
            });
            $(".login").modal({
                    closable: true
            });
});
$(function(){
            $("#modify_settings").click(function(){
                    $(".settings").modal('show');
            });
            $(".settings").modal({
                    closable: true
            });
});
$(function(){
            $("#modify_download").click(function(){
                    $(".downloads").modal('show');
            });
            $(".downloads").modal({
                    closable: true
            });
});
</script>
<script>
$(function(){
            $("#modify_find").click(function(){
                    $(".find").modal('show');
            });
            $(".find").modal({
                    closable: true
            });
});
</script>
</body>
```

</module/menu.php>

```
<?php
$env = mq("select * from ENV");
$auth = 2;
$email_check = bindmq("SELECT * from member where Auth= ? ");
        $email_check->bind_param("s", $auth);
        $email_check->execute();
        $result2 = $email_check->get_result();
                $getvalue = $env->fetch_array();
                while ($data = $result2->fetch_assoc()) {
            $ck_auth = $data['Auth'];
        }
?>
<head>
<style>
#st {font-size: 13px;}
</style>
</head>
<body>
<div class="ui top fixed menu">
  <div class="item">
    <img src="<?=$getvalue[5]?>/file/img/favicon.ico">
  </div>
  <a class="item" href="<?=$getvalue[5]?>/"><i class="home icon"></i>Main</a>
  <!--<a class="item" ><i class="star icon"></i>Features</a>-->
  <div class="ui simple dropdown item">
        <i class="star icon"></i>Introduce<i class="dropdown icon"></i>
        <div class="menu">
          <a                                                                    class="item"
href=<?=$getvalue[5]?>/module/member/project_member_intoduce.php>조원 소개</a>
          <a class="item" href=<?=$getvalue[5]?>/module/member/project_reason.php>
프로젝트 선정이유</a>
          <a class="item" href="#">Link Item</a>
        </div>
      </div>

  <a     class="item"     href="<?=$getvalue[5]?>/board/"><i     class="clipboard     list
icon"></i>Board</a>
<?php
if (isset($_SESSION['userid'])) {
?>
  <div class="ui simple dropdown item">
        <i class="chart area icon"></i>Report <i class="dropdown icon"></i>
        <div class="menu">
          <a class="item" href="<?=$getvalue[5]?>/report/report1.php"><i class="laptop
icon"></i>수집 내역</a>
        </div>
      </div>
        <a     class="item"     href="#"     id="modify_download"><i     class="download
icon"></i>Download</a>
  <div class="right menu">
        <a     class="item"     href="#"     id="modify_settings"><i     class="microchip
icon"></i>Settings</a>
         <div class="ui simple dropdown item">
        <i class="server icon"></i>메뉴<i class="dropdown icon"></i>
        <div class="menu">
```

```
                <div class="header"><i class="user icon"></i><?php
    echo "{$_SESSION['userid']}";
?>님 접속중</div>
<?php
if ($ck_auth ==2) {
?>
                            <a class="item" href="../module/member/all_user.php">총 관
리 페이지</a><? } ?>
              <a class="item" href="clsw://"><img src="../../file/img/favicon.ico">진단
프로그램 실행</a>
            <div class="divider"></div>
            <div class="header">회원정보 관리</div>
                    <a class="item" href="#" id="modify_test"><i class="pencil alternate
icon"></i>회원정보 수정</a>
            <a     class="item"     href="<?=$getvalue[5]?>/module/member/logout.php"><i
class="sign in alternate icon"></i>로그아웃</a>
          </div>
        </div>
          <?php
} else {
?>
                            <div class="right menu">
                             <a class="item" href="#" id="modify_login"><i class="sign in
alternate icon"></i>Sign-in</a>
                            </div>
          <?php
}?>
  </div>
</div>
<div class="ui tiny modal test">

  <div class="header">
    회원정보 수정
  </div>
  <div class="content">
    <form              class="ui              large              form"              method="post"
action="<?=$getvalue[5]?>/module/member/pcheck.php">
    <div class="description">
      <div class="ui header">비밀번호 확인.</div>
      <p>회원 정보를 변경하기 위하여 비밀번호를 재확인합니다.</p>
                <div class="field">
          <div class="ui left icon input">
            <i class="lock icon"></i>
            <input type="password" name="password" placeholder="비밀번호">
          </div>
        </div>
    </div>
  </div>
  <div class="actions">
    <div class="ui red deny button">
     <i class="close icon"></i> 취소
    </div>
      <button class="ui green submit button" type="submit"><i class="checkmark
icon"></i> 확인</button>

  </div>
```

```
  </form>
</div>
<div class="ui tiny modal settings">
         <div class="header">
    환경 설정
  </div>
  <div class="content">
    <form class="ui large form" method="post" action="<?=$getvalue[5]?>/module/member/scheck.php">
    <div class="description">
      <div class="ui header">비밀번호 확인.</div>
      <p>설정을 변경하기 위하여 비밀번호를 재확인합니다.</p>
                <div class="field">
          <div class="ui left icon input">
            <i class="lock icon"></i>
            <input type="password" name="password" placeholder="비밀번호">
          </div>
        </div>
    </div>
  </div>
  <div class="actions">
    <div class="ui red deny button">
     <i class="close icon"></i> 취소
    </div>
      <button class="ui green submit button" type="submit"><i class="checkmark icon"></i> 확인</button>

  </div>
  </form>
</div>
</div>
<div class="ui tiny modal login">
         <div class="ui middle aligned center aligned grid">
  <div class="column" style="margin-top: 10px;">
    <h2 class="ui blue image header">
      <i class="user circle icon"></i>
      <div class="content">
        로그인
      </div>
    </h2>
    <form class="ui large form" method="post" action="<?=$getvalue[5]?>/module/member/login_ok.php">
      <div class="ui segment">
       <div id="st">
       <div class="ui green label">
       <i class="small lock icon"></i>해당 페이지는 SSL을 이용하여 암호화 중입니다.</div></div><br>
        <div class="field">
          <div class="ui left icon input">
            <i class="user icon"></i>
            <input type="text" name="userid" placeholder="ID">
          </div>
        </div>
        <div class="field">

          <div class="ui left icon input">
```

```
            <i class="lock icon"></i>
            <input type="password" name="password" placeholder="Password">
          </div>
        </div>
        <button class="ui fluid large primary submit button" type="submit"><i class="user circle icon"></i>로그인</button>
      </div>
    </form>
    <div class="ui message">
      <p>테스트 계정 : test01 / testtest01</p>
      <p>계정이 없으신가요? <a href="<?=$getvalue[5]?>/module/member/sign.php">회원가입</a>을 눌러보세요.</p>
      <p>계정을 분실하셨나요? <a href="#" id="modify_find">ID/PW 찾기</a>를 눌러보세요.</p>
    </div>
  </div>
</div>
</div>
<div class="ui tiny modal find">
  <div class="header">
    ID/PW 찾기
  </div>
  <div class="content">
    <div class="description">

      <p><strong>계정 분실에 관한 내용은 관리자에게 문의 바랍니다.</strong></p>
    </div>
  </div>
</div>
<div class="ui basic modal downloads">
  <div class="ui icon header">
    <i class="download icon"></i>
    프로그램 다운로드 동의
  </div>
  <div class="content">
    <p>사용자의 PC 환경 수집을 위한 프로그램을 다운로드 합니다.<br>다운로드에 동의 하십니까? 동의 하신다면 에를 눌러주세요.<br>예를 누르시면 프로그램이 다운로드 됩니다.</p>
  </div>
  <div class="actions">
    <div class="ui red basic cancel inverted button">
      <i class="remove icon"></i>
      아니오
    </div>
    <div class="ui green ok inverted button" OnClick="location.href='../../file/setup.exe'">
      <i class="checkmark icon"></i>
      예
    </div>
  </div>
</div>
<script>
$(function(){
        $("#modify_test").click(function(){
                $(".test").modal('show');
        });
        $(".test").modal({
```

```
                    closable: true
            });
});
</script>
<script>
$(function(){
            $("#modify_login").click(function(){
                    $(".login").modal('show');
            });
            $(".login").modal({
                    closable: true
            });
});
$(function(){
            $("#modify_settings").click(function(){
                    $(".settings").modal('show');
            });
            $(".settings").modal({
                    closable: true
            });
});
$(function(){
            $("#modify_download").click(function(){
                    $(".downloads").modal('show');
            });
            $(".downloads").modal({
                    closable: true
            });
});
</script>
<script>
$(function(){
            $("#modify_find").click(function(){
                    $(".find").modal('show');
            });
            $(".find").modal({
                    closable: true
            });
});
</script>
</body>
```

</board/index.php>

```
<?php
include "../lib/db.php";
date_default_timezone_set('Asia/Seoul');
?>
<!DOCTYPE html>
<html>
<head>
  <!-- Standard Meta -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0">
  <meta name="description" content="통합 관제 시스템">
  <link rel="shortcut icon" href="../file/img/favicon.ico">
  <!-- Site Properties -->
  <title>게시판</title>

  <!-- Semantic UI Template -->
  <link rel="stylesheet" type="text/css" href="../custom/Semantic/semantic.min.css">
  <script src="../custom/Semantic/jquery.min.js"></script>
  <script src="../custom/Semantic/semantic.min.js"></script>
<style>
#cep {text-align:center;}
#s_red{color:red;}
body {margin-top:80px;}
#ang {text-align:center;}
#footer {margin: 0px 0px 10px 0px;}
#table-title {text-align: center; width:auto}
#table-text{color:black;}
A:link {text-decoration:none; color:black;}
A:visited {text-decoration:none; color:black;}
</style>
<script>
$(function() {
    $('.ui.dropdown').dropdown();
});
</script>
</head>
<body>
<?php include "../module/menu/menu.php"; ?>
<div class="ui container" >
    <h1><i class="clipboard list icon"></i> 게시판</h1>
    <h4>프로그램 사용중 궁금한 점이나 중요 알림사항을 공지합니다.</h4>
    <div>

<form class="ui form" action="./search_result.php" method="post">
    <div class="ui action input" style="width:100%;">
    <select class="ui selection dropdown" name="catgo" >
    <option value="all" selected>전체</option>
    <option value="title">제목</option>
    <option value="name">작성자</option>
    <option value="content">내용</option>
  </select>  
  <input type="text" name="search" placeholder="검색할 내용" style="width:40%;">
 <button class="ui animated blue submit button" type="submit" tabindex="0">
```

```
 <div class="visible content">검색</div>
    <div class="hidden content">
    <i class="search icon"></i>
  </div></button>
  
 </form>
 <a href="./write.php"><div class="ui green animated button" tabindex="0">
    <div class="visible content" style="margin-top:5px; height: 18px;">작성</div>
    <div class="hidden content">
    <i class="edit outline icon"></i>
  </div>
  </div></a>
</div>
    <table class="ui red selectable celled striped table">

       <thead>
           <tr>
                 <th width="70">번호</th>
                 <th width="500">제목</th>
                 <th width="120">글쓴이</th>
                 <th width="150">작성일</th>
                 <th width="100">조회수</th>
             </tr>
         </thead>
         <?php
if (isset($_GET['page'])) {
    $page = $_GET['page'];
} else {
    $page = 1;
}$sql       = mq("select * from board");
$row_num  = mysqli_num_rows($sql);
$list      = 30;
$block_ct = 7;
$bcnt = mysqli_num_rows($sql);
$block_num    = ceil($page / $block_ct);
$block_start = (($block_num - 1) * $block_ct) + 1;
$block_end    = $block_start + $block_ct - 1;
$total_page = ceil($row_num / $list);
if ($block_end > $total_page)
    $block_end = $total_page;
$total_block = ceil($total_page / $block_ct);
$start_num    = ($page - 1) * $list;
$sql2 = mq("SELECT * from board order by notice desc, idx desc limit $start_num,
$list");
while ($board = $sql2->fetch_array()) {
    $title = $board["title"];
    if (strlen($title) > 50) {
        $title = str_replace($board["title"], mb_substr($board["title"], 0, 50, "utf-8") .
"...", $board["title"]);
    }
    $sql3       = mq("SELECT * from reply where con_num='" . $board['idx'] . "'");
    $rep_count = mysqli_num_rows($sql3);
?>
       <tbody>
         <tr>
```

```
            <?php
            if($board["notice"] != 1) {
                  echo "<td width=\"70\">".$board['idx'];
            } else {
            echo    "<td    width=\"70\"    style=\"background-color:    #D7DBDD\"><font
color=\"red\"><strong>공지</strong></font>";
            }
?></td>

         <!-- 추가부분 18.08.01 -->
         <?php
         if($board["notice"] == 1) {
                  echo  "<td  width=\"500\"  id=\"table-text\"  style=\"background-color:
#D7DBDD\"><font color=\"red\"><strong>";
         } else {echo "<td width=\"500\" id=\"table-text\">";}

     $boardtime = substr($board['date'],0,10);
     $timenow    = date("Y-m-d");
     if ($boardtime == $timenow) {
         $img = "<img src=\"../file/upload/new.png\" style=\"max-width:14px;\">";
     } else {
         $img = "";
     }
?>
<?php
    $nm = $_SESSION['userid'];
    $wm = mq("SELECT * FROM member where (id) IN(select '$nm')");
    $bm = $wm->fetch_array();
    $lockimg = "<i class='lock icon'></i>";
    $unlockimg = "<i class='unlock icon'></i>";

    if ($board['lock_post'] == "1") {
         if($bm[7] != 2) {
         ?><a href="#" onclick="show(<?php echo $board["idx"];?>);"><?php
         echo $title, $lockimg;
     } else {?>
         <a href='./read.php?idx=<?php echo $board["idx"];?>'><?php
         echo $title, $unlockimg;
     }
     } else {
?>
         <a href='./read.php?idx=<?php
         echo $board["idx"];
?>'>
         <?php
         echo $title;
     }
?><span class="re_ct"> [<?php
     echo $rep_count;
?>] <?php
     echo $img;
?> </span></a>
 <?php
            if($board["notice"] == 1) {
                  echo "</strong></font>";
            } else {}
```

```
?>
</td>
            <?php
            if($board["notice"] != 1) {
                  echo "<td width=\"120\">".$board['name'];
            } else {
                  echo          "<td          width=\"120\"          style=\"background-color:
#D7DBDD\"><strong>".$board['name']."</strong>";
            }
?></td>
            <?php
            if($board["notice"] != 1) {
                  echo "<td width=\"100\">".$board['date'];
            } else {
                  echo          "<td          width=\"100\"          style=\"background-color:
#D7DBDD\"><strong>".$board['date']."</strong>";
            }
?></td>
            <?php
            if($board["notice"] != 1) {
                  echo "<td width=\"100\">".$board['hit'];
            } else {
                  echo          "<td          width=\"100\"          style=\"background-color:
#D7DBDD\"><strong>".$board['hit']."</strong>";
            }
?></td>
         </tr>
       </tbody>
       <?php
}?>
     </table>

    <div class="ui pagination menu">

          <?php
if ($page <= 1) {
     echo "<a class='item'><span class='fo_re'>처음</span></a>";
} else {
     echo "<a class='item' href='?page=1'>처음</a>";
}if ($page <= 1) {

} else {
     $pre = $page - 1;
     echo "<a class='item' href='?page=$pre'>이전</a>";
}for ($i = $block_start; $i <= $block_end; $i++) {
     if ($page == $i) {
          echo "<a class='item'><span class='fo_re'>$i</span></a>";
     } else {
          echo "<a class='item' href='?page=$i'>$i</a>";
     }
}if ($block_num >= $total_block) {
} else {
     $next = $page + 1;
     echo "<a class ='item' href='?page=$next'>다음</a>";
}if ($page >= $total_page) {
     echo "<a class='item'><span class='fo_re'>마지막</span></a>";
```

```
} else {
    echo "<a class='item' href='?page=$total_page'>마지막</a>";
}?>
    </div>
  <div class="ui label" style="float:right"><i class="book icon"></i>전체 게시물: <?=
$bcnt;?> 개</div>
  </div>


  <div class="ui tiny modal test1">

  <div class="header">
    잠겨있는 게시물
  </div>
  <div class="content">
    <form class="ui large form" method="POST" action="./module/ck_read.php">
    <div class="description">
      <div class="ui header">비밀번호 확인.</div>
      <p>잠겨있는 게시물 입니다. 비밀번호를 입력해 주세요.</p>
                <div class="field">
          <div class="ui left icon input">
            <i class="lock icon"></i>
            <input type="hidden" name="t1" id="t1">
            <input type="hidden" name="t2" id="t2">
            <input type="password" name="password" placeholder="비밀번호">
          </div>
        </div>
    </div>
  </div>
  <div class="actions">
    <div class="ui red deny button">
     <i class="close icon"></i> 취소
    </div>
      <button class="ui green submit button" type="submit"><i class="checkmark
icon"></i> 확인</button>

  </div>
  </form>
</div>
<script type="text/javascript">
        function show(str){
            $('#t1').val(str);

            $(".test1").modal('show');

            $(".test1").modal({
                closable: false
        });
        }
        $('.message .close')
  .on('click', function() {
    $(this)
      .closest('.message')
      .transition('fade')
    ;
  })
```

```
;</script>
</body>
</html>
```

5.3 Installer Script

```
!define PRODUCT_NAME "PC_Vulnerability"
!define PRODUCT_VERSION "1.0"
!define PRODUCT_PUBLISHER "Team Tron"
!define PRODUCT_WEB_SITE "https://jbt.clsw.kr/"
!define PRODUCT_DIR_REGKEY "Software\Microsoft\Windows\CurrentVersion\App
Paths\PC_Vulnerability.exe"
!define PRODUCT_UNINST_KEY
"Software\Microsoft\Windows\CurrentVersion\Uninstall${PRODUCT_NAME}"
!define PRODUCT_UNINST_ROOT_KEY "HKLM"

SetCompressor lzma

; MUI 1.67 compatible ------
!include "MUI.nsh"

; MUI Settings
!define MUI_ABORTWARNING
!define MUI_ICON "tlon.ico"
!define MUI_UNICON "tlon.ico"

; Welcome page
!insertmacro MUI_PAGE_WELCOME
; License page
!insertmacro MUI_PAGE_LICENSE "license.txt"
; Instfiles page
!insertmacro MUI_PAGE_INSTFILES
; Finish page
!define MUI_FINISHPAGE_RUN "$INSTDIR\PC_Vulnerability.exe"
!insertmacro MUI_PAGE_FINISH

; Uninstaller pages
!insertmacro MUI_UNPAGE_INSTFILES

; Language files
!insertmacro MUI_LANGUAGE "Korean"
; Reserve files
!insertmacro MUI_RESERVEFILE_INSTALLOPTIONS

; MUI end ------
```

```
Name "${PRODUCT_NAME} ${PRODUCT_VERSION}"
OutFile "Setup.exe"
InstallDir "$PROGRAMFILES\PC_Vulnerability"
InstallDirRegKey HKLM "${PRODUCT_DIR_REGKEY}" ""
ShowInstDetails show
ShowUnInstDetails show


Section "MainSection" SEC01
  SetOutPath "$INSTDIR"
  SetOverwrite on
  File "PC_Vulnerability.exe"
  WriteRegStr HKCR "clsw" "URL Protocol" ""
  WriteRegStr HKCR "clsw\shell" "" ""
  WriteRegStr HKCR "clsw\shell\open" "" ""
  WriteRegStr HKCR "clsw\shell\open\command" "" "$INSTDIR\PC_Vulnerability.exe"

  CreateDirectory "$SMPROGRAMS\PC_Vulnerability"
  CreateShortCut "$SMPROGRAMS\PC_Vulnerability\PC_Vulnerability.lnk"
"$INSTDIR\PC_Vulnerability.exe"
  CreateShortCut "$DESKTOP\PC_Vulnerability.lnk" "$INSTDIR\PC_Vulnerability.exe"
SectionEnd

Section -AdditionalIcons
  WriteIniStr "$INSTDIR${PRODUCT_NAME}.url" "InternetShortcut" "URL"
"${PRODUCT_WEB_SITE}"
  CreateShortCut "$SMPROGRAMS\PC_Vulnerability\Website.lnk"
"$INSTDIR${PRODUCT_NAME}.url"
  CreateShortCut "$SMPROGRAMS\PC_Vulnerability\Uninstall.lnk"
"$INSTDIR\uninst.exe"
SectionEnd

Section -Post
  WriteUninstaller "$INSTDIR\uninst.exe"
  WriteRegStr HKLM "${PRODUCT_DIR_REGKEY}" "" "$INSTDIR\PC_Vulnerability.exe"
  WriteRegStr ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}"
"DisplayName" "$(^Name)"
  WriteRegStr ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}"
"UninstallString" "$INSTDIR\uninst.exe"
  WriteRegStr ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}"
"DisplayIcon" "$INSTDIR\PC_Vulnerability.exe"
  WriteRegStr ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}"
```

```
"DisplayVersion" "${PRODUCT_VERSION}"
  WriteRegStr ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}"
"URLInfoAbout" "${PRODUCT_WEB_SITE}"
  WriteRegStr ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}" "Publisher"
"${PRODUCT_PUBLISHER}"
SectionEnd
Function un.onUninstSuccess
  HideWindow
  MessageBox MB_ICONINFORMATION|MB_OK "$(^Name)는(은) 완전히 제거되었습니다."
FunctionEnd

Function un.onInit
  MessageBox MB_ICONQUESTION|MB_YESNO|MB_DEFBUTTON2 "$(^Name)을(를) 제거하
시겠습니까?" IDYES +2
  Abort
FunctionEnd

Section Uninstall
  Delete "$INSTDIR${PRODUCT_NAME}.url"
  Delete "$INSTDIR\uninst.exe"
  Delete "$INSTDIR\PC_Vulnerability.exe"

  Delete "$SMPROGRAMS\PC_Vulnerability\Uninstall.lnk"
  Delete "$SMPROGRAMS\PC_Vulnerability\Website.lnk"
  Delete "$DESKTOP\PC_Vulnerability.lnk"
  Delete "$SMPROGRAMS\PC_Vulnerability\PC_Vulnerability.lnk"

  RMDir "$SMPROGRAMS\PC_Vulnerability"
  RMDir "$INSTDIR"

  DeleteRegKey ${PRODUCT_UNINST_ROOT_KEY} "${PRODUCT_UNINST_KEY}"
  DeleteRegKey HKLM "${PRODUCT_DIR_REGKEY}"
  DeleteRegKey HKCR "clsw"
  SetAutoClose true
SectionEnd
```

5.4 발표 PPT

OS 모니터링 시스템 구축

2019. 10. 29

지도교수: 이병천 교수님

팀명: 트론 (이재희, 나경민, 이영상, 이동준, 주영문)

1

목 차

- 조원 편성
- 주제 선정
- 구 상 도
- 추진 경과
- 개발 환경 및 시스템 개발
- 개발 시스템 운영
- 결론 및 기대효과

2

조원 편성

조 원	역 할
이재희	진단 프로그램 제작, PPT 작성 (총 괄)
이영상	진단 프로그램 제작, 보고서 작성
주영문	진단 프로그램 제작
나경민	연동 사이트 제작
이동준	연동 사이트 제작

3

주제 선정 (1/3)

보안 취약점 노리는 사이버 공격 급증...어떻게 대응할까

정보보안 취약점을 노리는 사이버 공격이 날로 증가함에 따라 보안 취약점 진단 중요성이 점점 높아지고 있다.

출처 : 전자신문 (2018.9.11)

한국인터넷진흥원이 7월 말 발표한 2분기 사이버 위협 동향 보고서에 따르면
올해 2분기에 확인된 고위험 보안 취약점은 891개에 달했다.
보안 관리자가 수 많은 취약점을 일일이 찾아 패치하는 데는 한계가 있는 만큼
IT 시스템·애플리케이션·웹 등에 있는 취약점을 자동 진단하는 취약점 분석 솔루션 수요는 더욱 늘어날 전망이다.

시스템 담당자는 취약점 현황 및 조치 결과는 물론 사용자가 원하는 다양한 기준에 부합하는 핵심 위협 정보와 진단 이력을 보다 쉽고 빠르게 파악할 수 있다.
즉 어느 부서가 어떤 보안 위협에 취약한지, 수많은 시스템 중 어떤 IT시스템이 공격 경로로 이용될 가능성이 높은지, 장기적으로 진단이 이뤄지지 않았거나 조치가 미흡했던 부분은 없는지 등을 보다 신속하고 정확하게 인지할 수 있다.

4

주제 선정 (2/3)

"2018년 한국 보안시장 2조원...전년比 4%↑"
가트너 전망 "2019년 한국 2조2천억원, 세계 1천240억달러"

2017-2019 년 전세계 분야별 보안 지출액 (단위: 백만 달러)

출처 : Zdnet Korea (2018.8.16)

시장 분야	2017	2018	2019
애플리케이션 보안	2,434	2,742	3,003
클라우드 보안	185	304	459
데이터 보안	2,563	3,063	3,524
IAM	8,823	9,768	10,578
인프라 보호	12,583	14,106	15,337
통합 리스크 관리	3,949	4,347	4,712
네트워크 보안 장비	10,911	12,427	13,321
그 외 정보 보안 소프트웨어	1,832	2,079	2,285
보안 서비스	52,315	58,920	64,237
소비자 보안 소프트웨어	5,948	6,395	6,661
총 계	**101,544**	**114,152**	**124,116**

5

주제 선정 (3/3)

KISA(한국인터넷진흥원) 배포
"주요정보통신기반시설 기술적 취약점 분석 평가 상세 가이드"

출처 : KISA (2018.6.28)

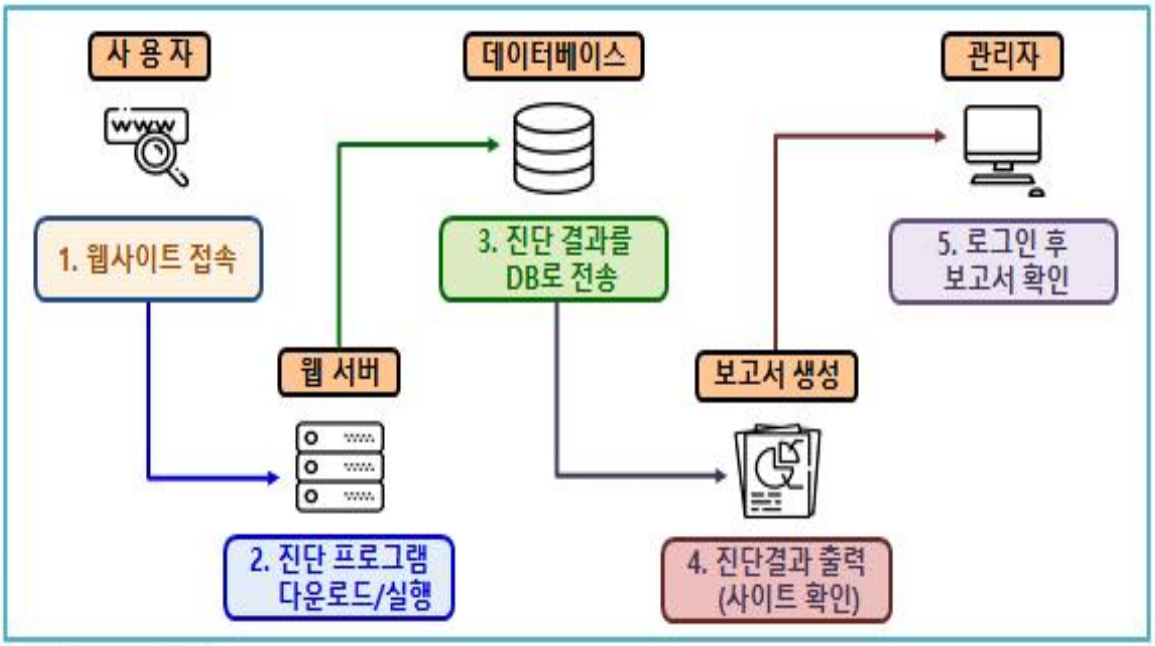
제목	주요정보통신기반시설 기술적 취약점 분석 평가 상세 가이드
담당자	융합기반보호팀 이성영 061-820-1641
등록일	2018-06-28
조회수	23859
첨부파일	(한국인터넷진흥원)_주요정보통신기반시설_기술적_취약점_분석_평가_상세_가이드_(2017).pdf

대분류	소분류	기술안내서 가이드	대상	수준
정보 보호 시스템 안전	정보 보호 시스템 관리	주요정보통신기반시설 기술적 취약점 분석 평가 상세 가이드	IT시스템관리자	중급

KISA 취약점 분석 평가 가이드에 기반하여 OS 모니터링 시스템을 개발

6

구상도



7

추진 경과

추진기간 (2019년) / 대상업무	3월	4월	5월	6월	7월	8월	9월	10월
계획 수립								
자료 조사/분석								
프로그램 제작								
웹 관리패널 작성								
성능시험 및 보완								
종합/보고서 작성								

8

개발 환경 및 시스템 개발 (1/8)

개발 환경

운영체제

- Windows 10

프로그래밍 언어

- C#
- JSON

웹 환경

- Nginx/PHP 7.2
- MariaDB
- Semantic UI

9

개발 환경 및 시스템 개발 (2/8)

메인 화면 구현

※ API: 특정 기능을 하는 함수

API

```
else if($calnm == 4) {
for($count=0; $count <=$TotalStartCount1; $count++){
$cve_id = $cve_id.$data->CVEList
}
return substr($cve_id , 0, -4);
}

else if($calnm == 5) {
for($count=0; $count <=$TotalSta
$cve_score = $cve_score.$data->C
}
return substr($cve_score , 0, -5
}
```

시각화

```
$no = 1; // 리스트 번호를 나타냄
$row = mysqli_fetch_array($result);
$list1 = explode(', ' , $row['TCP']);
$cnt = count($list1);

while($no-1 < $cnt){
echo "<tr>";
echo "<td>".$no."</td>";
echo "<td>" .$list1[$no-1]. "</td>";
echo "</tr>";
$no++; // 리스트 번호를 1씩 증가시킴
}
```

시각 정보 제공 취약점 진단 메뉴 등을 설정

10

개발 환경 및 시스템 개발 (3/8)

PC 정보 획득

운영체제 정보 획득

```
internal string OSName()
{
    string result = "Not found!";
    ConnectionOptions options = new ConnectionOptions();
    options.Impersonation = ImpersonationLevel.Impersonate;
    ManagementScope scope = new ManagementScope("/root/cimv2", options);
    scope.Connect();
    ObjectQuery query = new ObjectQuery("Select Caption From Win32_OperatingSystem");
    ManagementObjectSearcher search = new ManagementObjectSearcher(scope, query);
    ManagementObjectCollection queryCollection = search.Get();
    foreach (ManagementObject o in queryCollection)
    {
        result = o["Caption"].ToString();
        if (Environment.Is64BitOperatingSystem)
```

※ 캡슐화: 클래스 내부 접근만 가능한 Internal으로 구현 (보안상 이점)

분석 대상 PC를 확인/관리하는데 필수적인 정보를 획득

11

개발 환경 및 시스템 개발 (4/8)

서비스 진단

위험서비스 확인

```
ListViewItem XServiceList = new ListViewItem();
XServiceList.Text = service.DisplayName;
RegistryKey regKey1 = Registry.LocalMachine.OpenSubKey("SYSTEM\\CurrentControlSet\\services\\" + service.ServiceName);
string StartType = Convert.ToString(regKey1.GetValue("Start"));
// (StartType == 2, 3) 0: 부팅, 1: 시스템, 2: 자동, 3: 수동, 4: 사용 안함
int ServiceStatus = Convert.ToInt32(service.Status);
// (ServiceStatus == 0) 0: 실행 중, 1: 일시 중지 중, 2: 시작 보류 중, 3: 일시 중지 보류 중, 4: 일시 중지 후 서비스 시작 (대기
if (StartType == "2" || (StartType == "3") && (ServiceStatus == 0))
{
    if (StartType == "0") StartType = "부팅 시, 자동 시작";
    else if (StartType == "1") StartType = "시스템";
```

보안에 위험한 서비스가 구동 중인지 확인 ⇨ 취약 서비스 대응

12

개발 환경 및 시스템 개발 (5/8)

게시판 운영

게시글 작성

```
        <div id="write_area">
        <form class="ui form" action="write_ok.php" method="post" enctype="multipa
<div class="field">
  <label>제목:</label>
  <input type="text" name="title" id ="utitle" rows="1" cols="55" placeholder="제목">
</div>
                <div class="wi_line"></div>


                <div id="in_content">
                    <textarea name="content" id="ucontent"></textarea>
                    <script>
                      ClassicEditor
                      .create( document.querySelector( '#ucontent' ) )

                      .catch( error => {
                        console.error( error );
```

취약점 진단 결과 등을 공지하고 위협요인 정보 등을 안내

13

개발 환경 및 시스템 개발 (6/8)

보안요소 점검

메신저 설치 확인

```
internal bool Messenger(int check)
{
    var drive = Path.GetPathRoot(Environment.SystemDirectory);
    bool installed = false; string test;
    if (check == 0)
    { // Kakaotalk Check
        test = Convert.ToString(Registry.GetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft" +
               @"\Windows\CurrentVersion\Uninstall\KakaoTalk", "UninstallString", ""));
        if (File.Exists(test)) return true;
    }
    else if (check == 1)
    { // Line Check
        test = Convert.ToString(Registry.GetValue(@"HKEY_CURRENT_USER\Software\Microsoft" +
               @"\Windows\CurrentVersion\Uninstall\LINE", "UninstallString", ""));
```

메신저 소프트웨어들의 설치 유무를 점검

14

개발 환경 및 시스템 개발 (7/8)

보안요소 점검(1/2)

화면보호기 확인

```
// 화면 보호기 상태 확인
참조 1개
internal string ScreenSaver()
{
    string screensaver;
    RegistryKey reg = Registry.CurrentUser.OpenSubKey(@"Control Panel\Desktop", true);
    if (reg != null)
    {
        Object val = reg.GetValue("ScreenSaveActive");
        screensaver = Convert.ToString(val);
        Object ssval1 = reg.GetValue("ScreenSaveTimeOut");
        Object ssval2 = reg.GetValue("ScreenSaverIsSecure");
        if (val != null)
```

화면보호기 설정이 안전하게 되어 있는지 확인

15

개발 환경 및 시스템 개발 (8/8)

보안요소 점검(2/2)

※ CVE: 발표된 보안 취약점

CVE 취약점

```
internal void CVECheckRun(ListView lvw)
{
    var startInf
    {
        FileName
        Argument
        UseShell
        CreateNo
    };
    Process.Star
```

CVE: 소프트웨어의 보안 취약점을 가리키는 표기법 [미국 국립 표준 기술연구소(NIST) 지정]

CVE 정보 확인

Product	ID	Lisk Score
adobe_air 32.0.0.125	CVE-2013-0650	10
adobe_air 32.0.0.125	CVE-2013-0646	10
adobe_air 32.0.0.125	CVE-2013-1375	10
adobe_air 32.0.0.125	CVE-2013-1371	10
adobe_air 32.0.0.125	CVE-2010-2214	9.3
adobe_air 32.0.0.125	CVE-2010-2216	9.3
adobe_air 32.0.0.125	CVE-2010-2213	9.3
adobe_air 32.0.0.125	CVE-2010-0209	9.3
adobe_air 32.0.0.125	CVE-2010-2215	4.3

Vulmon(Vulnerability Intelligence Search Engine) : CVE 취약점을 진단하는 검색엔진

Vulmon의 개발 스크립트를 통해 진단형 결과를 인터페이스 제작

16

개발 시스템 운영 (1/5)

메인 화면

SW 중심으로 보안 취약점을 자동진단/모니터링하는 시스템 운영

17

개발 시스템 운영 (2/5)

로그인

로그인을 하면 취약점 진단 서비스를 제공받으며 시스템 사용이 가능

18

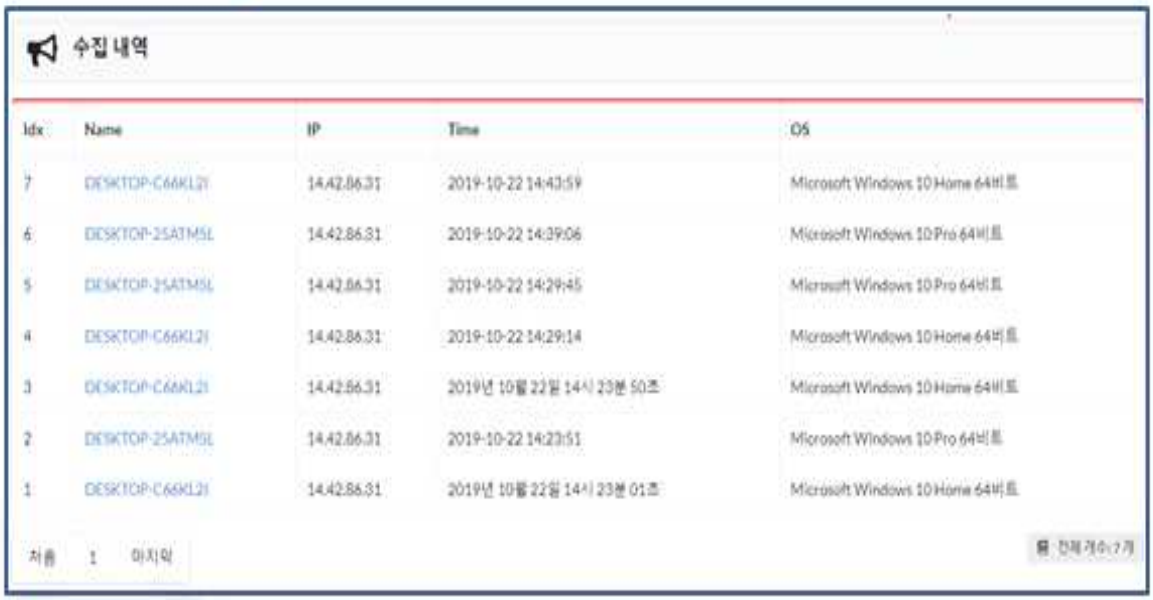
개발 시스템 운영 (3/5)

진단시스템 실행

19

개발 시스템 운영 (4/5)

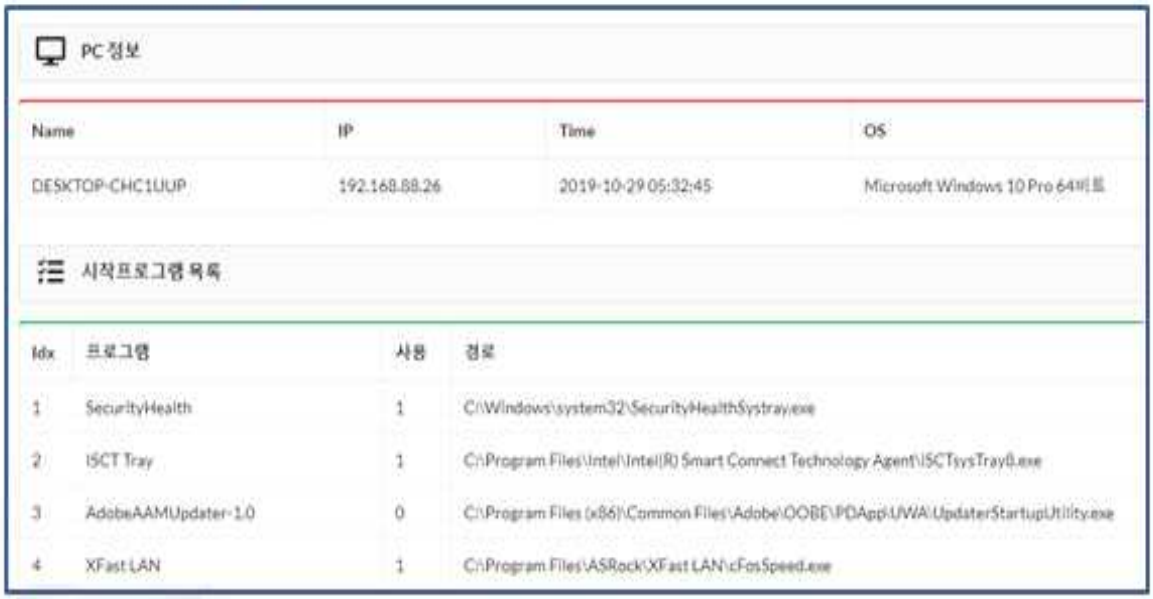
진단 PC 목록



20

개발 시스템 운영 (5/5)

상세 페이지



PC 정보

Name	IP	Time	OS
DESKTOP-CHC1UUP	192.168.88.26	2019-10-29 05:32:45	Microsoft Windows 10 Pro 64비트

시작프로그램 목록

Idx	프로그램	사용	경로
1	SecurityHealth	1	C:\Windows\system32\SecurityHealthSystray.exe
2	ISCT Tray	1	C:\Program Files\Intel\Intel(R) Smart Connect Technology Agent\ISCTsysTray0.exe
3	AdobeAAMUpdater-1.0	0	C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\UWA\UpdaterStartupUtility.exe
4	XFast LAN	1	C:\Program Files\ASRock\XFast LAN\cFosSpeed.exe

21

결론 및 기대효과

○ 결　　론

- KISA 취약점 분석 평가 가이드에 기반한 OS 모니터링 시스템이 완성
- 시스템에 의거 진단한 대상 PC에 대한 취약점 진단 보고서를 웹을 통해 확인 가능

○ 기대 효과

- OS 모니터링 시스템을 활용하여 관리자가 보안 취약점을 체계적으로 점검하고 적극적인 대응 처치가 가능할 것으로 기대

- 끝 -

22

Q&A

Thank you

23