〈ActiveX 의 위험성〉

수원대학교 flag 지선호(kissmefox@gmail.com)

ActiveX 는 JAVA 기술에 대응하여 마이크로소프트에서 내놓은 기술입니다. 일반 응용프로그램과 웹을 연결시켜주기 때문에 브라우저 상에서만 동작하는 스크립트 언어와는 달리 사용자의 로컬 시스템에 깊숙이 관여하게 됩니다. 그렇기 때문에 자바스크립트나 PHP 와 같은 스크립트 언어의 한계를 넘어 좀더 비주얼하고 편리하게 사용자에게 서비스를 제공할 수 있는 이점이 있습니다. 실제로 우리나라의 국가기관이나 금융권 등 주요 사이트도 ActiveX 기술을 이용하여 서비스를 제공하고 있고, 우리나라에서는 웹 표준 기술로 자리잡았다고 볼 수 있습니다.

하지만 성능이 뛰어난 만큼 그 기술이 악용된다면 위험성이 더욱 커질 수밖에 없습니다. 사용자의 로컬 시스템을 접근할 수 있다는 건 원격지에서 악의적인 목적으로 침투하기 위한 좋은 연결통로가 될 수 있습니다. 이미 예전부터 ActiveX 를 이용한 악성 코드 유포 사례가 늘어나기 시작했고, 갈수록 보안의식이 강화되면서 웹 방화벽, IDS 와 같은 장비에 의해 대부분의 포트들이 차단되고 감시되기 때문에 웹 서비스를 이용한 침입은 공격자에게 보다 쉽고, 안전한 길이 될 것입니다.

현재 XP SP2 이상에서 제공하는 웹 브라우저는 이와 같은 ActiveX 의 보안 문제점을 인지하고 웹에서 설치 요청이 들어오면 사용자에게 확인을 위한 경고창을 띄워 줍니다.

보안 의식이 있는 사용자라면 위의 경고 메시지를 확인하고 설치가 될 프로그램을 인지하고 설치 여부를 결정하게 될 것입니다. 하지만 대부분의 일반 사용자들은 저런 경고창을 그저 서비스를 제공받기 위한 설치 과정중의 하나로만 여기고 내용은 읽어보지도 않은채 빨리 설치가 되도록 무조건 클릭을 하게 됩니다. 어떤 사이트에서는 무조건 ActiveX 컨트롤을 설치하라는 페이지가 먼저 출력이 되기도 합니다.

워낙 ActiveX 기술이 사용되는 웹 사이트가 많기 때문에 일반 사용자는 ActiveX 에 의해 자동으로 설치되는 플랫폼의 위험성 여부를 판단하기가 힘듭니다. 아니, ActiveX 가 사용자의 시스템에 임의적으로 프로그램을 설치하고 레지스트리 값을 건드린다는 사실조차 인지하지 못하는것이 사실입니다.

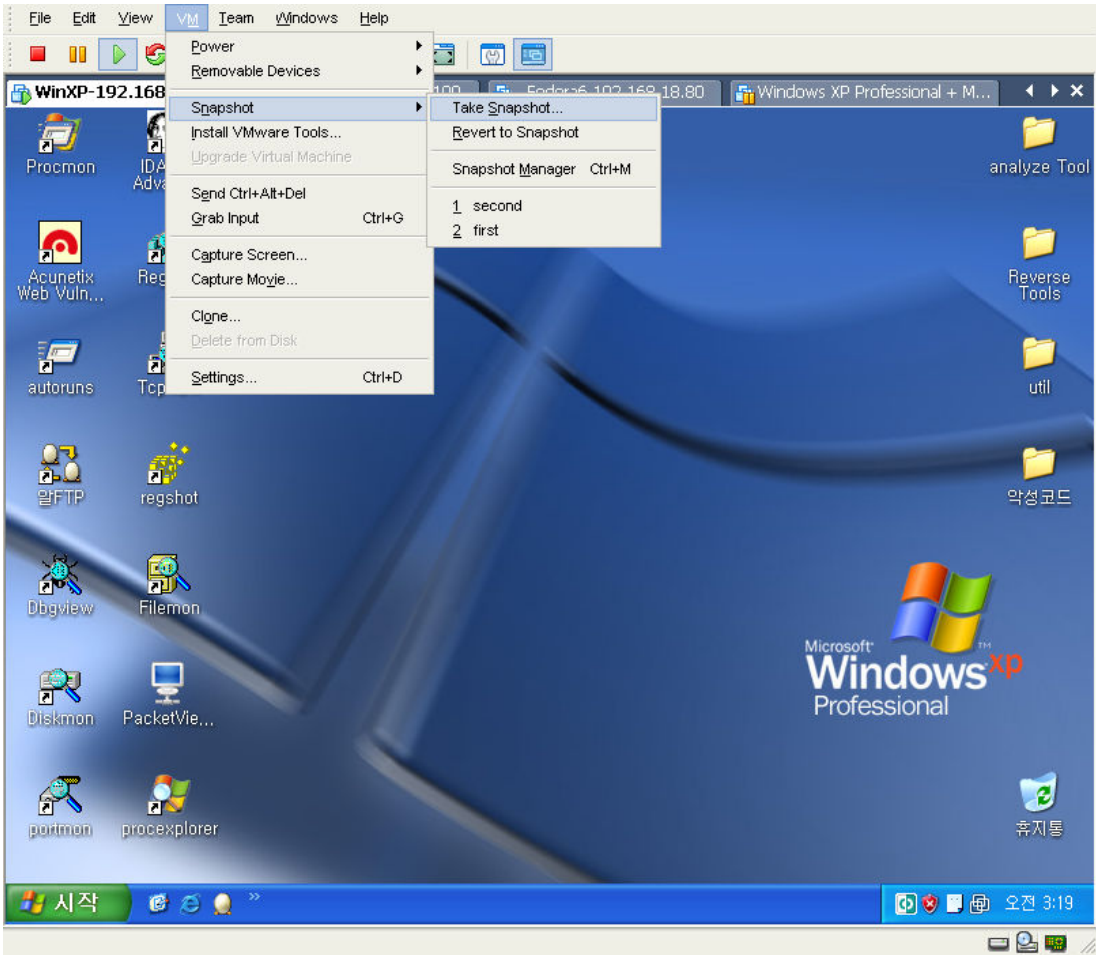
ActiveX 를 활용한 공격 시나리오는 얼마든지 다양하게 구성할 수 있습니다. 서비스를 제공하는 웹 서버에 침투하여 activeX 를 사용하는 웹페이지의 소스코드를 수정하여 참조하는 URL 의 주소를 변경하거나, 파일을 읽고쓰는 메소드를 이용하여 악성코드를 심거나 실행시킬 수도 있습니다. 또 각종 커뮤니티 게시판이나 이메일을 통해 스크립트를 삽입하여 불특정 다수에게 악성코드를 유포하는 방법도 사용될 수 있습니다. 가장 위험한건 많은 사용자가 접근하는 웹 사이트의 소스코드를 직접 수정하여 악성코드를 배포하는 경우입니다. 이경우는 이미 공격자에 의해 해당 웹 서버가 해킹을 당했다는 것이고, 관리자의 권한을 획득하여 공격자의 의도대로 코드를 수정하였다는 것이 됩니다.

직접 ActiveX 에 의해 유포되는 악성 코드를 분석하기 위해 웹서핑을 하던 중 nate.com 의 한 커뮤니티 게시판에서 ActiveX 설치 경고가 출력되었습니다. 의심이 가 소스의 링크 URL 을 확인하고 링크된 페이지의 소스코드를 확인해 보았습니다.

링크된 URL:
http://my.dreamwiz.com/killsman80/001.html
웹페이지 source code:

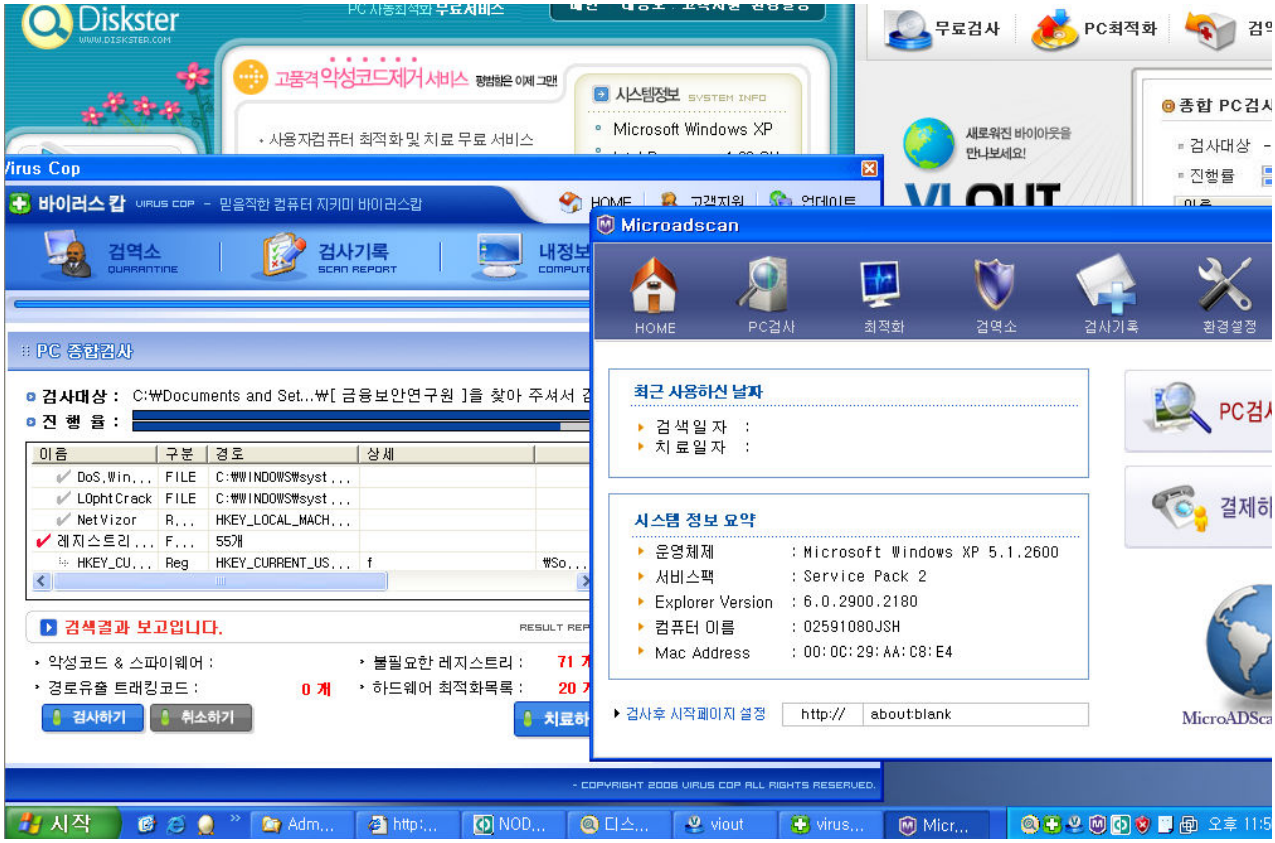
ActiveX 설치 구문이 5개가 보이고 설치되는 모듈이 현재 정상적으로 동작하고 있는 도메인 주소를 참조하고 있는 것을 볼 수 있습니다. 도메인 주소를 살펴보면 바이러스 치료를 위한 백신을 설치할 것이란 것을 짐작할 수 있습니다. 하지만 시스템에 이렇게나 많은 백신 프로그램이 설치되야 할지는 의문입니다.

설치되는 프로그램이 어떤 기능을 수행하는지 분석해볼 필요가 있는 것 같습니다. 먼저 안전한 분석을 위해 vmware에 서브 시스템을 구축하고 깨끗한 상태에서 스냅샷을 찍습니다.



OS : Windows XP sp2
사용하는 백신 : NOD32 2.70.16

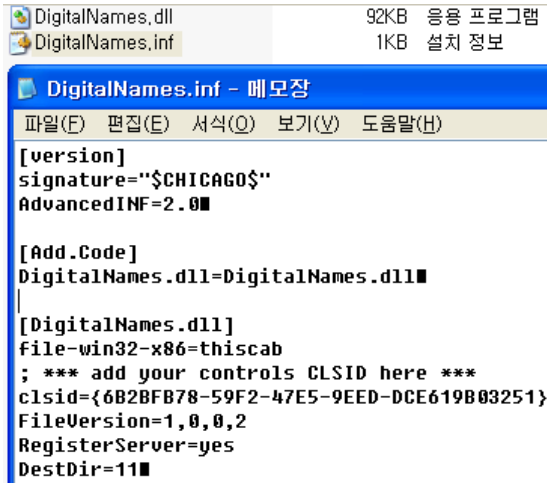
스냅샷을 찍기 전에 시스템을 검사한 결과 어떠한 바이러스나 에드웨어도 발견되지 않았습니다. 시스템은 초기설치 이후에 필요한 유틸만 설치하였고 백신 프로그램인 NOD32는 리소스는 적게 먹지만 검색기능이 우수하기 때문에 현재 시스템이 안전한 것을 어느정도 보장할 수 있습니다.
먼저 앞에서 확인한 웹페이지의 ActiveX 설치 모듈을 모두 설치해 보았습니다.



트레이에 아이콘들이 생겨나고 각종 악성코드검색 프로그램들이 자동적으로 시스템을 검사하며 리소스를 매우 잡아먹고 있습니다. 사용자 입장에서는 매우 불쾌한 일이 아닐 수 없습니다. 설치이전으로 돌아가 ActiveX 배포 모듈을 하나씩 설치해 보겠습니다. 웹페이지의 소스코드를 복사하여 저장한 뒤 OBJECT 구문을 하나씩 실행시켜 보았습니다.

```
<HTML><HEAD></HEAD>
<BODY>
<OBJECT id=AtlCtrl codeBase=http://upgrade1.digitalnames.net/toolbar1/install/DigitalNames.cab#ve
<!--<OBJECT codeBase=http://program.microadscan.com/Microadscan/microadscan.cab#version=1,0,0,0 h
<!--OBJECT id=viout codeBase=http://dw.viout.com/pgm/viout.cab#version=1,0,0,1 height=0 width=0 c
<!--OBJECT id=VirusCop codeBase=http://viruscop.co.kr/pgm/viruscop.cab#version=1,0,0,1 height=0 w
<!--OBJECT codeBase=http://cab1.diskster.com/hana/DHanaAct.cab#version=1,0,0,1 classid=CLSID:9F8E
```

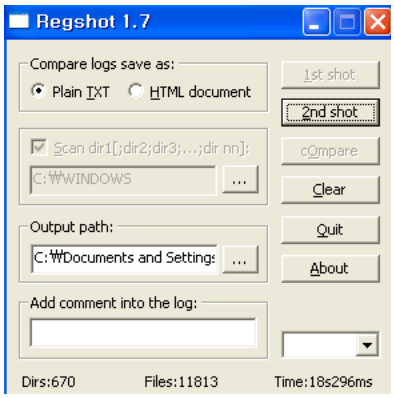
처음 설치되는 component 의 도메인으로 접속해보니 디지털네임즈라는 한글키워드서비스를 제공해주는 사이트입니다. DigitalNames.cab 파일을 받아 압축을 풀고 DigitalNames.inf 파일의 내용을 확인해 보았습니다.



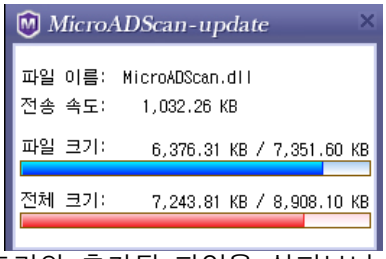
레지스트리에 clsid를 등록하고 DestDir=11 이기 때문에 윈도우 시스템 디렉토리에 Dll 파일이 복사되는 것을 확인할 수 있습니다. 아마 인터넷 익스플로어가 로딩될 때 같이 로딩되어 작업을 수행할 것입니다. 실제로 설치를 수행해 보니 특별히 악의적인 작업은 수행하지 않고 단지 BHO 의 역할을 수행하는 것을 확인할 수 있었습니다.

두 번째 설치 ActiveX 는 microadscan 이라는 익숙한 도메인을 가지고 있습니다. 설치 cab 파일을 미리 받아 파일의 내용을 확인해 보았습니다. 역시 설치를 위한 inf 와 MicroADScan.ocx 컨트롤이 포함되어 있습니다. 직접 설치를 해 보겠습니다.

설치를 하기 전에 레지스트리의 변화를 살펴보기위해 regshot 을 찍고 filemon 과 regmon 으로 상태 변화를 확인해 보겠습니다.



이후 사용자의 동의 없이 자동으로 MicroADScan 프로그램을 설치하는것을 볼 수 있습니다.



설치된 이후의 레지스트리와 추가된 파일을 살펴보니 악성코드가 숨겨져 있는것 같진 않습니다. 특별한 이상이 없으므로 다음 ActiveX 컨트롤을 설치해 보겠습니다.

〈OBJECT id=viout codeBase=http://dw.viout.com/pgm/viout.cab#version=1,0,0,1 height=0 width=0 classid=CLSID:2286D187-24CD-4B15-B5C3-6E973F34CDEF〉〈PARAM NAME="code1" VALUE="@0000751"〉〈PARAM NAME="code2" VALUE=""〉〈/OBJECT〉

다음 ActiveX 컨트롤 역시 viout 이라는 백신 프로그램 설치를 수행합니다. cab 파일안의 viout.inf 파일을 열어보니 MFC 의존성 프로그램으로 모듈을 자동으로 내려받는 것을 확인할 수 있습니다.

```
[version]
; version signature (same for both NT and Win95) do not remove
signature="$CHICAGO$"
AdvancedINF=2.0

[Add.Code]
vioutActiveX.ocx=vioutActiveX.ocx
; These are the necessary supporting DLLs for MFC 4.2 ActiveX Controls
mfc42.dll=mfc42.dll
msvcrt.dll=msvcrt.dll
olepro32.dll=olepro32.dll
; thiscab is a keyword which, in this case, means that dns.ocx
; can be found in the same .cab file as this .inf file
; file-win32-x86 is an x86 platform specific identifier
; See the ActiveX SDK - ActiveX Controls - Internet Component Download -
; Packaging component code for automatic download

[vioutActiveX.ocx]
file-win32-x86=thiscab
; *** add your controls CLSID here ***
clsid={2286D187-24CD-4B15-B5C3-6E973F34CDEF}
; Add your ocx's file version here.
FileVersion=1,0,0,1
RegisterServer=yes

; dependent DLLs
[msvcrt.dll]
; This is an example of conditional hook. The hook only gets processed
; if msvcrt.dll of the specified version is absent on client machine.
FileVersion=6,0,8168,0
hook=mfc42installer

[mfc42.dll]
FileVersion=6,0,8168,0
hook=mfc42installer

[olepro32.dll]
FileVersion=5,0,4261,0
hook=mfc42installer

[mfc42installer]
file-win32-x86=<LINK TYPE="GENERIC" VALUE="http://activex.microsoft.com
```

```
/controls/vc/mfc42.cab">http://activex.microsoft.com/controls/vc
/mfc42.cab</LINK>
; If dependent DLLs are packaged directly into the above cabinet file
; along with an .inf file, specify that .inf file to run as follows:
;InfFile=mfc42.inf
; The mfc42.cab file actually contains a self extracting executable.
; In this case we specify a run= command.
run=%EXTRACT_DIR%₩mfc42.exe
```

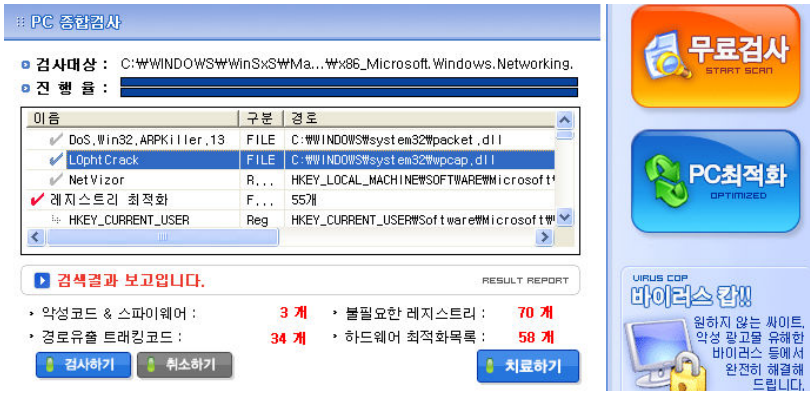
자동으로 설치가 되고 검사를 수행하니 예전에 없던 악성코드를 3개나 찾아내었습니다.

검사된 파일은 모두 winpcap 라이브러리 모듈이었습니다. 패킷 제어를 위해 필요한 라이브러리들이기 때문에 악의적인 용도로도 사용될 수 있으므로 백신회사의 판단에 따라 악성코드로 분류할 수 있는 여지가 있을 것 같습니다. 하지만 '경로유출 트래킹코드' 라는 이름으로 검사된 값들은 모두 단순한 쿠키 값이었습니다. 인증을 하기위해 개인정보가 담겨있는 쿠키값도 없고 단순히 웹서버와 세션이 이루어진 후에 남겨지는 카운터 값이나 팝업창 같은 정보들이 트래킹코드로 분류되는 것은 왠지 억지스러운 것 같습니다.

다음 ActiveX 컨트롤은 viruscop이라는 도메인을 가진 백신프로그램입니다.

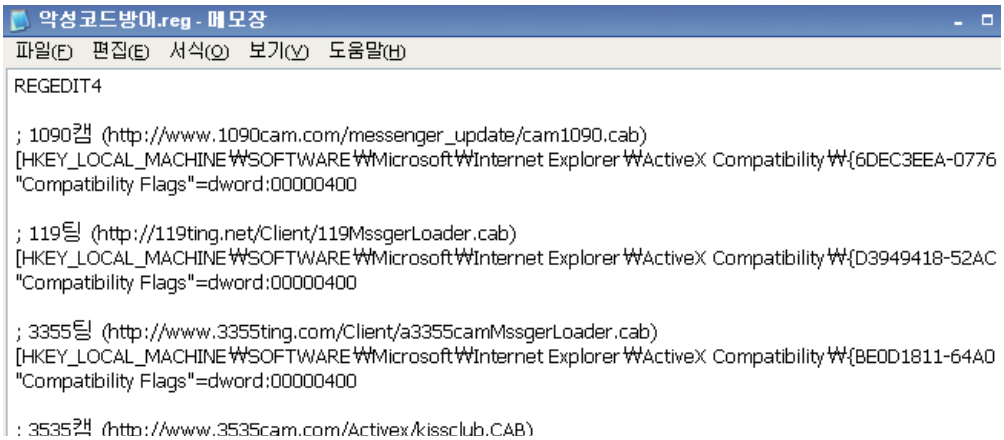
```
<OBJECT id=VirusCop codeBase=http://viruscop.co.kr/pgm/viruscop.cab#version=1,0,0,1 height=0 width=0 classid=CLSID:02F68151-CE20-4793-B092-BBF273D2C116><PARAM NAME="code1" VALUE="@0000751"><PARAM NAME="code2" VALUE=""></OBJECT>
```

설치해본 결과 위의 백신과 별다른 차이가 없습니다. 검사결과도 동일하고 왠지 같은 엔진을 사용한 것이 아닌가 하는 의심도 듭니다.

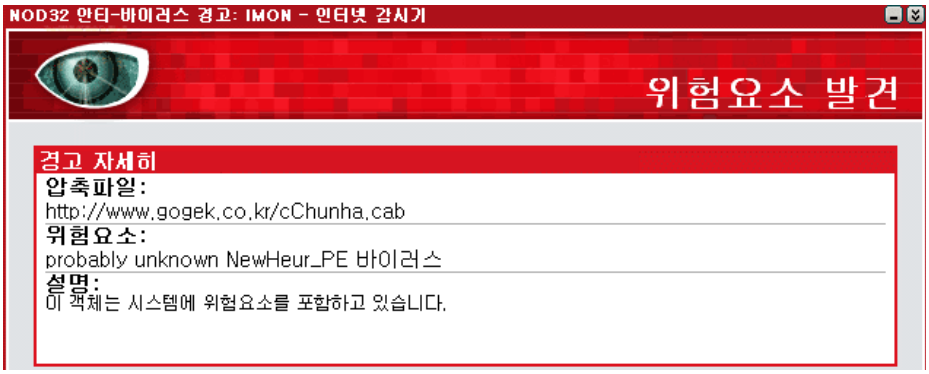


프로그램을 종료하려고 하면 무려 165개의 악성코드가 검색되었다며 치료를 강요합니다. 불필요한 레지스트리 값이나 쿠키값등을 모두 악성코드로 분류하여 사용자를 자극하는 것은 억지라는 생각이 듭니다.

마지막 ActiveX 컨트롤도 설치해보니 위의 백신들과 특별히 다른점이 없습니다. 잘못된 레지스트리 값을 악성코드라고 경고하며 치료를 위해 현금으로 결재할 것을 요구하고 있습니다. 모두 검사해봤지만 특별히 악의적인 목적보다는 상업적인 수단으로 제작된 ActiveX 컨트롤인것 같습니다. 다른 예제를 찾아보기위해 웹서핑을 하던 중 Killbit 이라는 ActiveX 개체 실행 방지 프로그램이 있었습니다. 다운받아서 확인해보니 단순히 레지스트리 값에 차단할 ActiveX 의 CLSID 값을 추가해 주는 형식이었습니다.



위 .reg 파일 안의 목록중에서 하나를 선택하여 분석해 보는 것이 좋을 것 같습니다.
http://www.gogek.co.kr/ 이라는 사이트에 접속해 보았습니다.



카스퍼스키 6.0 에서는 위험요소가 탐지되지 않지만 NOD32 에서는 아직 보고되지 않은 PE 파일 형식 위험요소라고 경고창을 띄웁니다. 왜 카스퍼스키에서 탐지하지 않은 위험요소를 NOD32 에서는 위험요소로 판단하였는지 ActiveX 컨트롤에 의해 설치되는 프로그램에 대해 분석해 보겠습니다.
.cab 파일을 받아 .inf 파일의 내용부터 살펴보겠습니다.

;DestDir은(는) Windows 디렉터리에서 10, Windows₩System(32) 디렉터리에서는 11이거나 Occache 디렉터리에서는 비워둡니다.
 	
[version] // hook 을 이용하기 위해
signature="$CHICAGO$"
	
	[MSINET.OCX]
	file-win32-x86=thiscab
	RegisterServer=Yes
	FileVersion=6,1,97,82 // file Key 가 가리키는 file 의 최소 요구 버전.

```
AdvancedINF=2.0

[DefaultInstall]
CopyFiles=install.files
RegisterOCXs=RegisterFiles
AddReg=AddToRegistry

[RInstallApplicationFiles]
CopyFiles=install.files
RegisterOCXs=RegisterFiles
AddReg=AddToRegistry

[DestinationDirs]
install.files= 11

[SourceDisksNames]
1=%DiskName%,cChunha.CAB,1

[Add.Code] // 인스톨될 모든 파일 나열
cChunha.ocx=cChunha.ocx
MSINET.OCX=MSINET.OCX
Msrchmom.exe=Msrchmom.exe
Msrchproc.exe=Msrchproc.exe
VB6KO.DLL=VB6KO.DLL

[install.files]
cChunha.ocx=cChunha.ocx
MSINET.OCX=MSINET.OCX
Msrchmom.exe=Msrchmom.exe
Msrchproc.exe=Msrchproc.exe
VB6KO.DLL=VB6KO.DLL

[SourceDisksFiles]
cChunha.ocx= 1
MSINET.OCX= 1
Msrchmom.exe= 1
Msrchproc.exe= 1
VB6KO.DLL= 1

[cChunha.ocx]
file-win32-x86=thiscab
RegisterServer=yes
clsid={0C5A73F6-1E4C-4D3C-8E0D-EA6A654F3954}
DestDir=
FileVersion=1,0,0,0

[Msrchmom.exe]
file-win32-x86=thiscab
RegisterServer=no
DestDir=10
FileVersion=1,0,0,0

[Msrchproc.exe]
file-win32-x86=thiscab
RegisterServer=no
DestDir=10
FileVersion=1,0,0,0

[VB6KO.DLL]
file-win32-x86=thiscab
RegisterServer=no
DestDir=11
FileVersion=6,0,89,88

[Setup Hooks] // Add.Code 에서 file 들이 세팅되기 전 실행되어야 할 hook 을 나열한다.
AddToRegHook(<--hookname)=AddToRegHook

[AddToRegHook]
InfSection=DefaultInstall2

[DefaultInstall2]
AddReg=AddToRegistry

[AddToRegistry]
HKLM,"SOFTWARE₩Classes₩CLSID₩{0C5A73F6-1E4C-4D3C-8E0D-EA6A654F3954}₩ImplementedCategories₩{7DD95802-9882-11CF-9FA9-00AA006C42C4}"
HKLM,"SOFTWARE₩Classes₩CLSID₩{0C5A73F6-1E4C-4D3C-8E0D-EA6A654F3954}₩ImplementedCategories₩{7DD95801-9882-11CF-9FA9-00AA006C42C4}"
HKCR,"Licenses",,,"Licensing: Copying the keys may be a violation of established copyrights."

[RegisterFiles]
%11%₩MSINET.OCX
%11%₩cChunha.ocx
```

inf 파일의 분석으로 위험요소로 의심받는 Msrchmom.exe 와 Msrchproc.exe 가 c:₩windows 디렉토리에 복사가 되고 생성되는 레지스트리의 CLSID값을 알 수 있습니다. 또한 인터넷 전송 ActiveX 제어 라이브러리인 MSINET.OCX 이 설치되는 것도 확인할 수 있습니다. 일부 백도어나 트로이목마에서 이 라이브러리를 설치하는 경우를 볼 수 있기 때문에 일단은 의심을 해 보아야 할 것 같습니다.

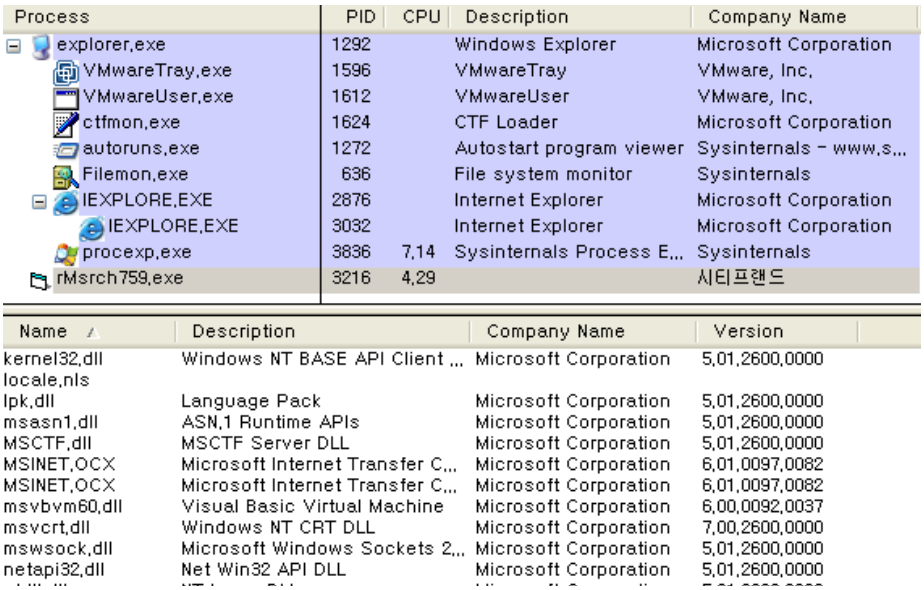
직접 설치를 해 보겠습니다. 설치가 되기전에 시스템의 변화를 확인하기위해 레지스트리 값을 저장하

고 툴들을 활용하여 새롭게 쓰여지는 파일이나 open 되는 포트의 변화를 관찰합니다.

ActiveX 컨트롤이 설치된 후의 변화를 살펴보겠습니다.

Process	Request	Path	Result	Other
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Attributes: RHA
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Options: Open Access: 001(
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Length: 749
Msrchmom.exe:3208	CLOSE	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Attributes: RHA
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Options: Open Access: Rea
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Length: 749
Msrchmom.exe:3208	CLOSE	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Options: Open Access: 001
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	Length: 749
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	FileNetworkOpenInformation
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩WindowsShell.Config	NOT FOUND	Options: Open Access: 001
Msrchmom.exe:3208	CLOSE	C:₩WINDOWS₩WindowsShell.Manifest	SUCCESS	
Msrchmom.exe:3208	READ	C:₩WINDOWS₩Msrchmom.exe	SUCCESS	Offset: 32768 Length: 4096
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩System32₩vb6ko.dll	SUCCESS	Attributes: A
Msrchmom.exe:3208	READ	C:₩WINDOWS₩system32₩oleaut32.dll	SUCCESS	Offset: 352256 Length: 327
Msrchmom.exe:3208	READ	C:₩WINDOWS₩system32₩oleaut32.dll	SUCCESS	Offset: 344064 Length: 819
Msrchmom.exe:3208	READ	C:₩WINDOWS₩system32₩oleaut32.dll	SUCCESS	Offset: 172032 Length: 204
Msrchmom.exe:3208	QUERY IN...	C:₩WINDOWS₩System32₩MSVBVM60.DLL	SUCCESS	Attributes: A
Msrchmom.exe:3208	READ	C:₩WINDOWS₩System32₩MSVBVM60.DLL	SUCCESS	Offset: 1019904 Length: 20
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩	SUCCESS	Options: Open Directory Ac
Msrchmom.exe:3208	DIRECTORY	C:₩WINDOWS₩	NO SUCH FILE	FileBothDirectoryInformati
Msrchmom.exe:3208	CLOSE	C:₩WINDOWS₩	SUCCESS	
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩	SUCCESS	Options: Open Directory Ac
Msrchmom.exe:3208	DIRECTORY	C:₩WINDOWS₩	NO SUCH FILE	FileBothDirectoryInformati
Msrchmom.exe:3208	CLOSE	C:₩WINDOWS₩	SUCCESS	
Msrchmom.exe:3208	READ	C:₩WINDOWS₩System32₩MSVBVM60.DLL	SUCCESS	Offset: 897024 Length: 122
Msrchmom.exe:3208	OPEN	C:₩WINDOWS₩Msrchproc.exe	SUCCESS	Options: Open Access: Rea
Msrchmom.exe:3208	CREATE	C:₩WINDOWS₩rMsrch759.exe	SUCCESS	Options: OverwriteIf Acce
Msrchmom.exe:3208	READ	C:₩WINDOWS₩Msrchproc.exe	SUCCESS	Offset: 0 Length: 65024
Msrchmom.exe:3208	WRITE	C:₩WINDOWS₩rMsrch759.exe	SUCCESS	Offset: 0 Length: 65024
Msrchmom.exe:3208	READ	C:₩WINDOWS₩Msrchproc.exe	SUCCESS	Offset: 65024 Length: 6502
Msrchmom.exe:3208	WRITE	C:₩WINDOWS₩rMsrch759.exe	SUCCESS	Offset: 65024 Length: 6502
Msrchmom.exe:3208	READ	C:₩WINDOWS₩Msrchproc.exe	SUCCESS	Offset: 130048 Length: 650
Msrchmom.exe:3208	WRITE	C:₩WINDOWS₩rMsrch759.exe	SUCCESS	Offset: 130048 Length: 102
Msrchmom.exe:3208	READ	C:₩WINDOWS₩Msrchproc.exe	END OF FILE	Offset: 131072 Length: 650
Msrchmom.exe:3208	CLOSE	C:₩WINDOWS₩rMsrch759.exe	SUCCESS	

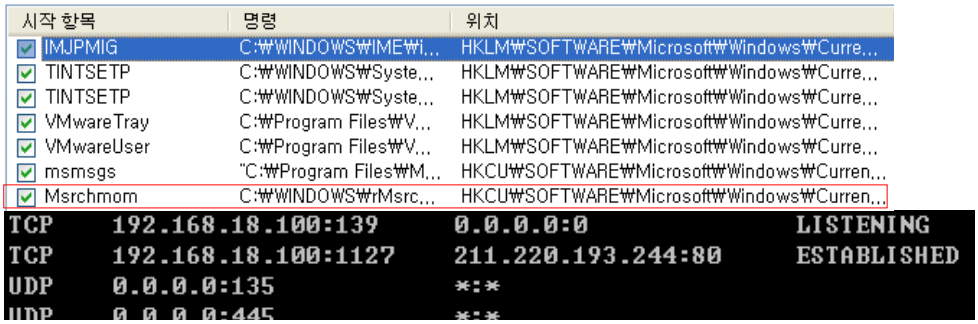
Msrchmom.exe 파일이 필요한 라이브러리들을 참조한 뒤에 rMsrch759.exe 파일을 CREATE 합니다. MSVBVM60.DLL 을 참조하는 것을 보아 비주얼 베이직 라이브러리로 제작된걸 확인할수 있습니다. 설치가 끝나면 rMsrch759.exe 파일이 메모리에 로드된 것을 확인할 수 있습니다.



Process	PID	CPU	Description	Company Name
explorer.exe	1292		Windows Explorer	Microsoft Corporation
VMwareTray.exe	1596		VMwareTray	VMware, Inc.
VMwareUser.exe	1612		VMwareUser	VMware, Inc.
ctfmon.exe	1624		CTF Loader	Microsoft Corporation
autoruns.exe	1272		Autostart program viewer	Sysinternals - www.s...
Filemon.exe	636		File system monitor	Sysinternals
IEXPLORE.EXE	2876		Internet Explorer	Microsoft Corporation
IEXPLORE.EXE	3032		Internet Explorer	Microsoft Corporation
procexp.exe	3836	7.14	Sysinternals Process E...	Sysinternals
rMsrch759.exe	3216	4.29		시티프랜드

Name	Description	Company Name	Version
kernel32.dll	Windows NT BASE API Client ...	Microsoft Corporation	5.01.2600.0000
locale.nls			
lpk.dll	Language Pack	Microsoft Corporation	5.01.2600.0000
msasn1.dll	ASN.1 Runtime APIs	Microsoft Corporation	5.01.2600.0000
MSCTF.dll	MSCTF Server DLL	Microsoft Corporation	5.01.2600.0000
MSINET.OCX	Microsoft Internet Transfer C...	Microsoft Corporation	6.01.0097.0082
MSINET.OCX	Microsoft Internet Transfer C...	Microsoft Corporation	6.01.0097.0082
msvbvm60.dll	Visual Basic Virtual Machine	Microsoft Corporation	6.00.0092.0037
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	7.00.2600.0000
mswsock.dll	Microsoft Windows Sockets 2,...	Microsoft Corporation	5.01.2600.0000
netapi32.dll	Net Win32 API DLL	Microsoft Corporation	5.01.2600.0000

시티프랜드라는 상호명이 보입니다. 프로세스의 스트링 값을 확인해 보았지만 특별히 악의적인 목적의 흔적은 보이지 않았습니다. 위험한 건 이 프로세스가 계속 시스템에 상주하면서 1127번 포트와 연결상태를 유지시켜놓고 있다는 것입니다.

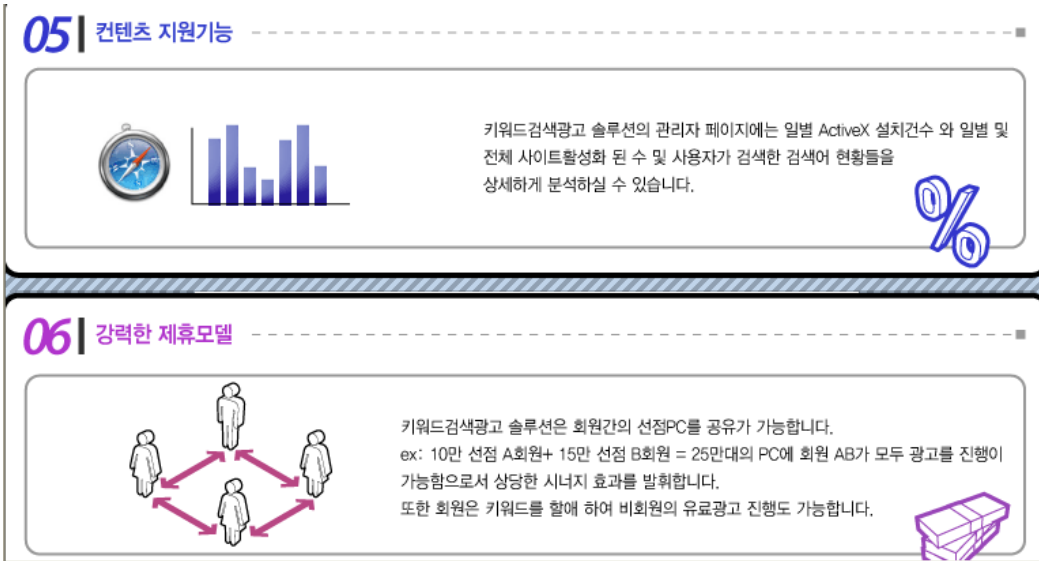


시작 항목	명령	위치
IMJPMIG	C:₩WINDOWS₩IME₩i...	HKLM₩SOFTWARE₩Microsoft₩Windows₩Curre...
TINTSETP	C:₩WINDOWS₩Syste...	HKLM₩SOFTWARE₩Microsoft₩Windows₩Curre...
TINTSETP	C:₩WINDOWS₩Syste...	HKLM₩SOFTWARE₩Microsoft₩Windows₩Curre...
VMwareTray	C:₩Program Files₩V...	HKLM₩SOFTWARE₩Microsoft₩Windows₩Curre...
VMwareUser	C:₩Program Files₩V...	HKLM₩SOFTWARE₩Microsoft₩Windows₩Curre...
msmsgs	"C:₩Program Files₩M...	HKCU₩SOFTWARE₩Microsoft₩Windows₩Curren...
Msrchmom	C:₩WINDOWS₩rMsrc...	HKCU₩SOFTWARE₩Microsoft₩Windows₩Curren...

```
TCP    192.168.18.100:139     0.0.0.0:0              LISTENING
TCP    192.168.18.100:1127    211.220.193.244:80     ESTABLISHED
UDP    0.0.0.0:135            *:*
UDP    0.0.0.0:445            *:*
```

시작프로그램에 자동으로 등록이 되어있습니다. IP 211.220.193.244의 80 포트와 계속 연결 상태를 유지하고 있기 때문에 공격자에게 취약점을 제공하는 경우가 발생할 수 있습니다. 저 IP 는 이 ActiveX 컨트롤이 설치된 웹서버인 gogek.co.kr 의 IP 주소였습니다. 특별히 로컬 시스템에 피해를 주지 않더라도 임의의 포트를 open 시켜 원격지에서의 공격 취약점을 증가시킨 셈입니다.

프로그램이 설치된 사이트에 접속하여 시스템에서 어떤 기능을 수행하는지 확인해 보았습니다. 키워드 검색광고 솔루션이란 이름으로 웹브라우저의 주소창에 한글로 찾고자하는 키워드를 입력하면 위의 프로세스가 키워드를 먼저 가로채어 업체에 등록된 사이트로 자동이동시키는 기능을 수행하고 있었습니다. 주소창에 입력되는 값을 후킹하려면 굳이 시스템에 늘 상주하여 가로챌 키 값을 기다릴 필요가 없습니다. API 후킹이나 메시지 후킹방식으로 라이브러리를 작성하여 웹 브라우저가 로딩될 때 같이 로딩될 수 있도록 IAT에 추가시켜주거나 로더를 제작하여 후킹작업을 수행할 수도 있었을 것입니다.
솔루션에 대한 설명을 더 읽어보니 위험한 문구가 보입니다.



사업을 목적으로 일반 사용자의 PC 에 포트를 열어놓았다지만, 다르게 보면 irc bot 이나 백도어같은 악성코드와 다를바가 없을 것 같습니다.. NOD32에서 위험요소로 탐지했던 이유가 있는 것 같습니다.

실질적인 악성코드를 내포한 ActiveX 컨트롤 분석을 위한 예제를 찾을 수 없었지만, 위의 과정만으로도 그 위험성을 충분히 알 수가 있습니다. 설치되는 ActiveX 컨트롤의 제목이나 설명으로만 판단하지 말고 이렇게 간단하게 분석을 수행해 보는 것도 좋을 것 같습니다.