요약

본 자료는 악의적인 외 · 내부자에 의한 웹 백도어를 예방하고 탐지하기 위한 방안을 담고 있다. 예방을 위한 관리적 방안으로 버전 관리 시스템과 개발 환경 · 운영환경의 분리, 역할에 따른 상호 견제를 권고하며, 탐지를 위한 기술적 방안으로 정기적인 파일 목록 감사, 패턴 매칭을 통한 웹 백도어 검출을 권고한다.

서문

웹 백도어(Backdoor)는 웹 서비스를 제공하는 조직에 심각한 보안 위협을 초래한다. 웹 백도어는 시스템 명령어 실행 · 파일 업로드 · DB 클라이언트 등의 역할을 수행하는 PHP, ASP, JSP 등과 같은 웹 프로그래밍언어로 작성된 프로그램으로 웹쉘(Webshell)이라고 흔히 불린다. 웹 백도어는 외부의 악의적인 공격자 또는 편의를 목적으로 한 내부자에 의해서 생성될 수 있다. 외부의 공격자와 내부자가 웹 백도어를 숨기는 이유는 허가되지 않은 권한과 정보를 획득하기 위한 것이다. 이로 인해 조직은 내부의 주요 정보가 외부로 노출되거나 비권한자에게 권한이 노출되며, 내부로의 침해 또는 타 조직(사이트)을 공격하는데 악용될 수 있는 보안 위협에 빠지게 된다.

웹 백도어로 인한 보안위협을 예방하기 위해서는 관리적 보안과 보안점검을 실시해야 한다. 웹 백도어는 내부자와 외부 공격자에 의해서 생성될 수 있으므로, 사전예방과 정기적인 점검이 필요하다. 사전예방 활동으로 조직과 시스템을 개선하는 관리적 보안의 강화가 이루어져야 한다. 정기적인 점검 활동으로는 웹 백도어가 현재 운영 중인 시스템에 존재하는 지를 정기적으로 점검하여 즉각적인 조치를 할 수 있도록 해야 한다.

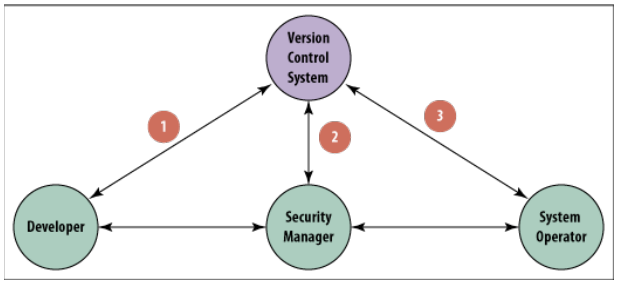
관리적 보안 강화

관리적 보안은 조직 내부의 이해관계자(개발자 · 시스템운영자 · 보안담당자)를 대상으로 보안체계를 수립하고 이에 대한 교육을 실시하는 것이다. 이를 위해 소프트웨어(소스 코드)에 대한 이력관리를 실시하고, 개발자(Developer) · 시스템운영자(System Operator) · 보안담당자(Security Manager)는 내부자에 의한 보안사고를 예방하기 위해 각각 역할 또는 조직을 분리한다. 또한 개발환경과 운영 환경은 반드시 구분한다.

첫째, **소스 코드에 대한 버전 관리**를 실시한다. 소스 코드에 대한 이력 관리는 개발자의 임의적인 애플케이션 조작을 예방하고, 보안감사를 수행할 수 있는 자료를 제공해준다. 또한 명백한 히스토리가 남음으로 인해 추후 침해사고 등으로 인해 소스가 변경되었을 때 이를 추적 · 관리할 수 있다.

둘째, **개발 환경과 운영 환경을 구분**한다. 개발환경과 운영 환경에 구분되어 있지 않을 경우, 개발을 하면서 생성되는 다양한 임시 · 테스트 파일들이 운영 환경에 그대로 방치되는 문제점이 발생한다. 개발단계에서 개발자는 보안보다는 기능 구현에 초점을 맞추고 있으며, 편의성을 우선시 하기 때문에 다양한 임시 · 테스트 파일들을 생성하고 이를 사용하게 된다. 이 중에는 백도어의 기능을 수행하는 파일들도 존재한다. 이들이 그대로 방치될 경우 외부 공격자에게 악용되거나, 내부자가 이를 악용할 수도 있다. 개발자가 운영 환경의 소스 코드를 수정할 경우, 개발자의 실수 또는 악감정 등의 문제로 인해 장애 및 악의적인 행위(소스파일 삭제 · 불법적 이익 취득 등)가 발생할 수 있다. 또한 개발자가 퇴사후 악의를 품고 외부에서 내부 시스템 및 정보를 침해하는 등의 행위가 발생할 수 있다. 따라서 운영 환경에 개발자가 직접적으로 접근하는 행위는 차단되어야 한다.

셋째, **역할에 따른 상호 견제**가 이루어져야 한다. 개발자 · 보안담당자 · 시스템운영자는 버전 관리 시스템(Version Control System)을 이용하여 아래 그림과 같이 업무를 처리함으로써, 상호견제를 통한 보안 강화를 실시할 수 있다. 이는 각각의 조직별 역할 구분으로 생각하여도 무방하다.



1. 개발자는 소스 코드를 개발·테스트한 후, 운영 환경에 적용할 버전을 버전 관리 시스템에 반영한다. 이때 변경에 대한 상세 주석을 작성하여, 변경 부분에 대한 이유와 히스토리가 반드시 남기도록 한다.
2. 보안담당자는 개발자가 버전 관리 시스템에 반영한 변경 내역을 확인하고, 소스 코드와 주석 등을 분석한다. 이때 보안상 문제가 될 수 있는 구현 부분이 있다면, 통보하여 수정될 수 있도록 한다. 또한 변경 내역과 다르게 구현된 부분이 있는 지를 점검하도록 한다. 만약, 모르는 코드가 있다면 반드시 개발자와 함께 리뷰를 실시하도록 한다.
3. 시스템운영자는 보안담당자가 검수를 완료하면, 버전 관리 시스템에서 해당 버전을 다운로드 받아 운영 환경에 반영하고 이상없이 동작하는지를 확인하다. 이때, 버전 관리스시템에서 생성되는 불필요한 파일들이 운영서버 측에 남지않도록 조치한다.

이러한 체계를 수립한 후에는 개발자, 시스템 운영자 등을 대상으로 보안절차에 대한 교육을 실시하여, 원활한 업무 협조와 내부자에 의한 보안예방을 사전에 할 수 있도록 한다.

정기적인 탐색 실시

웹 백도어 탐색은 주기적으로 악성코드를 내포하고 있는 파일을 검색해야 한다. 웹 백도어를 탐색하기 위해 실시간 탐지와 사후 탐지 방법 중 하나를 선택할 수 있다. 실시간 탐지는 운영 환경에 대한 자원 소모와 보안 솔루션의 오작동으로 인한 운영 위협이 존재할 수 있기 때문에 주기적으로 악성코드 탐색기를 이용한 점검이 효과적이다.

웹 백도어 탐지 도구 현황

웹 백도어 탐지에 대해 현재 국가기관, 보안기업, 보안그룹에서 유상·무상으로 관련 도구를 제공하고 있다. 한국인터넷진흥원(KISA)에서 제공하는 휘슬(WHISTL)은 단독 실행파일(Standalone)을 실행하여 웹 백도어를 탐지하는 형태이다. 보안 강화를 위한 목적이 아닌 사후대응을 위한 사고분석 도구로써 배포하고 있다. 이러한 이유에서인지 자유로운 배포 형태가 아니며, 사용신청서를 작성하여 제출 후 제공받을 수 있다[1]. (주)유엠브이기술의 ShellClean은 상용제품으로, 웹 서버에 설치하는 에이전트와 탐지내역을 모니터링하는 중앙집중 관리서버로 이루어져있다[2]. Geekslab에서 제공하는 WSF는 .NET 2.0을 기반으로 하고 있어 MS 윈도우즈에서만 실행가능하며, 도구 및 소스를 공개하고 있다[3]. 공개용이라는 장점을 가지고 있기는 하나, .NET 환경으로 구축되어 있어 MS 윈도우즈 서버 측에 .NET 프레임워크를 별도로 설치해야 한다는 단점을 가지고 있으며, Unix 환경에서의 점검이 불가능하다는 한계점을 가지고 있다.

구분	형태	주관	실행환경	L/C
WHISTL	S.A	KISA	리눅스 커널 2.4/2.6, MS 윈도우즈	?
ShellClean	C/S	(주)유엠브이기술	JAVA(에이전트)	상용
WSF	S.A	Geekslab	.NET 2.0 이상	GPL V3

[표 1] 웹 백도어 탐지 도구 현황

웹 백도어의 특징

웹 백도어를 탐지하기위해서는 웹 백도어의 특징을 알아야 한다. 웹 백도어는 웹 서버를 운영하는 시스템의 권한 획득 및 정보 획득을 주목적으로 하고 있다. 이러한 목적을 이루기 위해 다양한 언어로 개발되어 이용되고 있으며, 제공 기능 역시 나날이 발전하고 있다. 레지스트리 조작, 익스플로잇(exploit) 실행, 네트워크 포트(port) 스캐닝, 리버

스 커넥션(Reverse Connection) 등과 같은 특화된 기능을 제공하기도 하지만, 일반적으로 다음과 같은 기능들이 있다[4].

- 시스템 명령어 실행
- 임의의 파일 업로드 · 다운로드
- DB 클라이언트
- 시스템 정보 확인
- 파일 및 디렉터리 조작(수정 · 삭제 · 생성 등)

최근 등장하는 웹 백도어들은 위의 기능들을 종합적으로 제공하는 것이 일반적이나, 공격자의 성향에 따라서 단일 기능만을 포함하고 있는 웹 백도어를 선호하기도 한다.

웹 백도어의 탐지 방법

웹 백도어를 탐지하는 방법은 파일 목록 감사와 파일 내부의 패턴 검사이다. 각 방법은 이해관계자, 사전 준비 정도, 해당 웹 서비스에 이해도에 따라서 장점과 단점을 가지고 있다.

첫째, **파일 목록을 감사**하는 것은 파일 목록에 생성시간 · 변경일자 · 해쉬값 등을 목록화하고 주기적으로 변경 여부를 확인하는 것이다. 파일 목록 정보를 이용하여 비정상적으로 이루어진 변경을 탐지할 수 있으며, 업로드 디렉터리와 같이 비정상적인 위치에 존재하는 웹 실행 파일(php, asp, jsp 등의 확장자)의 확인을 통해 웹 백도어를 신속하게 찾아낼 수 있는 장점이 있다. 이러한 점검방법은 파일 목록이 사전에 생성되어야 한다는 점과 생성된 파일 목록을 신뢰할 수 있는가라는 문제점을 가지고 있다. 또한 소스 코드에 대한 감사를 하지 않는 상황에서는 내부자에 의해 생성된 웹 로그 파일은 탐지하기 어렵다는 제약사항을 가지고 있다. 내부자는 소스 업데이트 과정에 정상적인 파일인 것처럼 웹 백도어를 숨겨둘 수 있으며, 기존의 파일을 수정하여 백도어 기능을 숨길 수 있기 때문이다.

둘째, **패턴 매칭**은 소스파일에서 웹 백도어에서 자주 사용되는 패턴과 이미 알려진 웹 백도어의 패턴을 검사하는 것이다. 각 개발 언어별로 시스템 명령어를 호출하는 함수 등과 같이 웹 백도어의 기능을 구현하기 위해서 사용되는 함수들과 악성코드 제작자들이 자주 사용하는 cmd, webshell 등과 같은 문자열을 검증함으로써 의심스러운 파일들을 추척할 수 있다. 또한 이미 널리 알려져있는 r57shell, c99shell 등과 같은 웹 백도어에서 검출한 패턴을 이용하여 잘 알려진 웹 백도어를 검출할 수 있다. 패턴 매칭을 통한 검출은 새로 등장하는 패턴을 꾸준히 업데이트해야 한다는 단점이 있다. 또한 공격자들이 패턴 매칭 기법을 우회하기 위해 유니코드(UNICODE), 인코딩(Encoding) 등과 같은 각종 우회기법들을 적용하고 있기 때문에 전적으로 패턴 매칭에 의존한 검출에는 한계점이 존재한다. 현재 패턴 매칭을 이용한 탐지는 완벽하지는 못하지만 가장 효과적이고 현실적인 대안이다.

웹 백도어 패턴

패턴 매칭을 통한 웹 백도어 탐지를 하기위해서는 정확성이 높은 패턴들이 필요하다. 최근 웹 백도어의 경향은 패스워드 설정, 패턴 탐지 우회(인코딩, 문자열 분할), 정상적인 소스파일에 포함되는 것이다. 웹 백도어가 일반적으로 가지고 있는 패턴과 잘 알려진 웹 백도어의 특정 문자열 패턴이 사용될 수 있다. 너무 무분별한 패턴 등록을 할 경우 오탐율이 높아지고 실행속도가 떨어지기 때문에 주의가 필요하다.

웹 백도어가 일반적으로 사용하는 패턴은 **위험 함수들과 특징적인 문자열**이다[5]. 앞서 살펴본 웹 백도어의 다양한 기능들 중에서 오탐을 최소화하면서 점검에 사용할 수 있는 것은 **시스템 명령어 실행**이다. DB 호출, 파일 조작 등은 일반적으로 사용되기 때문에 많은 오탐을 유발 시킬 수 있다. 웹 백도어의 제작자가 흔히 사용하는 특징적인 문자열은 'hack', 'cmd', 'webshell' 등 이다. 최근 국내에는 중국발 해킹이 자주 발생하고 있기 때문에 중국어 간체 사용 여부(gb2312)를 확인하는 것도 좋은 패턴으로 활용될 수 있다. 그리고 패턴 탐지를 우회하기 위해 사용되는 **인코딩 함수**들도 패턴으로 등록하여 주의를 기울일 필요가 있다. 시스

템 명령어 실행이나 인코딩 등의 함수들은 정상적인 경우에도 사용될 수 있으므로, 해당 패턴으로 검출된 결과는 주의깊게 살펴보아야 한다.

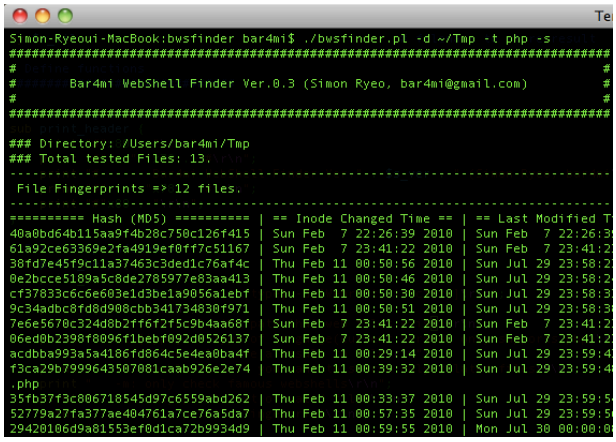
개발언어	일반적 패턴
PHP	• 명령어 실행을 위해 passthru(), system() 함수를 사용. • 패턴 매칭을 우회하기 위해 eval(), base64_decode() 함수를 이용.
ASP	• 명령어 실행을 위해 Wscript.Shell, Shell.Application 객체를 사용. • 패턴 매칭을 우회하기 위해 인코딩(VBScript.Encode)을 수행.
JSP	• 명령어 실행을 위해 Runtime.getRuntime 객체를 사용. • 패턴 매칭을 우회하기 위해 JavaScript를 이용하여 문자열 치환·분할

[표 2] 각 언어별 주요 함수 및 특징

잘 알려진 웹 백도어를 분석하여 특정 문자열을 추출하여 패턴으로 등록할 수 있다. 추출 가능한 문자열은 웹 백도어의 이름(c99shell, Cod3rz Shell, ASP木2006 등), 제작자 이름·닉네임, 특수한 파라메터명·변수명, 사용된 우회기법의 형태 등이다. 이때 일반적으로 사용되는 것을 오인하여 등록하지 않도록 주의해야 한다.

BWSFinder의 소개 및 이용방법

BWSFinder(Bar4mi WebShell Finder)는 perl을 기반으로 하여, 앞서 설명한 웹 백도어를 탐지하는 두 가지 방법을 모두 사용할 수 있는 **GPLv3 기반의 공개용 도구**이다. 이동성과 이식성을 고려하여 perl 언어로 제작되었기때문에 MS 윈도우즈를 비롯하여 UNIX 등의 시스템에서 범용적으로 사용할 수 있는 이점을 가지고 있다. 사용을 하기위해서는 시스템에 perl이 설치되어야 있어야 하지만, 시스템에 별도의 perl을 설치하기 어려운



[그림 1] 파일의 증적 출력

경우 Perl2Exe와 같은 컴파일러를 이용하여 운영체제별 실행파일을 생성하고 실행할 수 있다[6].

BWSFinder은 파일 목록을 생성하는 기능과 패턴 매칭을 통해 웹 백도어를 탐색하는 기능을 가지고 있다. BWSFinder는 다음과 같은 옵션을 제공하고 있다.

옵션	설명	비고
-d	웹 소스 파일이 존재하는 디렉터리를 명시하여 준다.	필수
-t	개발 언어의 종류(php, asp, jsp)를 명시한다.	필수
-r	결과값을 파일로 저장하고 싶을 경우, 파일명을 명시하여 준다.	
-e	개발 언어별로 관련된 확장자의 파일만 점검하고 싶은 경우, 이 옵션을 명시하여 준다.	
-m	잘 알려진 웹 백도어의 패턴만을 이용하여 점검을 하고 싶은 경우, 이 옵션을 명시하여 준다.	
-l	패턴 매칭 점검을 하지 않고, 검사하게 될 파일 목록들을 보고 싶은 경우, 이 옵션을 명시하여 준다.	
-s	파일 목록의 해쉬값, inode 변경시간, 수정시간 등을 생성하고자하는 경우, 이 옵션을 명시하여 준다.	

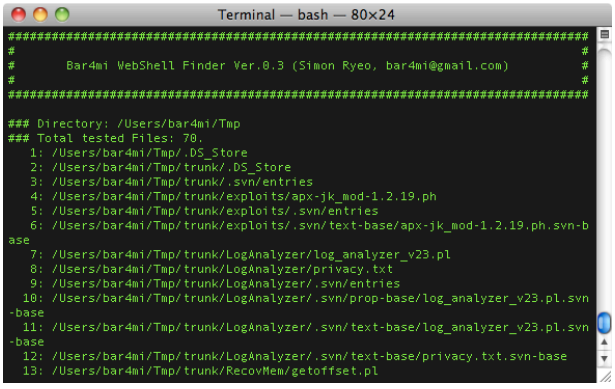
[표 3] BWSFinder의 옵션

BWSFinder를 이용하여 웹 디렉터리의 파일 목록을 생성하고자 하는 경우 아래와 같이 실행할 수 있다. BWSFinder는 [디렉터리]에 존재하는 이미지 파일, 문서 파일, 압축파일 등과 같은 불필요한 파일을 제외한 파일들에 대해 해쉬값(MD5), 생성시간(inode 변경시간), 마지막 수정시간 등을 나열한다. 만약, 개발 언어가 PHP일 경우 해당 개발 언어와 연관된 파일 확장자(php, phtml, php3, php4, inc 등)에 대해서만 점검을 하고 싶다면, '-e' 옵션을 추가하여 실행하면 된다. 수행 결과를 저장하고 싶다면 '-r [저장할 파일명]' 을 입력하면 된다.(BWSFinder의 모든 결과는 -r 옵션을 통해 지정된 파일로 저장할 수 있다.)

```
./bwsfinder.pl -d [디렉터리] -t [타입] -s
```

BWSFinder를 이용하여 점검하고자 하는 디렉터리에서 점검하게 될 파일 목록을 열람하고 싶다면 아래와 같이 실행할 수 있다. 패턴 검증을 하지 않으며, 현재 옵션 상태에서 검사하게 될 파일 목록을 출력한다. 만약, 개발 언어와 연관된 파일 목록만을 보고 싶다면 '-e' 옵션을 활성화한다.

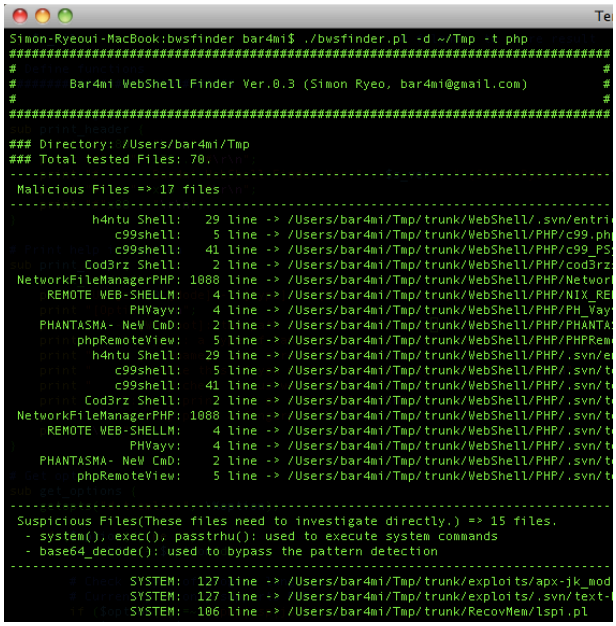
```
./bwsfinder.pl -d [디렉터리] -t [타입] -l
```



[그림 2] 현재 옵션에서 검사할 파일 목록을 출력

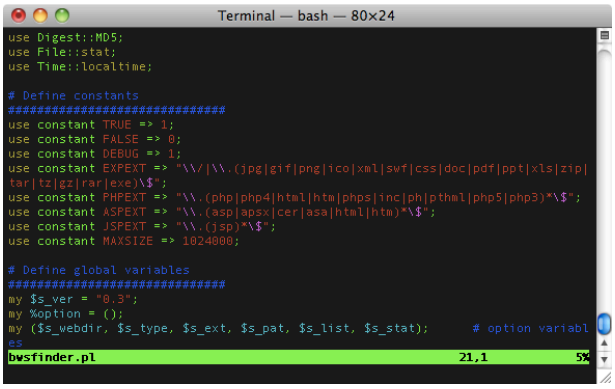
BWSFinder를 이용하여 전체 패턴 검증을 실시하는 경우, 아래과 같이 실행할 수 있다. 디렉터리와 개발 언어 이외의 특별한 옵션을 주지 않을 경우 기본값은 의심스러운 패턴을 사용하고 있는 파일과 잘 알려진 패턴이 포함된 파일에 대해 점검을 하고, 그 결과를 출력한다. 파일의 개수가 많을 경우 전체 패턴 매칭은 패턴 매칭에 시간을 많이 소비하기 때문에 시간이 오래 걸릴 수 있다. 잘 알려진 웹 백도어 패턴에 대해서만 점검을 하고자 하는 경우는 '-e' 옵션을 사용하여 빠르게 점검할 수 있다. 수행결과는 점검한 전체 파일의 개수, 의심스러운 패턴과 잘 알려진 패턴으로 구분하여 해당 패턴이 검색된 원인과 패턴이 검색된 소스 내 라인의 위치, 그리고 파일명을 출력한다.

```
./bwsfinder.pl -d [디렉터리] -t [타입]
```



[그림 3] 전체 패턴 매칭을 실시한 결과

BWSFinder는 소스와 패턴 DB 목록이 함께 제공되기 때문에 점검자가 기능을 자유롭게 조정할 수 있다. 만약 '-e' 옵션으로 관련된 파일 확장자 또는 기본으로 예외 설정하는 파일 확장자를 추가하거나 제거하고자 하는 경우 소스 파일에서 상수(constant)를 간단하게 수정할 수 있다. 파일 확장자와 웹 백도어의 패턴은 정규 표현식(Regular Expression)으로 정의되어 있다. 따라서 정규표현식에 맞추어 해당 패턴들을 정의하면 된다.



[그림 4] BWSFinder의 소스 설정

BWSFinder는 다양한 기능과 상황을 고려하여 제작하였다. 이미지 등과 같이 파일 크기가 큰 파일을 점검하는 것을 피하기 위해 1MB 이상의 파일은 점검에서 제외되어 있다.(일반적으로 소스 파일의 경우, 1MB를 초과할 이유가 없다고 생각한다.) 제작자가 생각하는 한계는 다음과 같은 것들이 있다.

• 현재 등록한 패턴은 도구 제작 및 개인적 사정으로 인해 충분한 패턴양을 갖추고 있지 못하다. 현재 적용되어 있는 다양한 우회 패턴을 반영하고 있지 못하다.
• 패턴 매칭의 한계는 오탐이 발생할 수 있는 것이며, 이에 대해서는 점검자의 직접적인 확인이 반드시 필요하다.(본 도구는 효율성을 높이기 위한 도구일뿐이며, 완벽한 솔루션을 제공하는 것은 아니다.)
• Apache 환경에서 발생하는 AddType[7], 자체적인 파일 확장자 사용 등과 같이 상황으로 인해 다양한 파일 확장자(php.bak, php.jpg, 등)가 웹 백도어에 사용될 수 이다. 이러한 상황에서 관련된 파일 확장자만 검색(e 옵션)하는 것은 올바른 검색을 실시하지 못한다. 현재의 파일 확장자는 파일 확장자가 늘어남에 따라서 검색 시간이 늘어나는 것을 최소화하기 위한 것으로, 점검자의 상황에 따라서 변경하여 점검하는 것이 바람직하다.

참고자료

[1] 웹쉘탐지프로그램(WHISTL), KISA, http://toolbox.krcert.or.kr/MMVF/MMVFView_V.aspx?MENU_CODE=6&PAGE_NUMBER=15
[2] ShellClean, (주)유엠브이기술, http://www.umv.co.kr/ShellClean
[3] Webshell Finder(WSF), Geekslab, http://wglo.egloos.com/category/%2BWebshell%20Finder
[4] “ASP 웹쉘 상세 분석 및 탐지방안”, 인터넷 침해사고 동향 및 분석 월보(2008.05), KISA
[5] “Webshell 탐지기법”, Geekslab, http://wglo.egloos.com/4493450
[6] Perl2Exe, Indigostar, http://www.indigostar.com/perl2exe.php
[7] “Common Apache Misconception”, SANS diary, http://isc.sans.org/diary.html?storyid=6139