Maintaining Credible IIS Log Files

By Mark Burnett
Last updated November 13, 2002

번 역 : blksaint@wowhacker.org (wowcode at wowhacker team)
번역일 : 2003-08-09
참 고 : 원문을 충실히 따라서 번역을 하였습니다만, 문장을 깔끔하게 처리하기 위하여 의역한 부분이 있으니 원문의 내용이 더 정확함을 알려드립니다. 오타나 틀린 내용이 있으면 알려주세요.

최근까지 많은 네트워크 관리자는 결국은 법적인 행동을 야기시키는 심각한 웹 서버 침입에 직면해 왔다. 종종 IIS 로그 파일은 웹 침입자의 자취를 찾아 보는 첫번째 증거로 사용되었다. 하지만 만약에 당신의 IIS 로그의 신빙성이 법정에서 무능력하게 된다면 어떤 일이 벌어질까? 만약에 피고측이 로그가 증거로써 충분히 용인될 수 있는지를 주장하게 된다면 어떻게 될까?

나는 범죄 수사의 일부분으로 심각한 침입에 대해서 조사한 적이 있다. 침입자는 IIS 서버를 무력화 시키고, 어떤 툴들을 업 로드하고 나서 그 회사의 내부 데이터베이스에 접속하였다. 우리는 언제 침입이 일어났는지 대략적으로 알게 되었지만, 수많은 서버에서 몇 백 개의 웹 사이트 중 어떤 것이 위협에 직면했는지는 알지 못했다.

나는 웹 서버에 저장된 수백개의 로그 파일을 파헤치면서, 수천개의 로그 엔트리를 가진 하나의 로그 파일에 도달했는데, 그 파일엔 빈 줄이 하나 있었다. 나는 그 파일의 최종 수정 시간을 알아 보았는데, 바로 그 로그 파일이 닫히기(closed) 이틀 전에 최종 수정되었다는 것을 발견했다. 수백 메가 바이트의 로그 파일 증거가 갑자기 그 빈 줄 하나 때문에 무용지물이 되었다. 왜냐하면 로그파일은 위험에 직면한 같은 서버상에 저장되었고, 침입자는 쉽사리 증거를 지울 수 있었을 것이고, 최악의 경우 어떤 다른 것을 가리키는 잘못된 증거로 바꿀 수도 있었을 것이기 때문이다. 하나의 로그 파일에 대한 변경은 그 서버의 모든 로그 파일들의 타당성에 대해 어쩔 수 없는 의문을 생기게 한다.

당신의 로그 파일이 신뢰할 수 있다는 것을 증명하기 위해서는 로그 파일이 신뢰할 수 있다는 설득력 있는 인자와 증거로써 유효하다는 것을 제공할 필요가 있다. 당신은 당신의 로그 파일의 정확성(accuracy), 확실성(authenticity:출처가 확실함)과 접근성을 보호하기 위하여 측정을 수행해야만 한다. 비록 수많은 법적인 복잡성도 있고 이러한 경우들에 있어서 당신 자신의 법적인 의견을 항상 찾아 봐야 하지만, 이제부터 소개하는 몇 가지 팁은 당신의 IIS 로그들의 신뢰성을 증가시켜 줄 것이다.

Log File Accuracy

정확성은 당신의 로그 파일 데이터가 당신의 웹 서버에서 일어난 행위들을 진실되게 표현한다는 것을 증명할 수 있다는 것을 의미한다. 아주 작은 부정확성이라도 전체 데이터 셋의 유효성에 의문을 야기시킬 수 있다. 다음 단계는 당신의 데이터가 정확하다는 것을 보증하는데 도움을 준다.

Log Everything - 당신의 IIS가 모든 유효한 필드를 기록하도록 설정하라. 어떤 관리자는 여분의 정보를 저장할 때 몇몇 값만을 살펴보지만, 모든 필드는 포렌식 조사 때 중요성을 가진다. 나는 한번은 IIS 침입 후에 어떤 로그들을 조사해 보았고, 공격자가 C: \ WINNT 디렉터리에 파일을 생성했다는 것을 알려주는 활동을 보았다. 하지만 나는 하드 디스크 어디에서도 그러한 파일들을 찾을 수 없었다. 로그 파일을 더 자세히 살펴봄으로써, 로그 파일에 기록된 이름과 서버의 이름이 일치하지 않는다는 것을 알아 차렸다. 최근에 회사가 웹사이트를 새로운 서버로 이전했다는 것이 밝혀졌다. 이전 서버를 살펴 봄으로써, 즉각 문제의 파일을 찾았다. 로그에 기록된 호스트 네임이 없었다면, 나는 실마리를 풀지 못했을 것이다.

또한 웹 방문자에 대한 정보를 수집하는 것도 공격이 어떤 특정한 컴퓨터 시스템 혹은 사용자로부터 왔다는 것을 아는데 도움을 준다. 예를 들어, 피고측이 해커가 그의 컴퓨터에 침투하여 백도어 프록시를 설치하고 나서 백도어 프록시를 이용하여 다른 시스템을 공격했다고 주장한다고 가정해 보자. 어떤 방법으로 트래픽이 특정한 사용자의 웹 브라우저로부터 왔는지 혹은 다른 사람으로부터 프록시 된 공격이라고 증명할 수 있을까? 이것이 항상 증명되는 것은 아니지만, 당신이 더 많은 정보를 수집하면 할수록 이러한 문제를 해결할 기회가 더 많이 주어진다.

Keeping Time - 당신의 IIS 서버를 윈도우즈 타임 서비스(Windows Time Service)를 이용하여 외부 시간 출처(time source)와 동기화 하라. 만약 당신이 도메인을 사용한다면, 타임 서비스는 자동적으로 도메인 컨트롤러와 동기화를 할 것이다. 독립형 서버에서는 다음 레지스트리 엔트리를 설정함으로써 외부 출처와 동기화 할 수 있다.

Key: HKLM \ SYSTEM \ CurrentControlSet|Services \ W32Time \ Parameters \
Setting: Type
Type: REG_SZ
Value: NTP

Key: HKLM \ SYSTEM \ CurrentControlSet|Services \ W32Time \ Parameters \
Setting: NtpServer
Type: REG_SZ
Value: tock.usno.navy.mil (public NTP 서버들의 목록은 http://tycho.usno.navy.mil/ntp.html를 참고하라.)

Key: HKLM \ SYSTEM \ CurrentControlSet|Services \ W32Time \ Parameters \
Setting: Period
Type: REG_SZ
Value: 24 (하루동안 동기화 할 횟수를 나타낸다. 24는 하루에 24번 동기화 한다는 것을 의미한다.)

또 하나의 시간에 관련된 이슈는 IIS는 UTC 타임을 사용하여 기록한다는 것이다. 이것은 서버를 여러 개의 타임 존에서 운용할 때 동기화 이슈에 도움을 준다. 하지만 윈도우즈는 UTC 타임을 그 시스템 타임 존을 가지고 시스템 시간 값을 오프셋 함으로써 계산한다. UTC 타임이 정확하다는 것을 보장하기 위해서는 지역 타임 존 설정이 정확하다는 것을 보장해야만 한다.

** UTC Universal Time Coordinated 협정 세계시(時)
** 한국표준과학연구원 URL : http://www.kriss.re.kr/참고

검증할 수 있는 하나의 트릭은 IIS가 지역 타임을 사용하여 로그 하도록 설정되었다는 것이다. 당신은 나중에 로그 파일에서 첫번째 엔트리를 살펴 봄으로써 서버의 타임 존 설정을 검증할 수 있다. 만약 당신의 서버가 UTC -0600으로 설정되었다고 하자, 그러면 첫번째 로그 엔트리는 18:00(00:00 - 06:00 = 18:00)이 되어야 한다. 왜냐하면 UTC는 daylight saving(Daylight Saving Time: 일광시간 절약 시간제, 섬머 타임)을 따르지 않기 때문에, 당신은 이것도 고려해야만 한다. 예를 들어, UTC -6:00은 실제로 일년의 반은 -5:00이 될 것이다.

Use Multiple Sensors - 만약 두개의 분리된 장치가 동일한 정보를 기록하게 되면 로그 엔트리에 반박을 하기가 힘들어진다. 여러 개의 장치로부터 로그를 합침으로써 당신은 각각의 가치를 강화시킬 수 있다. 방화벽 로그, IDS 로그 그리고 심지어는 TCPDump 만큼 단순한 어떤 것도 특정 IP 어드레스가 특정한 시간에 특정한 서버를 사용했다는 것을 증명하는데 도움을 줄 수 있다. IIS 로그를 보충하기 위해서 Snort를 사용하는 예를 http://www.iissecurity.net/4361.htm에서 살펴보라.

Avoid Missing Logs - IIS 로그의 한가지 문제점은 만약 서버가 24시간동안 한번도 접속이 이루어지지 않으면 어떠한 로그 파일도 생성되지 않는다는 것이다. 하지만 어떤 로그 파일도 존재하지 않으면, 해당 서버가 실제로 접속이 없었는지(하루 동안 오프라인이라고 말한다.) 또는 로그가 실제로는 삭제되었는지 구분할 방법이 없다. 이러한 문제를 해결하기 위하여, 나는 항상 로그가 존재하도록 매일 몇 번의 접속을 예약해둔다(schedule).

그렇게 하기 위해서 나는 Graburl을 사용하는데 http://www.kiraly.com/software/utilities/graburl/에서 다운로드 할 수 있다.

태스크 스케줄러를 사용해서, 나는 웹 서버에 두개의 접속을 예약해 둔다: 하나는 localhost로부터 이고, 다른 하나는 외부 호스트로부터 이다. 명령어는 단순하다.

Graburl.exe www.example.com

두개의 접속을 예약하는 이유는 첫번째 localhost로부터의 접속은 서버가 운영되고 있다는 것을 검증할 수 있고, 두 번째는 인터넷에서 접속이 가능하다는 것을 검증해 주기 때문이다. 더욱이 두 번째 접속은 또한 시간 동기화를 검증해 준다. 만약 두 번째 접속이 매일 1:00AM에 일어나도록 예약되었다면, 이에 상응하는 로그 엔트리는 항상 1:00AM에 발생해야만 한다. 일반적으로 예약된 요청은 로깅 매카니즘이 알맞게 기능을 수행한다는 것을 증명하는데 도움을 준다.

만약 웹 서버가 24시간 이상 전원이 나간다면, 어떠한 로그 파일도 생성되지 않을 것이다. 하지만 당신의 EventLog는 서버가 전원이 나갔다는 것을 알려줄 것이다. 이러한 사실에 의해서 로그 파일이 없다면, 아마도 고의적으로 삭제되었기 때문일 것이다.

Log File Authenticity

만약 로그 파일이 원래 기록된 후에 변경이 없다는 것이 증명이 되면 출처가 확실하다고 할 수 있다. IIS 로그 파일은 단순한 텍스트 파일이라서 쉽게 변경이 가능하다. 파일의 날짜와 시간도 역시 쉽게 변경될 수 있다. 로그 파일의 초기 상태 그대로는 IIS 로그 파일은 출처가 확실하다고 할 수 없다. 하지만 다음의 몇 개의 팁이 당신을 구제할 수 있다.

Move the Logs - 확실성 보장을 시작하기 위해서, IIS 로그를 웹 서버 외부로 이동시켜라. 만약에 서버가 위험에 처하게 되면 당신은 로그 파일 역시 위험에 처할 수 있다는 것을 고려해야만 한다. 로그 파일을 주 서버로 이동시키고 나서 최대한 빨리 테잎, CD 또는 WORM 장치 같은 offline 매체로 이동시켜야 한다.

Signatures, Encryption & Checksums - 로그 파일이 수정되지 않았다는 것을 완벽하기 하기 위한 수단은 서명을 하고 PGP나 다른 공개키 암호화 방법을 사용하여 로그를 암호화하는 것이다. 파일 서명은 만약 하나의 파일이 잘못되어도 다른 나머지 로그들을 무효화하지 않기 때문에 도움이 된다. 당신은 또한 Fsum 같은 툴을 사용하여 신속하게 파일에 대한 MD5 해쉬를 생성할 수도 있다. 서명과 해쉬를 파일과 함께 저장하고 다른 장소에 보안 복사본도 저장하라.

만약에 로그 파일 서명을 위한 자동화된 프로세스를 사용한다면, 당신은 항상 신뢰할 수 있는 관리자에 의해 수동으로 서명된 것을 따라야 한다는 것을 명심해야 한다.

파일을 암호화할 때, 당신은 생성, 수정 그리고 접근 날짜에 끼치는 영향을 고려해야만 한다. 당신은 Fdir같은 유틸리티를 사용함으로써 이러한 날짜들을 다른 위치에 기록하기를 원할지도 모른다.

Work With Copies - 어떤 로그 파일 분석을 할 때에는 절대로 원본 파일로 작업해서는 안 된다. 어떤 후처리나 로그 파일 분석을 행하기 이전에 복사본을 만들어야 한다. 원본 로그는 여전히 출처가 명확하고 원본 형식을 그대로 유지한다는 것을 보장하는데 도움을 주기 위해서 절대로 건드리면 안 된다. 만약 당신이 로그 파일을 법적 증거로 제시한다면, 당신은 원본 형식을 가진 원본 파일을 제공해야만 한다. 미국에서는 증거에 대한 연방법률(Federal Rules of Evidence)을 보면 정확한 출력물 또한 원본 증거로 고려될 수 있다(Federal Rules of Evidence 1001(3)을 보라).

Ensure System Integrity - 당신은 당신의 시스템 파일이 유효하다는 것을 보장하기 위해서 항상 최신의 서비스 팩과 hotfix를 유지해야만 한다. 또한 당신은 당신의 WINNT 디렉터리의 모든 바이너리 파일의 변화에 대한 감사를 수행해야 한다. 만약 침입자가 로그 파일을 기록하는 시스템 파일을 수정할 수 있다면, 증거로써 로그 파일의 사용에 대해서는 의문이 생기게 된다.

Have a Process - 잘 수립되고 문서화된 프로세스는 실제로 출처의 확실성을 공고히 하는데 도움을 준다는 것을 명심하라. 일관된 결과를 도출하는 잘 수립된 절차는 파일이 유효하고, 확실하다는 것을 공고히 하는 데 도움을 줄 것이다. 더욱이, 부가적인 증거(공격자의 IP 어드레스에 대한 네트워크 진단 utility의 사용 같은)를 수집하기 위하여 문서화되고 일관된 방법을 가져야 한다. 왜냐하면 소송 전에(anticipation of litigation) 만들어진 사업적인 기록이 항상 법정에서 용인될 수 있는 것은 아니기 때문이다. 이것은 특히 만약 법 집행자가 부가적인 증거를 모으기 위하여 (적절한 법 명령 없이) 당신의 네트워크에 sniffer와 같은 툴을 사용하도록 지시한다면 사실이다.

프로세스를 공고히 하는 것은 각각의 매뉴얼 또는 자동화된 수행 단계의 윤곽을 잡는 문서를 만드는 것을 의미한다. 더욱이 로그 파일 처리에 사용하는 어떠한 스크립트도 정확히 어떠한 처리가 행해지고 있는 지를 설명하는 주석문을 포함해야만 한다. 당신이 당신의 프로세스에 사용하는 기술은 일반적으로 로그 파일 관리에 사용하는 절차로써 받아들여지는 것이어야만 한다.

Access Control

로그 파일이 한번 생성되면, 파일의 접속을 방지하고 인가된 접속과 비인가된 접속을 감시하는 것이 중요하다. 만약 당신이 NTFS 퍼미션을 사용하여 로그 파일을 안전하게 하고 감사한다면, 로그 파일의 신빙성을 공고히 하는데 도움을 주는 문서화된 증거를 가지게 될 것이다.

Restrict File Access - 로그 파일은 IIS가 파일에 기록할 수 있도록 어떤 퍼미션을 필요로 한다. 하지만 로그 파일이 닫히고 난후에는 아무도 파일 내용을 수정할 수 없어야 한다. 당신은 로그 파일이 닫힌 후

에 파일 퍼미션과 감사를 막아버리는 명령의 수행을 예약하는 것을 고려할지도 모른다. 또한 당신이 로그 파일을 이동시킬 때, NTFS 퍼미션이 새로운 위치에서도 정확하게 설정되는 지를 보장해야 한다.

Chain of Custody - 당신이 서버로부터 로그 파일을 옮기고, 나중에 offline 장치에 옮김에 따라, 당신은 파일이 어디로 흘러가는지를 추적해야 한다. 이러한 추적은 기술인 방법 혹은 기술적이지 않은 방법으로 할 수 있다. 예를 들어, 내 고객 중에 하나는 그들의 백업 테잎을 봉합하고 테잎의 물리적인 이동을 기록하기 위하여 라벨을 사용한다. 증거를 관리하는 것을 추적하는 것은 범죄 수사 시에 백업으로부터 내용물을 추출할 때 특별히 중요하다.

매일 로그를 수집하는 프로세스는 언젠가 범죄 수사에 증거를 수집하는데 일부가 될 수 있다는 것을 명심하라. 당신은 항상 마치 이미 범죄가 이루어지고 있는 것처럼 당신의 IIS 웹 서버를 다루어야 하고, 로그가 이미 증거인 것처럼 로그 파일을 취급해야 한다.

비록 많은 법과 사법권이 존재하지만, 로그 파일은 종종 사업적인 기록과 동일하게 취급된다. 따라서 당신은 로그 파일의 허용성과 중요성을 보장하기 위한 어떤 룰을 따라야만 한다. 무엇이 용인될 수 있고, 무엇이 신뢰성이 있는지에 대한 명확한 정의는 언제나 존재하는 것은 아니다. 하지만 법정 증인으로써, 로그 파일은 종종 신뢰를 받는다. 그리고 더 문서화된 증거를 가지고 있을수록, 당신의 IIS 로그는 더 많은 신뢰를 가지게 된다.

물론 당신의 국부적인 증거와 사적인 법규가 다르고, 당신의 변호사가 로그 파일의 신뢰하게 만드는데 대해서 다른 생각을 가질지도 모른다. 이슈를 당신의 변호사와 토의해서 당신의 IIS 로그 파일의 정확성, 확실성 그리고 통제된 접속을 보장하는 프로세스를 공고히 하도록 하라.

References

Federal Rules of Evidence

Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations

Computer Records and the Federal Rules of Evidence