침해사고 대응 및 예방 프레임워크

이은영[0],김도환,박응기
국가보안기술연구소[0]
{eylee[0], dkim, ekpark}@etri.re.kr

A Framework for Preventing and Counter-measuring Computer Incidents

Eun Young Lee[0], Do-Hwan Kim, Eungki Park
National Security Research Institute

요 약

최근의 웜이나 바이러스로 인한 침해 사고는 다수의 불특정 시스템을 대상으로 하고 있으며 짧은 시간동안 다수의 시스템을 감염시킨다. 현재의 침해 사고의 처리는 시스템이 감염된 후 이를 처리하는 방식이나 이러한 방식으로는 짧은 시간 안에 급속도로 확산되는 침해사고를 막기가 힘들다. 본 논문에서는 침해 사고의 효과적인 처리를 위한 새로운 프레임 워크를 제안하고자 한다. 평소에 관리하고자 하는 서버들의 정보를 수집하고 관리함으로써 새로운 취약점이 발견되었을 때 또는 침해 사고가 발생 하였을 때 취약할 가능성이 있는 서버들을 신속히 파악하며, 취약점을 점검하는 도구를 생성하는데 필요한 공통 프레임 워크를 개발하여 보다 빠르게 취약점을 점검하는 도구를 생성, 이를 이용해 취약할 가능성이 있는 서버들을 점검한다. 점검결과 감염된 시스템은 치료하고 취약하나 감염이 되지 않은 시스템은 취약점을 제거한다. 제안된 프레임 워크는 평소에 서버들의 정보를 수집함으로써 침해사고가 발생 하였을 감염 가능성이 있는 서버들을 빠르게 확인 할 수 있다. 또한 취약점 점검 도구 생성 프래그램를 사용함으로써 점검 코드의 삽입만으로 쉽게 취약점 점검 도구를 생성, 확산되기 전에 타 시스템에서 동종의 침입을 예방할 수 있다.

1. 서 론

과거에는 특정 시스템을 대상으로 한 침해사고가 주를 이루었지만 근래에는 다수의 불특정 시스템을 공격 대상으로 삼는 인터넷 웜이나 바이러스가 기승을 부리고 있다. 이들 웜이나 바이러스는 확산 속도가 빨라 짧은 시간에 대다수의 시스템을 감염시킨다. 2002년도의 slammer worm 과 2003년도의 blaster worm의 예를 보면 발견 된지 1~2일 만에 전 세계로 확산되어 그 피해의 정도가 매우 컸음을 알 수 있다. blaster worm은 발생된 지 이틀째에 전 세계적으로 12만 4천여 대를 감염시켰으며 아직 까지도 그 여파가 남아있다[1]. 이러한 사실은 대부분의 침해 사고가 피해가 발생한 후에 이의 후처리에 중점을 두나 동종의 침입에 대한 예방이 이루어지지 않았음을 말해 주고 있다. 특히 규모가 큰 회사와 연구소는 다수의 시스템을 보유하고 있기 때문에 침해사고가 발생 하였을 때마다 모든 시스템을 일일이 점검하기가 어려우며, 많은 시간과 비용이 소요된다. 또한 침해 사고가 발생 하였을 때 점검도구를 생성하여 이용하는데 있어 제작 기간이 길기 때문에 급속히 확산되는 웜이나 바이러스를 예방하는 데는 한계가 있다.

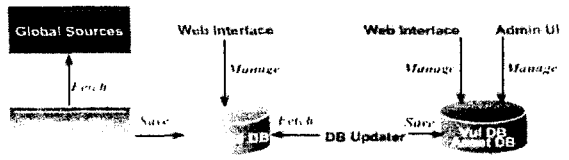
과거에 비해 침해사고의 확산 속도가 빠르고 그 피해 정도가 크기 때문에 침해사고의 예방이 더욱 중요시 되고 있다. 침해사고는 발생 후처리 하는것 보다는 미리 점검하여 동종의 침입을 예방하는 것이 중요하다. 침해사고의 확산 속도가 갈수록 빨라지고 있기 때문에 이를 진단하기 위한 취약점 진단도구 제작 기간을 단축하는 것이 필요하며 이를 이용하여 침해사고의 확산을 보다 효과적으로 막을 수 있다.

본 논문에서는 침해사고가 발생하였을 때 대응 및 예방을 위한 프레임 워크를 제안하고자 한다. 수시로 관리하는 시스템들의 정보를 수집함으로써 시스템들의 상태를 파악하고 새로운 취약점이 발표되거나, 침해사고가 발생 하였을 때는 취약한지를 점검하는 일차적 대상으로 사용한다. 침해 사고 시 서버들을 점검하고 상태를 파악하기 위해서 일차적 감염 가능성이 존재하는 서버들을 대상으로 스캐닝이 필수적이다. 이렇듯 진단 도구 생성에 필수적인 스캐닝 과정, 프레임 형성과정 등에 관한 공통적인 프레임워크를 개발하고 진단 코드를 작성하는데 필요한 공통 API를 제공함으로써 진단도구 제작 기간을 단축시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 연구와 비교분석하고 3장에서는 제안한 시스템에 대해 다루며 4장에서는 시스템의 설계를 설명하며 5장에서 결론으로 끝을 맺는다.

2. 관련 연구

본 장에서는 관련연구로 시큐리티맵(주)에서 개발한 실시간 취약점 관리시스템에 대해 살펴본다.[10] 실시간 취약점 관리 시스템은 취약점이 발견될 때마다 이를 데이터베이스에 저장하여 관리한다. 그리고 이와 함께 관리·보호해야할 시스템들의 자산 데이터베이스를 유지한다. 새로운 취약점이 발견되었을 때 이 취약점을 자산 데이터베이스와 비교를 하여 만약 관련된 자산이 있는 경우 관리자에게 이를 알려 취약점에 대비를 하도록 한다. 다음은 실시간 취약점 관리시스템의 구조이다.



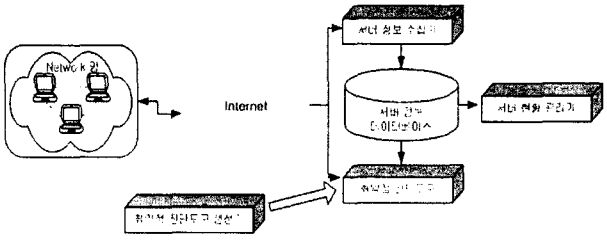
< 그림 1 >실시간 취약점관리시스템 구조

왼쪽은 취약점 데이터베이스에 입력을 담당하는 부분으로 시큐리티맵에서 관리한다. 새로운 취약점이 발견되었을 때 이를 데이터베이스에 저장 및 관리를 담당한다. 오른쪽 파란 사각형은 고객사에 위치한 시스템으로 취약점 데이터베이스와 자산 데이터베이스를 유지하고 있다. 새로운 취약점에 발견될 때마다 취약점 데이타베이스가 업데이트 되고 취약점의 영향을 받는 자산을 파악한다. 관련 있는 자산은 관리자에게 알려져 취약점을 대비하도록 한다.

3. 제안한 프레임워크

제안한 프레임 워크는 크게 둘로 나누어 볼 수 있다. 하나는 평상시 관리해야할 서버들의 정보를 수집/분석하여 데이터베이스에 저장하는 시스템이다. 즉, 관리하고자 하는 시스템의 정보를 유지함으로써 새로운 취약점이 발표되거나, 해킹을 포함한 악의적인 공격이 발생하였을 때 공격 가능성을 지닌 대상을 미리 파악하고 대처함으로써 공격을 미연에 대처할 수 있다. 다른 하나는 침해사고에 대한 취약점 진단을 빠르게 수행할 수 있도록 침해사고 발생시 취약점 진단 코드만을 삽입함으로써 수집된 서버정보를 토대로 취약점 분석을 수행하는 점검 도구를 빠르게 생성하는 시스템이다. 즉, 취약점 점검 코드의 삽입만으로 수집된 시스템의 정보를 대상으로 취약점 점검을 수행하는 도구를 자동으로 생성하는 것이다.

다음 그림은 제안한 프레임워크에 따른 시스템의 구성도이다.



< 그림 2> 침해사고 대응 및 예방 프레임워크 구성도

- 서버정보 수집기 : 정해진 스케줄에 따라 특정 대역의 서버정보를 수집하며 지정된 규격으로 데이터베이스에 저장한다.
- 서버현황 관리기 : 관리해야할 시스템을 등록하며 정보수집의 스케줄을 설정한다. 데이터베이스에 저장된 서버정보를 관리하며 정보에 대한 검색/수정/삭제 등이 가능하다.
- 데이터베이스 : 수집된 서버정보를 저장한다. 수집된 서버정보는 분석 및 1차 취약점 진단의 기본 데이터로 활용한다.
- 취약점 진단도구 : 특정한 공격에 대해 취약점을 진단할 수 있는 도구로 데이터베이스 검색을 통한 1차 취약점 진단에서 검색된 서버를 대상으로 실질적인 취약점 진단을 수행 한다.
- 취약점 진단도구 생성기 : 특정 취약점을 점검하는 코드만을 삽입한 후 컴파일하면 취약점 진단도구를 생성한다.

침해사고 대응 및 예방 프레임워크는 다음과 같이 동작한다. 관리하고자 하는 네트워크 망내 시스템의 정보를 서버 현황 관리기에 등록 하고 스케쥴링 한다. 서버 정보 수집기는 스케쥴링에 따라 시스템의 정보를 수집하고 데이터베이스에 저장한다. 새로운 취약점이 발견되거나 침해 사고 발생하였을 때 취약점 점검 코드를 삽입하여 취약점 진단도구 생성기가 취약점 점검도구를 생성한다. 생성된 취약점 점검도구는 데이터베이스에 저장된 시스템 중에서 감염이나 공격의 대상이 될 가능성이 있는 서버들을 대상으로 취약점 점검 코드를 이용하여 시스템을 점검한다. 점검결과에 따라 조치를 취하도록 서버관리자에게 통보한다.

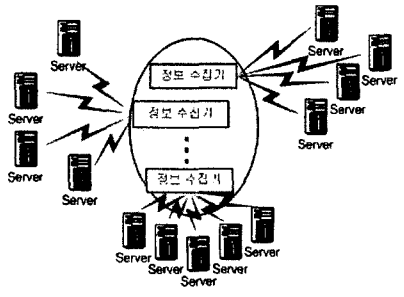
침해사고 대응 및 예방 프레임워크는 관리해야할 자산에 대한 현황을 미리 파악하고 있다. 따라서 새로운 취약점이 발견되었을 때 바로 취약 가능성이 있는 시스템을 파악할 수 있으며 즉시 조치가 가능하다. 또한 적절한 조치를 취하였는지를 검사할 수도 있다. 또한 침해사고가 발생하였을 때에도 바로 조치가 가능하여 동종의 침입에 대해 빠르게 대응할 수 있다.

4. 시스템 설계

4.1 서버정보 수집기[2]

서버 현황 관리기에서 정해진 스케줄에 따라 서버들의 정보를 수집한다. 기본적으로 운영체제와 포트의 open여부를 점검하며 대표적인 서비스들에 대한 banner정보를 수집하여 데이터베이스에 저장한다.

스캔을 통한 정보 수집은 네트워크 대역에 심각한 부하를 초래하지 않아야 한다.[6-9]한 단위 네트워크에 너무 많은 패킷이 전달되면 안 되므로 분산화된 스캔 과정이 필요하다. 즉, 각 서버들의 작동하는 네트워크의 부하를 줄이도록 분산화된 스캔을 수행하여야 한다.[2][3][4]



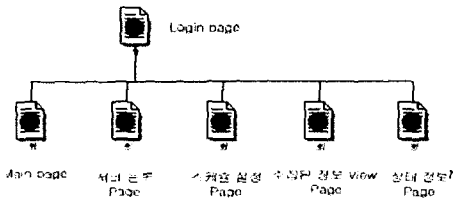
<그림 3> 분산화된 서버정보 수집

4.2 서버 현황 관리기 및 데이터베이스

서버 정보 수집기에 의해 수집된 정보들은 데이터베이스에 저장되며 새로운 취약점이 발견되거나 침해사고가 발생 하였을 때 감염 가능성이 있는 서버를 선별하는 1차 정보로 이용된다. 이와 함께 평소에 서버들이 지원하는 서비스나 네트워크 상황을 파악함으로써 보안정책의 기준으로 삼을 수 있다.

서버 현황 관리기는 서버들의 등록/삭제/수정 기능을 가지며 등록된 서버들의 스케줄링을 설정한다. 또한 수집된 서버들의 정보들을 보여주며 이들의 통계작업을 수행함으로써 관리자가 관리하는 서버들의 현황을 파악할 수 있도록 도와준다.

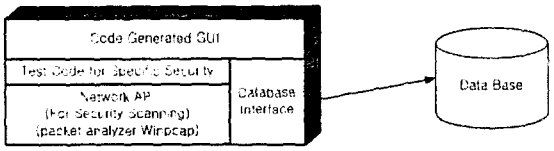
이와 함께 서버 관리자 정보를 연계하여 관리함으로써 침해 사고 발생시 관리자에게 침해사실의 발생을 알림으로써 동종의 침입을 예방한다.



<그림 4> 현황 관리 사이트 구조

4.3 취약점 진단 도구[3][4]

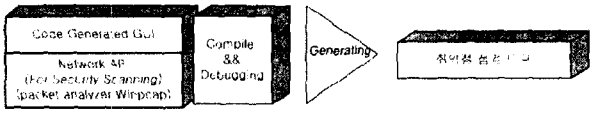
취약점 진단도구는 취약점 진단도구 생성기로부터 생성된 도구로써 특정 취약점에 대한 진단을 수행 하는 도구이다. 일차적으로 운영체제와 포트를 기준으로 취약 가능성이 있는 서버들을 일차적으로 검색하고 이들을 대상으로 취약점 여부를 수행한다. 즉 데이터베이스에 연결하여 수집된 서버정보를 바탕으로 os, 서비스 포트, 배너 정보를 기준으로 취약 가능한 서버 리스트를 산출하며, 각각의 서버에 대해 취약점에 대한 취약 여부를 판단한다.



<그림 5> 취약점 점검 도구 구조

4.4 취약점 진단 도구 생성기

취약점을 점검하는데 공통이 되는 스캐닝 과정, 프레임 형성과정에 관한 공통 적인 프레임워크를 개발 하여 특정 취약점을 점검 하는 코드만을 삽입 컴파일 함으로서 취약점 진단도구를 생성하는 도구이다.



<그림 6> 취약점 진단 도구 생성기 구조

취약점 진단 코드 작성 시 공통적으로 필요한 모듈로는 통신 모듈, 분석 모듈 등이 포함된다. 이러한 모듈들을 API 함수로 제공함으로써 진단 코드의 작성을 돕는다.

5. 결론 및 향후 연구 방향

현재 대부분의 침해 사고 대응은 침해사고가 발생한 이후 이를 처리하는데 초점을 두고 있다. 하지만 이러한 방식은 불특정 다수를 대상으로 하고 자동화, 분산화된 공격으로 짧은 시간 안에 전 세계로 확산되는 침해사고에는 초기 대응이 어렵다. 제안된 프레임워크는 관리 대상 서버들의 정보를 수집, 관리함으로써 평소에 이들 서버들의 현황을 파악하여 보안 지침을 제공할 수 있으며 새로운 취약점이 발견되거나 침해 사고가 발생하였을 때는 취약점에 대한 진단 도구를 빠르게 생성하여 관리하는 시스템들을 점검함으로써 침해 사고에 대한 빠른 대응이 가능하게 한다.

향후 연구과정으로는 제안된 시스템에 취약점 점검을 통해 침해를 당했다고 밝혀진 서버들을 복구하고 치료하는 대응 방법을 융합함으로써 예방과 대응이 가능한 침해사고 처리 방안의 연구가 필요하다.

참고문헌

[1] 서울 연합뉴스, 이정재 , 2003.08.12
[2] nmap homepage http://www.nmap.org
[3] nessus homepage http://www.nessus.org
[4] Security Administrator's integrated Network Tool http://www.wwdsi.com
[5] Internet Security Systems http://www.iis.net
[6] Metta Security Limited, IP Network Scanning & REconnaissance, 2002
[7] Fyodor, The Art of Port Scanning
[8] Joel Scambray, Hacking Exposed 2nd
[9] Examing port scan methods - Analysing Audible Techniques
[10] 실시간 취약성 관리시스템, www.securitymap.com