System profile과 Attack bucket을 이용한 침입시도정보 필터링

장명근[01], 이은영[2], 이상훈[2], 박응기[2], 채송화[1], 김동규[1]
[1]아주대학교 정보 통신 전문 대학원, [2]국가보안기술연구소
jinmi80[0]@hotmail.com, {eylee, melsh, ekpark}@etri.re.kr, {portular, dkkim}@ajou.ac.kr

Intrusion Alert Filtering Using System Profile and Attack Bucket

Myoung-Geun Jang[01], Eun-Young Lee[2], Sang-Hun Lee[2], Eung-Ki Park[2], Song-Hwa Chae[1], Dong-Kyoo Kim[1]
[1]GSIC AJOU University, [2]NSRI(National Security Research Institute)

요 약

인터넷상에서 해킹도구들을 구할수 있게 되고 이러한 정보들이 쉽고 빠르게 전파됨에 따라 쉽게 해킹을 시도할수 있게 되었고 이로인해 침입시도의 수가 급증하고 있다. 그결과 침입탐지시스템(Intrusion Detection System, IDS)에서 발생하는 침입시도정보의 수도 늘어나고 있다. 또한 이렇게 생성되는 많은 침입시도정보들에서 긍정오류(false positive)와 같은 잘못된 침입시도정보들이 큰 문제이다. 침입으로 오인된 정보가 너무 많음으로 인해 네트워크 관리자가 정확하게 판단을 하여 대응하는데 많은 노력이 요구된다. 이러한 노력을 줄여주기 위하여 긍정오류와 반복되는 침입시도정보를 줄여주는 기법이 필요하다. 본 논문에서는 이러한 필터링 시스템을 제안한다. 시스템 정보를 이용하여 위험이 될수 없는 공격을 제거하여 관리자에게 정확한 정보를 전달하고 동일한 공격들을 제거하여 침입시도정보의 수를 줄여주는 방법을 제안한다.

1. 서 론

최근 DDoS, worm등의 공격이 네트워크 시스템에 큰 위협이 되고 있다. 게다가 인터넷상에서 손쉽게 구할수 있는 해킹도구들은 다양한 형태의 공격들을 양산하는데 큰 역할을 한다. 이렇게 증가하는 공격시도들을 사전에 막기위해 많은 INFOSEC도구들을 네트워크에 설치하게 되었다. 하지만 다양한 보안도구들이 발생시키는 침입시도에 대한 정보의 수는 이미 관리자가 제어할 수 있는 범위를 넘어섰다. 더욱이 침입탐지 시스템의 경우에는 과도한 침입시도정보들 중에 많은 양이 긍정오류(false positive)라고 조사된바 있다. 이러한 긍정오류를 제거하기 위하여 IDS가 보다 세분화된 기준으로 공격을 탐지할 수 있지만 이로 인해 부정오류(false negative)가 증가하게 된다. 이러한 문제를 해결하기 위하여 IDS로부터 발생된 침입시도정보를 분석하여 보다 정확하고 명확한 탐지결과를 생성하기 위한 기술들이 연구되고 있다[1][2]. 하지만 IDS가 발생시킨 침입시도정보들을 그대로 이용할 경우 과도한양의 데이터로 인해 그 성능에 문제가 생긴다. 따라서 사전에 전체 시스템에서 고려하지 않아도 되는 침입시도 정보들을 제거할 필요가 있다.

침입시도정보 필터링 기술은 발생한 침입시도정보들중에 잘못된 침입시도정보들을 줄여주어 관리자나 다른 시스템들이 분석할 때 용이하게 만드는 역할을 한다. 본 논문에서는 이러한 필터링 시스템을 제안한다. 실제 시스템에 존재하지 않는 취약성에 대한 공격은 시스템에 아무 영향도 줄수 없다. 이러한 공격들에 대한 침입시도정보는 관리자에게는 부담일 뿐이다. 그러므로 현재 시스템의 정보를 이용하여 시스템에 영향을 주지 않는 침입시도정보들을 제거해야 하는 필터링이 요구된다. 또한, 공격에 따라 같은 공격이 과도하게 발생하여 생성되는 중복되는 침입시도정보들이 문제가 된다. 이러한 침입시도정보들은 공격자들에 의해 여러 번 발생하지만 모두 같은 작업을 나타낸다. 그러므로 이러한 중복된 침입시도정보들은 모두 하나의 침입시도정보로 병합하면 관리자가 판단하기에 큰 도움이 된다.

2. 관련연구

시스템 정보를 이용한 필터링은 호스트 정보와 호스트에서 동작하는 제품정보등의 현재 시스템의 정보에 따라 달라지는 시스템의 취약성정보를 이용하는 것이다.[3] 시스템에서 성공할수 없는 공격들을 제거하고 그 시스템에 존재하는 취약성을 공격하는 것들만을 보고하는 것이다.

토큰 버킷(token bucket) 필터는 데이터 흐름의 비율을 조절하는 알고리즘이다.[4] 토큰은 토큰 비율에 따라 생성되고 버킷에 저장된다. 만약 침입시도정보가 발생하는 비율이 버킷의 토큰 비율보다 작다면 이것은 저장되거나 관리자에게 보고된다. 하지만 침입시도정보가 발생하는 비율이 토큰 비율을 넘어간다면 이 침입시도정보는 저장되지 않고 버려지게 된다. 이러한 버킷필터는 시그네쳐(signature)나 공격타입에 따라 설정되거나 전체적으로 적용될수 있다.

3. 제안하는 시스템

IDS에서 발생된 침입시도정보는 서론에서 제안한 시스템 정보를 이용한 필터링과 동일한 침입시도정보의 필터링을 이용하여 보다 정확한 정보가 되고 그 수가 줄어들게 된다. 이러한 필터링 기법을 적용하는데 있어서 우선 시스템 정보를 사용하는 필터링이 먼저 적용되고 다음에 동일 침입시도정보 필터링을 적용하는 방법이 필요하다. 시스템 정보를 이용한 필터링은 수많은 종류의 공격중에서 현재 시스템에 존재하는 취약성에 해당하는 공격들만을 남긴다. 반면에 동일 침입시도정보 필터링은 모든 공격마다 테이블을 생성하게 되므로 공격 종류의 수가 수행시간에 큰 영향을 미친다. 그러므로 [그림 1]과 같이 우선 공격의 종류와 수를 줄여주는 system profile을 이용한 필터링을 적용한후 동일 침입시도정보 필터링을 적용한다.

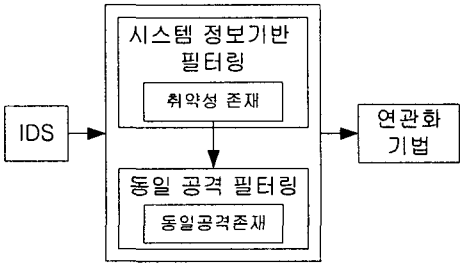

그림 1 필터링 시스템의 구조

3.1 시스템 정보를 이용한 필터링

시스템 정보는 네트워크를 구성하는 각각의 호스트들에 대한 정보와 그 호스트에서 동작하는 제품이나 제공하는 서비스로 구성된다. 각 호스트는 이름과 IP 주소로 정의된다. 제품정보는 각 호스트에서 동작하는 OS나 제품의 이름과 버전정보들로 구성된다. 서비스도 마찬가지로 호스트에서 제공되는 서비스 이름과 포트넘버등으로 구성된다. 취약성 정보는 데이터베이스로 따로 관리되는데 제품이나 서비스에서 나타나는 취약성들로 나타낸다. 이렇게 따로 관리함으로써 취약성정보의 갱신이나 system profile의 변경시 서로의 정보에는 상관없이 각각의 정보의 변경만을 수행하면 되는 장점이 있다.

필터링 방법은 필터에 침입시도정보가 들어오면 우선 침입시도 정보에 정의된 취약성을 찾아낸다. 그리고 취약성정보가 저장된 데이터베이스를 이용 어떠한 system profile에 영향을 끼치는지를 알아낸다. 침입시도정보의 목적지 주소를 통해 해당하는 호스트를 인지한후 호스트 정보를 통해 system profile에 해당하는지를 판단한다. 만약 system profile에 해당한다면 그 공격은 시스템의 취약성을 공격한 것이므로 관리자에게 보고되거나 연관기법등을 위해 따로 저장된다. 하지만 해당되는 사항이 없다면 이는 시스템에 영향을 끼치지 않는 공격이므로 따로 보고 하거나 저장하지 않는다. 하지만 어떠한 공격시도가 어느정도나 이루어지고 있는지는 관리자에게 중요한 정보가 된다. 그러므로 각 호스트별로 일정시간동안 발생한 침입시도정보들에서 공격타입만을 고려하여 카운트 하여 관리자에게 보고한다.

3.2 동일 침입시도정보 필터링

어떠한 시스템에 침입을 시도할 때 같은 공격을 여러 번 반복하거나 IDS자체에 대한 공격으로 많은 수의 같은 공격이 이루어진다. 또한, DDOS나 웜의 경우에 한 호스트에 동일한 공격이 엄청나게 발생한다. 이러한 공격들은 IDS로 하여금 많은 침입시도정보를 발생시켜서 관리자나 연관기법에서 처리하는데 많은 노력이 필요하게 한다.

[표 1]은 동일공격임을 판단하는데 쓰이는 테이블이다. 테이블은 공격이름과 소스주소, 목적지주소, 침입시도정보ID로 이루어진다. 공격이름은 각각의 침입시도정보가 나타내는 M2D2에 정의된 공격이름이고, 소스주소와 목적지 주소는 어떠한 곳에서 어떤 호스트에대한 공격인지를 나타낸다. 침입시도정보ID는 IDS로부터 발생된 침입시도정보들의 고유 ID이다.

표 1 동일 침입시도정보의 비교를 위해 생성되는 테이블

공격이름	소스주소	목적지주소	침입시도 정보ID
GainOsInfo	210.107.44.6	210.107.19.1	54786
GainOsInfo	218.122.5.10	210.107.19.1	55214

동일한 침입시도정보의 필터링은 취약성을 이용해 필터링된 침입시도정보가 들어오면 현재 저장된 테이블에서 공격이름과 두 주소값을 이용한 비교가 수행된다. 같은 값이 존재하지 않는다면 테이블의 속성들에 맞도록 침입시도정보의 속성값들로 채운다. 만약 테이블에 존재한다면 그 침입시도정보는 시간정보를 제외한 대부분의 정보가 같은 동일 침입시도정보이므로 상호연관화 기법에 필요없는 것이므로 버려지게 된다. 미리 정해진 시간이 경과하면 테이블의 값들은 모두 상호연관화 기법을 위해 저장되거나 관리자에게 보고된후 비어진 테이블에서 다시 필터링이 수행된다.

동일한 침입시도정보가 테이블에 존재하여 버려질 때 고려해야할 사항이 있다. 동일한 공격정보이지만 시간값은 다르기 때문에 전제조건을 이용한 상호연관화기법(Correlation using Prerequisite)처럼 침입시도정보들간의 상호연관화에 시간정보를 사용하는 경우 문제가 된다. 그러므로 [표 2]에서 볼수 있듯이 저장될 대표 침입시도정보ID에 버려지는 각각의 침입시도정보들의 시간정보들을 저장해주어 상호연관화 기법에 사용될수 있도록해야 한다. 이 방법은 특정 속성을 사용하는 다른 상호연관화 기법에서도 적용될 수 있다.

표 2 상호연관화 기법을 위해 생성되는 테이블

침입시도정보 ID	시간정보
54786	2004-08-15 20:08:08
54786	2004-08-15 20:08:10

4. 필터링 실험 및 결과

4.1 실험 방법

실험을 위하여 [그림 2]와 같은 네트워크를 구성하였다. 각 호스트는 고유의 이름과 IP주소를 가지고 있다. [표 3]의 System profile은 각 호스트별로 동작하는 제품들과 서비스들중 대표적인 것만 나타내었다. 취약성 정보는 ICAT 데이터베이스에 정의되어 있는 CVE를 사용하였다[6].

두 필터의 특징을 실험 하기 위하여 각 필터에 특화된 두가지 데이터 집합을 이용하여 실험 하였다. 하나는 취약성 필터에 특화된 것이다. 공격 패킷을 생성시켜주는 snot를 이용해서 1000개의 패킷을 랜덤하게 생성하였다. 이 패킷을 Snort를 이용하여 탐지하고 생성된 침입시도정보를 사용하였다. 두번의 랜덤한 패킷생성을 통해 얻은 60종류의 시그네쳐를 가진 400개의 침입시도정보와 49종류의 시그네쳐를 가진 침입시도정보를 이용해 필터링하였다.

다른 데이터 집합은 동일 침입시도정보 필터링을 위해서 같은 종류의 공격들이 주로 나타나는 LLDOS공격 데이터 집합인 DARPA 2000[5]의 데이터를 이용하였다. 이 데이터를 RealSecure가 탐지하고 생성된 침입시도정보들을 사용하였다. 이 데이터 집합은 14종류의 공격으로 이루어진 900여개의 침입시도 정보가 생성된다.

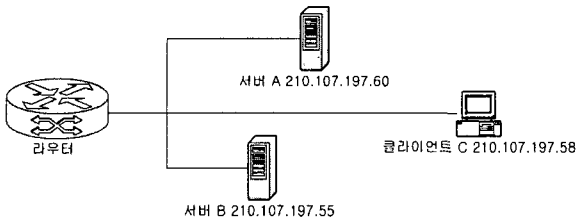

그림 2 실험 네트워크 환경

표 3 System Profile

호스트	제품명 or 서비스	버전 or 포트
A	Red Hat Linux	7.1
A	MySQL	3.23
A	database	8000
B	Red Hat Linux	7.1
B	ProFTPD	1.2
B	FTP	21
C	Windows XP	Professional
C	explore	5.5
C	HTTP	80

4.2 실험결과

표 4 첫 번째 데이터 집합을 이용한 실험결과

#	필터링 전	필터링 후
1	416	213
2	431	167

표 5 두 번째 데이터 집합을 이용한 실험결과

#	필터링 전	필터링 후
1	891	463
2	922	511

[표 4]와 [표 5]는 침입시도정보의 수가 두 필터링 기법에 의해 줄어듬을 보여준다. [표4]는 랜덤하게 생성되는 공격이므로 동일한 공격은 적지만 시스템에 영향을 주지 않는 공격이 많이 생성되므로 첫번째 필터의 영향이 크고 [표 5]는 동일한 공격이 많이 발생하는 DDOS이므로 두번재 필터의 영향이 크게 나타났음을 알 수 있다.

5. 결론 및 연구과제

취약성을 이용한 필터링으로 실제 시스템에 피해를 주지 않는 공격에 대한 침입시도정보를 제거할수 있다. 또한 동일한 침입시도정보 필터링을 이용해 동일한 침입시도정보들 또한 제거할수 있다. 이로인해 상호연관화 기법에서 실제 시스템에 위협을 가할 수 있는 공격들에 대해서만 상호연관화를 수행할수 있도록 도움을 주고 줄어든 침입시도정보들로 인해 보다 적은 시간에 상호연관화를 수행할수 있다.

향후 연구과정으로는 취약성정보와 시스템정보 갱신의 자동화가 필요하다. 취약성의 정도는 계속해서 변화하고 또한 새로운 취약성이 발견되거나 시스템의 변화에 의해 취약성은 변경될수 있다. 이러한 변화에 대응하여 관리자의 개입없이 시스템내에서 자동으로 갱신하는 방법이 필요하다.

5. 참고 문헌

[1] Ning. P, An intrusion alert correlator based on prerequisites of intrusios, Technical Report TR-2002-01, 2002

[2] A.Valdes, Probabilistic alert correlation, RAID 2001, pp 54-68 , 2001

[3] Jong Woon. P, The Design Principles of High-Speed NIDS considering Performance, PARA'04 State-of-the-art in scientific computing, 2004

[4] Roesch. M, Adaptive Alert Throttling for intrusion Detection System, USENIX, 1999

[5] Lincoln Lab MIT. DARPA 2000 intrusion detection evaluation datasets. http://ideval.11.mit.edu/2000, 2000

[6] Icat vulnerabilities database, http://icat.nist.gov/icat.cfm