Honeypot에서의 효과적인 Data Control 방안

이원석° 신휴근 김동규
아주대학교 정보통신전문대학원
{koes°, lightroo, dkkim}@ajou.ac.kr

Efficient Data Control in Honeypots

Wonseok Lee° Hyukeun Shin Dongkyoo Kim
Dept. of Information Communication Engineering GSIC AJOU

요 약

최근들어 인터넷의 보급이 급속도로 확산되면서 인터넷을 통한 개인 정보의 불법적인 침해 사고도 많이 발생하고 있다. 갈수록 다양해지는 공격 방법에 대비하기 위하여 공격 정보를 수집할 필요성이 생기게 되었는데 그에 따라 등장한 것이 Honeypot이라는 개념이다. Honeypot은 고의로 공격자에 의해 공격을 당함으로써 공격 정보를 수집하는 네트웍 자원을 말한다. Honeypot을 구현할 때에는 그것이 다른 정상적인 자원을 공격하는 데에 사용되지 않도록 해야 하는데, 기존의 방법들에서는 확실한 제한이 이루어지지 않았다. 따라서 본 논문에서는 패킷의 방향 재설정을 통하여 Honeypot 오용을 확실히 제한하고, 더 많은 공격 정보를 수집할 수 있는 방법을 제안하였다.

1. 서 론

최근 몇 년간 인터넷의 급속한 성장으로 인해 많은 사람들은 인터넷을 통하여 손쉽게 각종 업무와 서비스 등을 이용할 수 있게 되었다. 인터넷 산업의 성장률과 인터넷 이용률도 갈수록 높아만 가고 있다. 하지만 인터넷의 개방성으로 인해 개인정보 침해 등의 그 역기능도 무시할 수 없는 수준에 이르렀다. 따라서 인터넷에서의 정보 보호에 대한 관심도 높아지게 되었고 실제로 IDS나 방화벽, 취약점 진단 도구 등의 정보 보호 솔루션도 많이 나오고 있다.

그러나 개인 정보에 대한 공격 방법들이 갈수록 매우 다양해지고 있어서 그에 대한 대응책을 마련하기는 점점 더 어려워지고 있는 실정이다. 이러한 상황에서 공격자의 공격 방법이나 그 동기 등의 정보를 모으고, 그것에 대해 연구함으로써 각종 공격 방법에 효과적으로 대처하자는 생각이 등장하였고 그에 따라 나온 개념이 바로 Honeypot이다.

Honeypot은 일반 상용 서버와 같은 하나의 자원으로서, 그것이 공격자에 의해 발견되고 공격, 침해 당함으로써 의미를 가지게 된다[1]. 즉, Honeypot은 공격자에 의해 공격을 당하고 그 정보를 저장함으로써 우리에게 공격에 대한 정보를 제공해 주는 것이다.

Honeypot을 구현함에 있어서 요구되는 여러 사항 중 한 가지는 Honeypot이 네트웍의 정상적인 자원에 해를 끼쳐서는 안된다는 것이다. 다시말해 공격자가 어떤 가치있는 서버를 공격하는데에 Honeypot이 쓰여서는 안된다[2]. 이것을 Data Control 이라고 하는데, 본 논문에서는 공격 정보의 수집과, Honeypot이 다른 서버를 공격하는데 이용되는 것을 방지하는데 있어서 좀 더 효과적인 Data Control 방안에 대하여 제시하고 있다.

2. 관련연구

2.1 Honeypot의 유형과 분류

Honeypot은 그것이 가지게 되는 의미에 따라 Production Honeypot과 Research Honeypot으로 나뉘게 되는데 각각이 가지는 의미는 다음과 같다[2].

- Productions Honeypot : 어떤 조직에 있어서 공격에 대한 위험도를 감소시키는데에 의미가 있다. Honeypot을 통해 그 Honeypot이 속해 있는 네트웍에 어떤 공격이 이루어지고 있음을 탐지 하고, 침해 사고가 발생했을 경우라도 Honeypot에 저장된 공격 정보를 이용하여 빠른 대처를 할 수 있도록 한다.
- Research Honeypot : 가능한 한 많은 정보를 모음으로써 공격자와 그 공격 방법을 연구하는데에 도움을 주는 Honeypot이다. 이렇게 수집되고 연구된 정보는 전반적인 정보 보호 기술의 발전에 기여하게 된다.

Honeypot은 그것과 공격자 사이에 이루어지는 상호작용의 정도에 따라서 세가지 단계로 분류되는데, 그 특징은 다음과 같다[3].

- Low-Involvement Honeypot : 특정 포트에 대해 접속만 허용하고 서비스는 제공하지 않는다.
- Mid-Involvement Honeypot : 실제 서비스를 제공하는

것처럼 행동하는 가상의 서비스를 제공한다.
•High-Involvement Honeypot : 실질적인 O/S와 모든 서비스를 공격자가 이용할 수 있도록 제공한다.

Honeypot과 공격자간의 상호작용의 정도가 높아질수록 수집할 수 있는 정보는 많아지지만, 그에 따른 위험도 증가한다.

2.2 Honeypot 구성에 있어서의 요구사항

네트웍상에 Honeypot을 설치할 때에는 다음과 같은 사항을 고려해야 한다[4]. 이 요구사항들은 Research Honeypot, High-Involvement Honeypot에 주로 적용된다.

•Data Capture : 가능한 한 많은 정보를 수집하되, 공격자가 자신의 모든 행동이 기록되고 있다는 것을 알아서는 안된다.
•Data Control : 침해당한 Honeypot이 네트웍 상의 다른 정상적인 자원에 해를 끼치는데 이용되어서는 안된다. 이러한 제한도 공격자가 알아차리지 못하도록 유연하게 이루어져야 한다.

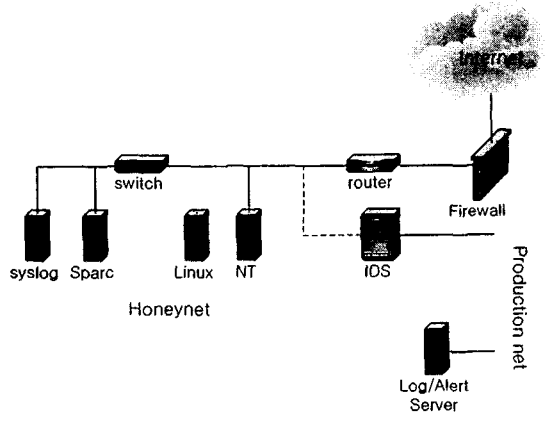
2.3 Honeynet

Honeynet은 Honeypot의 하나로써 공격에 대한 연구를 목적으로 만들어진 Research Honeypot이며, 일반적인 서버가 제공하는 서비스를 모두 제공하는 High-Involvement Honeypot이다. Honeynet이 기존 Honeypot과 다른점은 단일 시스템이 아닌, 여러 개의 O/S로 구성된 네트웍 자체를 Honeypot으로 이용한다는 것이다. 이렇게 함으로써 더욱 더 다양하고 많은 정보를 수집할 수 있다.

Honeynet은 구현 방법에 있어서 1세대와 2세대로 나누어지는데 각각의 내용은 다음과 같다[4].

2.3.1 1세대 (1999~2001년)

1세대 Honeynet 네트웍은 <그림 1>과 같이 방화벽, IDS, 라우터 등을 이용하여 구성된다.

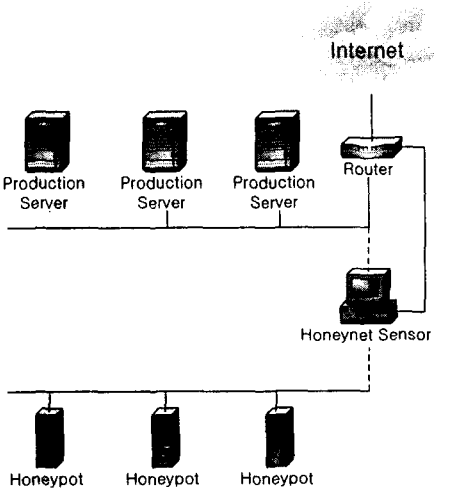


<그림 1> 1세대 Honeynet

•Data Capture : 방화벽을 이용하여 Honeynet으로 들어오고 나가는 모든 연결 정보를 수집하고, IDS를 이용하여 모든 네트웍 트래픽 정보를 수집하여 공격에 대한 정보를 모은다. 또한 키 입력과 같이 Honeypot 시스템 내에서 이루어지는 공격자의 활동도 캡쳐 프로그램을 이용하여 모두 수집한다.
•Data Control : 방화벽을 이용하여 Honeynet으로부터 외부로 시도되는 연결의 개수를 제한한다. 라우터를 이용하여 방화벽의 존재를 숨기고 패킷의 발신 주소 검사, ICMP 트래픽 제한 등의 부가적인 Data Control을 수행한다.

2.3.2 2세대 (2002년~)

모든 요구사항을 하나의 장치에 구현하고, 그 장치가 네트웍 브릿지로 동작하도록 하여 공격자가 Honeynet의 존재를 알아차리기 어렵도록 한다. 네트웍은 <그림 2>와 같이 구성하였다.



<그림 2> 2세대 Honeynet

•Data Capture : 모든 정보를 커널 수준에서 수집하여, 암호화된 정보도 알아낼 수 있다. 수집된 정보는 정상적인 네트웍 트래픽인 것처럼 보이게 하여 원격으로 저장함으로써 공격자가 공격정보의 수집 여부를 알아내기 힘들게 한다.
•Data Control : 패킷의 내용을 보고 공격자의 행동이 무엇인지 판단하여 외부에 대한 공격인 경우 패킷을 드롭시키거나 위험하지 않은 패킷으로 수정한다.

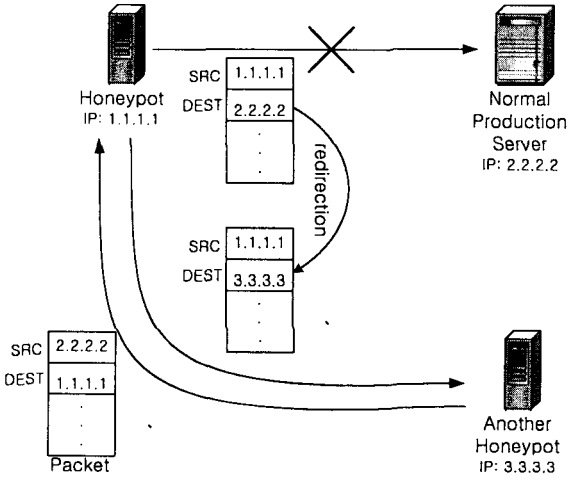
3. 제안된 Data Control 방안

1세대 Honeynet에서 이루어졌던 Data Control은 단순히 연결의 개수만을 제한하는 방법이어서 확실한 Data Control이 이루어지지 않았고, 연결의 제한을 공격자가 알

아차리기가 쉬웠다[2]. 또, 2세대 Honeynet의 Data Control은 위험한 패킷의 여부를 판단하는 방법이 IDS의 공격패턴 정보에 의존적이어서, 공격패턴 정보에 없는 공격의 경우 확실한 제한이 이루어지지 않는 한계를 가지고 있다.

3.1 패킷 방향 재설정 (Redirection of Packet)

본 논문에서 제안하는 Data Control 방법은 Honeypot으로부터 일반 상용 서버로 나가는 패킷의 목적지를 재설정하는 것이다. <그림 3>과 같이 Honeypot에서 정상적인 서버로 보내는 패킷의 수신 주소를 또 다른 Honeypot의 주소로 바꿈으로써 정상적인 서버에는 아무런 피해를 끼치지 않게 된다.

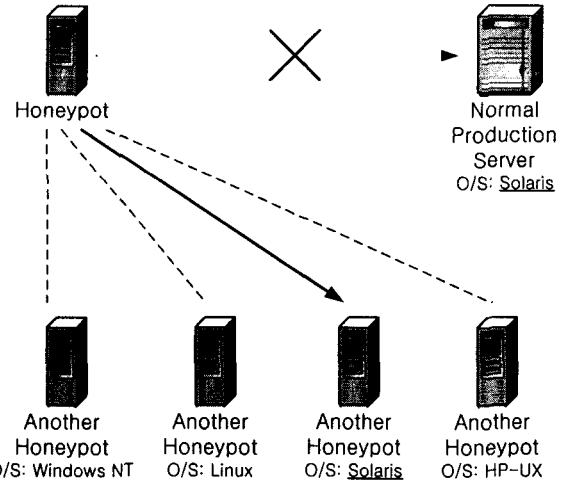


<그림 3> 패킷 방향 재설정을 이용한 Data Control

공격의 타겟이 된 Honeypot은 공격에 따른 반응을 하기 때문에 공격자는 원래 의도한 서버에 공격이 이루어지는 것으로 알게 되므로 그 후의 추가적인 행동도 계속 하게 된다. 이전에 이루어졌던 Data Control 방법은 공격 행위 자체를 제한하였지만, 제안된 방법은 공격자에게 Honeypot을 이용한 다른 서버에의 공격을 허용하면서도 실제적으로는 아무런 피해를 주지 않는다. 따라서 확실한 Data Control이 이루어짐과 동시에 더 많은 공격 정보를 수집할 수 있도록 해주는 것이다.

3.2 Intelligent Redirection

앞에서 제시한 방법에서 만약 새로 공격의 타겟이 된 Honeypot이 Windows NT O/S이고 원래 공격하기로 의도된 일반 서버는 Solaris O/S였다면 공격자가 대상 서버의 반응을 보고 무언가 잘못되고 있다는 것을 알아낼 수가 있다. 이렇듯 원래 의도된 서버와 공격 타겟 서버가 공격에 대해 너무 상이한 반응을 보일 경우 문제가 될 수 있는 것이다. 이러한 문제는 패킷의 목적지를 원래 의도된 일반 서버와 같은 종류의 O/S를 가진 Honeypot으로 설정함으로써 해결될 수 있다. 즉, 패킷을 목적지를 변경하기 전에 원래 의도된 서버의 O/S 종류를 알아내고, 자신이 가지고 있는 협력 Honeypot 목록에서 가장 비슷한 종류의 Honeypot을 찾아서 그쪽으로 패킷의 목적지를 설정하는 것이다.



<그림 4> Intelligent Redirection

4. 결론 및 향후 연구 방향

본 논문에서는 Honeypot의 유형과 Research Honeypot의 하나인 Honeynet에 대하여 알아보고, 기존의 Honeynet에서 사용하는 Data Control 방법의 제한점을 살펴보았다. 또한 Honeypot에서 외부로 나가는 패킷의 목적지를 또 다른 Honeypot으로 재설정 하는 방법을 통하여 Honeypot을 이용한 정상적인 서버에의 공격을 확실히 방지하고, 더 많은 공격 정보를 수집할 수 있다는 것도 알아보았다. 하지만 아직 제안된 방법의 구현 방안에 대해서는 자세한 내용이 부족하다. 따라서 앞으로의 연구에서는 패킷 방향 재설정 방법의 더 구체적인 프로토콜과, 원래 의도된 일반 서버와 공격 타겟 서버간의 매칭 등에 대해 심도있게 알아보는 것이 필요하다.

5. 참고 문헌

[1] Lance Spitzner, "Honeypots: Definitions and Value of Honeypots", http://www.enteract.com/~lspitz/honeypots.htm, 2002.5

[2] Lance Spitzner, "Honeypots: Traking Hackers", Addison-Wesley, 2002.12

[3] Reto Baumann, "White Paper: Honeypots", http://security.rbaumann.net/download/whitepaper.pdf, 2002.2

[4] Honeynet Project Members, "Know Your Enemy: Honeynets", http://www.honeynet.org/papers/honeynet/, 2002.9