가상머신에서의 포트 분리 관리 허니팟과 허니넷 설계

임인빈° 최재호
전북대 미디어통신 연구실
{cnick°, wave}@chonbuk.ac.kr

Design of Honeynet with separated port managing Honeypot on VM

Inbin Yim° Jaeho Choi
Media Communication Lab, Chonbuk National University

요 약

네트워크가 복잡해지면서 다양한 형태의 위협에 노출된다. 일반적인 보안 솔루션으로 사용하는 방화벽(Firewall)이나 침입탐지시스템(IDS)은 허가 받지 않은 외부의 접속이나 알려진 공격만을 차단하는 단순하고 수동적인 시스템이다. 이에 반해 허니팟은 웹서버와 같은 실제 Front-End 시스템과 유사하거나, 밀접한 관련을 갖고 직접적으로 반응하므로 신뢰성 높은 실시간 정보를 얻을 수 있어서 관리자가 다양한 위협에 능동적이고 효과적으로 대응할 수 있다. 실제로 웜이나 DRDoS(Distributed Reflection DoS)등 수 분만에 네트워크를 점유하는 자동화 공격과 함께 시스템 구성 취약점을 파고들거나 계정 획득을 통한 DB서버 등의 Back-End 시스템에 대한 수동 공격이 혼재한다. 따라서 시스템 전반적인 관리의 중요성이 강조되고 있다. 본 논문에서는 그간의 실험결과를 바탕으로 가상 머신으로 허니팟을 구성하고 특성별로 포트를 분리하여 관리하는 허니넷을 제안하고자 한다. 이를 통해 1) 유연한 보안 시스템 구성이 가능하고 2) 관리 효율이 높아지며 3) 하드웨어 도입 비용 절감을 통해 시스템의 TCO(Total Cost of Ownership)를 감소시키는 효과를 기대할 수 있다.

1. 서 론

다양한 위협이 존재하는 네트워크에서 허니팟은 방화벽(Firewall)과 침입탐지시스템(IDS)의 단순하고 수동적인 특성을 보완하는 능동적인 솔루션이다. 공격자는 보안시스템에 포착되지 않으려 한다. 허니팟(Honeypot)은 회피해야 할 보안 시스템이 아니라 공격자가 침입하고자 하는 환경으로 오판하게 한 후 공격 내용을 저장하고 향후에 이를 제어할 용도로 사용한다.[1] 이를 위해 웹서버 혹은 FTP서버와 같은 Front-End 시스템과 유사한 거짓응답을 보내거나 실제 시스템과 밀접한 관련을 갖고 공격에 직접적으로 반응하므로 관리자에게 보다 신뢰성 높은 정보를 제공한다. 웜이나 DRDoS 등 수 분만에 네트워크를 점유하는 자동화 공격과 함께 시스템 구성 취약점을 파고들거나 계정 획득을 통한 DB서버 등의 Back-End 시스템에 대한 수동 공격이 혼재하는 현실에서 관리자는 다음의 두 가지 문제에 직면하게 된다. 첫째, 다양한 공격이 뒤섞인 로그를 분석하고 관련성을 찾아 이들을 분리하는 작업에 시간을 허비하게 된다. 둘째, 조직의 환경에 가장 적합한 보안 시스템을 구축하고 이를 지속적으로 관리하며 때에 따라선 유연하게 변경할 필요성이다.[2] 이 두 가지를 유연하고 효과적으로 해결할 수 있는 방법으로 가상머신을 생각할 수 있다. 본 논문에서는 그간의 실험결과를 바탕으로 가상 머신으로 허니팟을 구성하고 특성별로 포트를 분리하여 관리하는 허니넷을 제안하고자 한다. 이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고 3장에서는 실험 시스템에 대한 소개를 4장에서는 그간의 실험 데이터를 통해 제안한 시스템의 효용을 고찰하고 5장에서는 결론과 함께 향후 연구과제와 개선점을 기술한다.

2. 관련 연구

네트워크 보안 제품은 각각의 동작 특성에 따라 'Prevention', 'Detection', 'Response'로 나눌 수 있다. 이는 순서대로 방화벽, 침입탐지시스템, 허니팟으로 대표되는데 그 중 허니팟은 1) 순수 소프트웨어 기반 2) 하드웨어 기반 3) 가상머신의 세 가지 시스템을 생각할 수 있다.[3] 우선, 순수 소프트웨어 기반 허니팟은 거짓 응답(Fake Reply)을 보내는 것으로 시스템에 부담을 주지 않고, 추가적인 하드웨어가 필요하지 않으며, 쉽게 사용할 수 있지만 정해진 단순 응답만을 보내므로 공격자에게 쉽게 노출된다. 다음으로 하드웨어 기반 허니팟은 일반적으로 라우팅 알고리즘 등을 적용하여 각 시스템의 의존성을 낮출 수 있고 부하 분산에 유리하지만 시스템의 유연성이 떨어지고 하드웨어 비용이 급증한다. 또한 Nmap 등의 네트워크 스캐닝 도구에서 지원하는 '스텔스 스캐닝'으로 특정 대역에 대한 검색이 성공하는 경우엔 전체적인 시스템의 구조가 그대로 노출될 위험이 있다. 마지막으로 가상머신으로 허니팟을 구축한 경우엔 실제 시스템을 통해 데이터의 수집과 제어가 이루어지는 사이 가상머신은 공격에 응답하는 역할을 담당한다.[4] 가상머신으로 허니팟을 구축하는 경우, 가장 큰 장점은 추가적인 하드웨어 도입 비용을 들이지 않고도, 가상머신이 지원하는 다양한 환경의 허니팟을 다 수개 만들 수 있다.[3] 하지만 가상 머신이 실제 시스템의 자원을 사용하므로 확장에 한계가 있고, 호스트 시스템의 성능에 허니팟의 성능이 의존적이다.

3. 제안 시스템

3.1 시스템 개요

본 논문에서는 패킷이 호스트 전단의 방화벽을 통과한 후 호스트에 가상머신으로 구축한 두 개의 허니팟 사이에 위치한 버추얼 스위치를 통해 각각의 허니팟에선 할당된 범위의 패킷만을 처리하고 나머지는 폐기하는 것으로 포트 분리를 구현하였다.

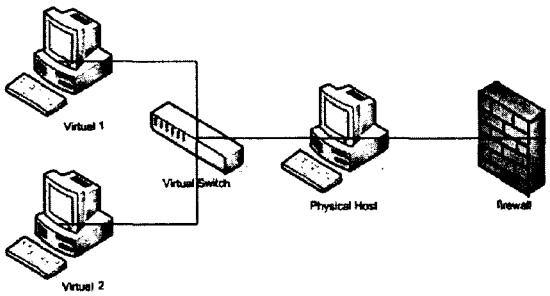

그림 1. 제안 시스템 개요

3.2 주요 요소별 역할과 특성

3.2.1 방화벽 (firewall)

시스템의 전단에서 기본적인 패킷 필터링을 수행한다. 방화벽은 기본적으로 Nmap등의 스캐닝 툴로 현재 활성화되어있는 서비스와 운영체제를 검색하는 것에 대한 응답을 막아서 시스템의 상태를 노출하지 않기 위해 사용한다. 제안한 시스템에서는 이와 함께 1) 불필요한 외부 트래픽 유입을 최소화하면서 2) 시스템 자원 낭비를 막고 3) 시스템 의존성을 최소화하기 위해 네트워크에서 제공하는 방화벽을 사용했다.

3.2.2 가상 스위치 (Virtual Switch)

본 시스템에서 가상 스위치는 각각의 가상머신으로 구축된 허니팟에 패킷을 전달하는 역할을 한다. 현재까지 사용되는 공격 방법 중 네트워크를 고립시키는 방법의 하나는 스위치나 라우터를 공략하는 것이다. 그러나 제안한 시스템에서 사용할 스위치는 실제 네트워크 환경을 Bridge로 하는 가상의 IP를 갖고 있어서 공격자는 스위치의 존재 여부를 알 수 없다. 설령 가상 스위치의 존재를 인식하더라도 가상의 스위치에 대한 개별적인 공격은 그 자체가 불가능하다.

3.2.3 가상 머신(Virtual 1,2)

먼저, 가상머신에서 사용할 운영체제의 선택에 있어, 2003년 한국 정보보호 진흥원의 분석에 따르면,[5] 전체 피해 운영체제 중 마이크로소프트社의 제품군이 차지하는 비율이 96%를 차지하며 증가추세에 있는 것을 볼 수 있다.

2002년		2003년	
운영체제	비율	운영체제	비율
윈도우NT군	44.8%	윈도우NT군	62.6%
윈도우95/98	41.3%	윈도우95/98	33.5%
리눅스	11.3%	리눅스	3.7%
솔라리스	1.8%	솔라리스	0.2%
기타	0.8%	기타	0.1%

표 1. 해킹 피해 운영체제별 분류 (출처: 한국정보보호진흥원)

따라서 가상 머신의 운영체제로 윈도우 제품군을 선택하는 것이 가장 현실적이라고 생각했다.
다음으로, 포트 분리 관리에 대해서는 아래에 표로 정리한 기존의 통계[6]를 토대로 1024번을 분리의 기준으로 삼았다.

포트 번호	용도	발생 빈도
21	FTP	17,270
22	SSH	2
25	SMTP	200
43	Whois	6
80	HTTP	3,634
443	HTTPS	6,647
1024 이후	Worm, Trojan ...	26,842

표2. 포트별 로그 발생 빈도 (출처 : 참고문헌[6])

이에 따라 왼쪽의 그림1. 시스템에서의 Virtual 1은 0~1024번까지의 포트를, Virtual 2는 1024번 이후의 포트를 관리한다.

4. 성능 평가

4.1 데이터 수집

본 논문의 실험데이터는 2004년 6월 16일부터 8월 16일까지 총 62일간 수집하였으며, 6월 16일부터 8월 1일까지는 대조군을 생성하기 위해 순수 소프트웨어 기반 허니팟을 사용한 허니넷 환경으로 실험했고 제안한 시스템을 구성하고 실험한 성능 데이터는 8월 2일부터 16일까지 15일간의 결과를 바탕으로 한 것이다.

4.2 실험 환경

4.2.1 하드웨어 환경

인텔社의 펜티엄4 2.6GHz (HyperThreading 지원) CPU에 512MB메모리를 가상 머신에 각각 128MB씩 할당하였고, 120 GB 의 하드 중 가상 하드로 각각 4GB 공간을 할당하였다.

4.2.2 소프트웨어 환경

현재 출시되어 있는 가상머신 제품 중 윈도우 환경에서 사용할 수 있는 것은 Virtual PC와 VMWare의 두 가지가 있다. 그러나 Virtual PC는 특정 시스템에만 최적화되어 있어서, 본 논문에서

는 일반적으로 성능과 안정성이 Virtual PC보다 뛰어나고 지원하는 운영체제가 많은 VMWare Workstation 4버전으로 두 개의 가상머신을 생성한 후 각각 서비스팩 1이 적용된 한글 윈도우즈XP (Professional)을 운영체제로 설치하였다. 또한 포트 감시 도구로는 IDS 기능이 있는 KFSensor 2.2.1 버전을 사용했다. 추가로 가상 머신 1번에는 널리 알려진 1024번 이전 서비스 중 공격에 사용되는 것으로 알려진 요청에 대한 거짓 응답을 보내도록 했다.

4.3 실험 결과

4.3.1 시스템 구축 전

포트 번호	내용	발생빈도
21	FTP	4159
23	Telnet	109
25	SMTP	47
80	HTTP	5373
443	HTTPS	7852
110	POP3	1
1024 이후	Worm, Trojan	9649

표 3. 포트별 로그 발생 빈도

기간		발생빈도	주원인
6월	2주	2	Worm
	3주	3	HTTPS
	4주	3	HTTPS
7월	1주	6	Worm
	2주	7	HTTPS,
	3주	9	Worm
	4주	17	Telnet
8월	1일	1	Worm

표4. 허니팟 시스템다운 빈도와 주원인

상태	CPU	RAM 점유	반응시간
평소 상태	22 %	190MB (+8MB)	
웜/ HTTPS	47 %	201MB (+19MB)	+45초

표5. 상태에 따른 시스템 성능과 관리에 소요한 대처 시간

4.3.2 제안한 시스템 구축 후

본 논문에서 제안한 시스템을 구축한 이후 15일간은

1) 각 포트별 로그 발생빈도는 표3과 유사한 통계를 보였다.

2) 표4에서와 같은 시스템 다운이 단 1번도 발생하지 않았다.

3) 가상 머신을 사용함에 따라 표5에서 보인 호스트의 상태는 다음과 같이 변화하였다.

상태	CPU	RAM 점유	반응시간
평소상태	41 %	318 MB	
웜/HTTPS	70 %	345MB (+27MB)	+37초

표6. 가상머신을 사용한 시스템의 성능과 관리에 소요한 대처 시간

5. 결론 및 향후 연구 과제

5.1 결론

위에서 살펴본 바와 같이 본 논문에서 제안한, 가상 머신으로 특성에 따라 포트를 분리하여 관리하는 허니팟을 사용하는 허니넷이 대조군과 비교해서 대처에 필요한 시간이 단축되었다. 특히'시스템 생존율'(System Survivability)이 증가한 것을 알 수 있다. 이는 제안한 시스템이 관리자의 트래픽 관리와 시스템 보호에 효과적이란 사실을 증명한다고 할 수 있을 것이다. 즉, 특성이 구분되는 수동과 자동화 공격의 트래픽을 각각의 허니팟에서 단일 시스템보다 빠르게 판별하고 구분해서 관리할 수 있었기 때문으로 생각한다.

그런데, 실험 기간 내내 외부에서 내부로 침투하는 수동 공격보다, 부주의한 내부 네트워크 사용자들이 초래하는, 1) 메일에 첨부되어 침투하는 웜 2) 특정 사이트에서 사용하기 설치하는 Active-X나 불법적인 경로를 통해 설치한 프로그램에 숨어있던 트로이 목마 등이 발생시키는 내부 네트워크에서 외부로 나가기 위해 허니팟으로 Redirection 되는 트래픽이 상당한 것을 발견할 수 있었다. 또한 제안한 시스템 구축 전에 있었던 대조군 시스템 다운의 주원인 중에 Telnet접속에 의한 경우가 있었다. (표3. 7월 4주) 로그를 분석한 결과 자동화된 사전대입식 공격에 의한 것이었는데, 직접적인 관련성을 찾진 못했지만, 마침 중국 해커에 의해 국가 기관의 정보가 유출되었다는 뉴스가 떠들썩하던 기간이었다.

5.2 향후 과제

내부 네트워크로부터 허니팟으로, 웜에 의해 단기간에 대량의 패킷이 일정 시간동안 발생하는 경우, 두 개의 가상머신이 호스트의 자원을 경쟁적으로 사용하는데 따르는 "Race Condition"이 호스트에 미치는 영향이 증가하여, 호스트의 다른 작업들에 대한 응답 시간이 불규칙적으로 변화하는 것을 발견할 수 있었다. 따라서 이후로는 내부 네트워크에서 호스트로 Redirection 되는 대량의 패킷을 처리하는 방안에 대한 추가적인 연구를 진행하려 한다.

6. 참고문헌

[1]Lance Spitzner, "Honeypots: Simple, Cost-Effective Detection" www.tracking-hackers.com , Apr. 2003

[2]Reto Baumann, "White Paper: Honeypots", Feb. 2002

[3]Andrew Lamb, "The Distributed Honeypot Project", www.lucidic.net , May. 2004

[4]Niels Provos, "A Virtual Honeypot Framework", CITI Technical Report 03_1, Oct. 2003

[5]심원태, "2003년 인터넷 침해사고 유형분석", 한국 정보보호 진흥원, p10-p24, 2003

[6]Marc Dacier/Fabien Pouget, "Honeypots: Practical Means to Validate Malicious Fault Assumptions", Proceeings of 10th IEEE PRDC'04, 2004