실행시간 악성실행코드 탐지 시스템 설계

오형근°, 배병철, 김은영, 박중길
국가보안기술연구소
{hgoh°, bcbae, eykim, jgpark}@etri.re.kr

Design of Malicious Execution Code Detection System at run-time

Hyung Geun Oh°, Byung Chul Bae, Eun Young Kim, Joonggil Park
National Security Research Institute

요 약

네트워크 환경이 발전함에 따라 엑티브엑스 컨트롤과 같은 이동 실행 코드들의 사용이 증가하고 있으며 동시에 사용자가 본래 의도했던 행위 대신에 로컬 자원에의 불법적인 접근 및 시스템 파괴와 같은 악성 행위로 인한 피해가 증가하고 있다. 이러한 악성실행코드들은 바이러스와 더불어 웹의 발전으로 광범위하게 확산될 것으로 예상되며 피해 규모도 바이러스에 버금갈 것으로 예상된다. 이에 본 논문에서는 기존에 알려진 악성실행코드뿐만 아니라 알려지지 않은 악성실행코드들에 의해 사용자 컴퓨터에서 발생할 수 있는 각종 악성행위를 탐지하고 그 행위를 차단하며 탐지된 정보를 신속히 공유함으로써 악성실행코드에 대한 대응력을 강화시킬 수 있는 실행시간 악성실행코드 탐지 시스템을 설계한다.

1. 서 론

인터넷을 위한 다양한 기술이 등장하고 인터넷 접속자가 폭발적으로 증가하게 되면서 이들을 대상으로 하는 각종 컨텐츠 산업이 빠르게 성장하고 있다. 또한, 웹을 통해 컨텐츠를 액세스하는데 있어서 보다 향상된 기능 및 편의성이 요구되면서 기존의 방법으로는 한계를 느끼게 되어 자바 애플릿(Java Applet), 자바 스크립트(Java Script) 그리고 엑티브엑스 컨트롤(ActiveX Control) 등과 같은 다양한 실행 코드 기술들이 발전하고 있으며 각종 실행 코드의 배포가 활발해지고 있다. 그러나 엑티브엑스 컨트롤과 같은 이동 실행 코드들은 사용자가 본래 의도했던 행위 대신에 로컬자원에의 불법적인 접근 및 시스템 파괴와 같은 악성 행위를 수행할 수 있다. 이와 같은 악성실행코드들은 바이러스와 더불어 웹의 발전으로 인하여 광범위하게 확산될 것으로 예상되며 그 피해 규모도 바이러스에 버금갈 것으로 예상된다.

본 논문에서는 각종 실행코드 실행시 불법적인 내부 자원 접근 및 유출을 방지함으로써 사용자 컴퓨터를 보호하고 각 개별 에이전트에서 탐지된 악성실행코드 정보를 공유함으로써 악성실행코드의 확산을 방지하고 피해를 감소시키기 위한 방법을 제안하도록 한다. 본 악성실행코드 탐지 시스템은 사용자 영역에서 실제 악성실행코드를 탐지하는 악성실행코드 탐지 에이전트와 이를 관리하기 위한 관리서버로 구성되어 있다.

2. 기술 동향

인터넷의 원활한 사용을 원하는 사용자의 요구에 따라 웹 브라우저 개발업체는 실행코드에 대한 코드분석이 아닌 신뢰를 바탕으로 한 사용자의 결정에 따라 보안정책을 완화시킬 수 밖에 없다.

자바 기술이나 엑티브엑스 기술은 로컬 시스템의 자원을 직접 접근할 수 있으며 웹 브라우저는 이러한 자원 접근을 막거나 감시할 수 없다. 마이크로소프트를 비롯한 웹 브라우저 개발 업체는 이러한 자원 접근에 따른 사용자들의 피해를 줄이는 방지책으로 실행코드에 디지털 서명을 하도록 하여 실행코드가 인터넷을 통해 다운로드 되었을 때 인증할 수 있는 기능을 추가 보완하였다. 그러나 이러한 디지털 서명 기법은 단지 배포자에 대한 인증만을 수행할 뿐 실행코드의 행위를 감시하거나 검사할 수 없다. 만약 배포자 고의나 또는 해커에 의해 악성코드가 내포되어 있다면 디지털 서명 기법으로는 그 피해를 막을 수가 없다.

결국 악성실행코드의 피해를 막기 위해서는 실행코드의 행위를 감시할 수 있는 별도의 대책이 필요하다. 실행코드의 행위를 감시하고 악성 여부를 판단하기 위한 대책으로 스캐닝 프로그램을 통한 스캐너의 활용 방법이 있다. 현재 상업적으로 Finjan사의 Surfinshield와 SurfinGate 및 Digitivity사의 Cage 그리고 Trend사의 AppletTrap 등의 스캐닝 프로그램이 일반에 활용되고 있다.

2.1 SurfinShield

클라이언트 단에서 사용할 수 있는 솔루션이며 실시간 모니터링이 가능하다. 웹 브라우저에 포함된 자바 라이브러리 기능의 일부를 대체해야 하는 단점이 있어 클라이언트가 웹 브라우저의 버전을 갱신할 경우에는 Sunfinshield도 같이 갱신해야 한다.

2.2 SurfinGate

서버 솔루션인 SurfinGate는 HTTP 프락시서버에 설치되며 별도의 클라이언트용 스캐너 없이 악성실행코드를 탐지한다. 그러나 SurfinGate는 악성실행코드에 대해 정적 스캐닝을 수행하며 스캐닝 알고리즘이 다소 느리다는 단점을 가지고 있어 이를 극복하기 위해 자체적으로 데이터베이스를 유지하고 있다. 이러한 방식은 한번 안전하다고 판단한 실행코드는 추후 변조되어 악성코드를 내포하고 있더라도 악성 여부를 재검사하지 않기 때문에 피해를 볼 수 있다.

2.3 Cage

Cage 역시 HTTP 프락시서버에 설치하는 서버 솔루션이다. 그러나 실시간으로 모니터링이 가능한데 이는 X 윈도우즈와 유사하게 동작시킴으로써 가능하다. 실행코드의 실행은 Cage 서버에서 이루어지며 이에 따른 GUI 요청은 클라이언트의 에이전트에 보내진다. 이를 받은 X 터미널 역할을 하는 사용자들은 실행코드가 로컬에서 실행되는 것 같이 여기게 된다. X 윈도우즈와 같이 동작하기 때문에 서버 단에 과중한 부하를 야기 시키게 된다.

2.4 AppletTrap

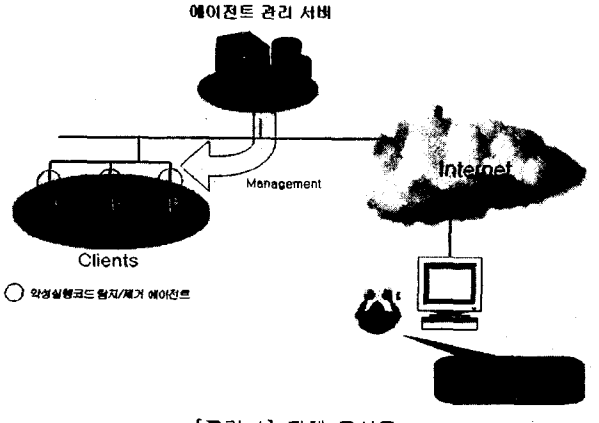
AppletTrap는 서버 단 혹은 에이전트 단에서의 부하를 감소시키기 위해 서버에 설치하면서도 부하를 클라이언트에 분배시

켜 부하 분산을 이루도록 하고 있다. AppletTrap은 HTTP 프락시서버에서 동작하며 정적 스캐닝과 실시간 스캐닝을 조합한 형태로 동작한다. 먼저, 기존의 악성실행코드를 탐지하고 이를 제거한다. 필터링을 통해 악성으로 판명되지 않은 이동코드는 보안정책이 포함된 보안감시코드를 실행코드에 추가한 후 클라이언트에 보낸다. 실행코드는 클라이언트 단에서 실행되면서 보안 정책에 위배되는 행위를 수행할 경우에는 실행이 중지되며 서버에 악성실행코드의 ID와 위반 사항이 보고 되고 프락시 서버는 향후 필터링시 이를 반영하여 재차 다운로드 되는 악성실행코드를 사전에 방지한다.

3. 제안 방식

3.1 전체시스템 구성

악성실행코드 탐지 시스템은 인터넷을 통해 들어온 악성실행코드를 사용자 영역에서 분석 및 모니터링하여 탐지하는 악성실행코드 탐지/제거 에이전트(이하 '에이전트')와 탐지되어 보고된 악성실행코드 정보를 바탕으로 새로운 보안 정책을 생성/배포/관리할 수 있는 악성실행코드 탐지/제거 관리서버(이하 '관리서버')로 구성되는 클라이언트/서버 모델의 지능형 시스템이다. 전체 시스템 구성도는 아래 [그림 1]과 같다.



[그림 1] 전체 구성도

3.2 에이전트

사용자 컴퓨터에서 악성실행코드의 행위를 모니터링하는 에이전트는 크게 파일(파일 매니저), 네트워크(네트워크 매니저), 프로세스(프로세스 매니저) 그리고 레지스트리(레지스트리 매니저)로 탐지 영역을 구분하여 작동된다. 각 파일, 네트워크, 프로세스 및 레지스트리 탐지 모듈은 이벤트가 발생되었을 경우 데이터 매니저로 로그를 전달하며, 데이터 매니저는 다시 침입탐지 모듈에서 보안 정책과 비교하여 탐지 여부를 판단할 수 있도록 해당 로그를 전달한다.

DB 매니저는 정책을 서버로부터 수신하고 각 매니저가 DB로부터 정책을 로드 및 저장하도록 데이터 관련 작업을 관리하는 매니저이다. 정책 매니저는 로그 전달 이외에도 서버로부터의 정책들을 수신 및 전달 기능을 담당하는 매니저이다. 이와 같이 스캐너 엔진(Scanner Engine)은 각 모듈의 역할을 분리시킴으로써 최적의 탐지 구조를 갖도록 하였으며 새로운 기능의 추가 및 제거도 원활하게 되도록 구성하고 있다.

에이전트는 3단계의 악성실행코드 필터링 단계를 수행한다.

우선, 알려진 악성실행코드를 탐지하는 DB 필터링 단계에서는 기존 탐지된 악성실행코드의 해쉬 데이터 값과 비교하여 필터링을 수행한다.

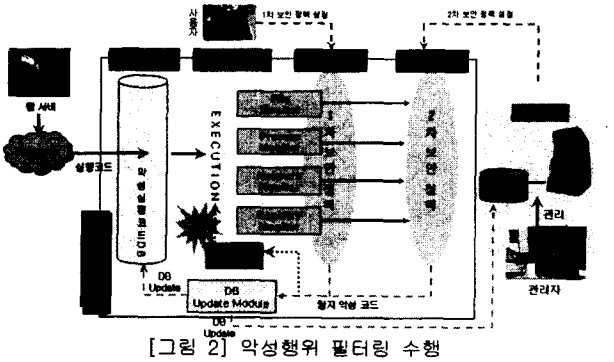
[표 1] 에이전트 기능

에이전트	탐지/제거	- 악성실행코드 DB를 이용한 알려진 악성실행코드 탐지 - 행위 위험도 분석를 통한 알려지지 않은 악성실행코드 탐지
	DB 관리	- 악성실행코드 패턴 정보 관리 - 탐지 보고된 악성실행코드 패턴 정보 관리
	사용자 GUI	- 간편한 보안 정책 - 3단계 보안 설정 - 사용 환경별 정책 설정
	수동 검사	- 특정 실행코드 사용시 수동 검사
	업데이트	- 탐지시 악성실행코드 자동 전송

두 번째로는 사용자 및 관리서버에 의해 설정된 1차 보안 정책에 의한 2차 필터링을 수행한다. 2차 필터링은 특정 디렉토리에의 접근, 특정 프로세스의 실행 방지, 특정 레지스트리의 접근 방지 그리고 특정 포트의 사용 금지 등 단순한 형태의 보안 정책을 반영한 필터링 과정이다. 이때, 관리자가 수립/배포한 1차 보안 정책은 사용자에 의해 변경될 수 없도록 함으로써 사용자의 정책 설정 실수에 의한 피해가 없도록 하였다.

3차 필터링은 관리서버에 의해 배포되는 2차 보안정책을 통해 행위 위험도 산정을 수행함으로써 이루어진다. 2차 보안정책에서는 각 파일, 네트워크, 프로세스 및 레지스트리별 행위에 대한 위험도 산정 포인트 테이블을 유지하면서 데이터 매니저를 통해 보고된 2차 로그에 해당 이벤트별 위험도를 부여한다. 침입탐지 모듈에서는 프로세스별 전체 이벤트 위험도를 누적하고 이 누적된 위험도가 악성행위로 판정되기 위한 임계치 값을 초과하는지 판단하여 초과할 경우에는 프로세스의 실행을 중지시키고 해당 프로세스에 대한 정보와 위험도가 부여된 2차 로그를 관리서버에 전송한다. 이때, 임계치 값은 사용자가 설정한 보안 등급에 따라 달라질 수 있는데 본 설계에서의 보안 등급은 '높은', '보통', '낮음'으로 구분하도록 함으로써 사용자 영역에서의 사용 편리성을 제공하도록 하고 있다.

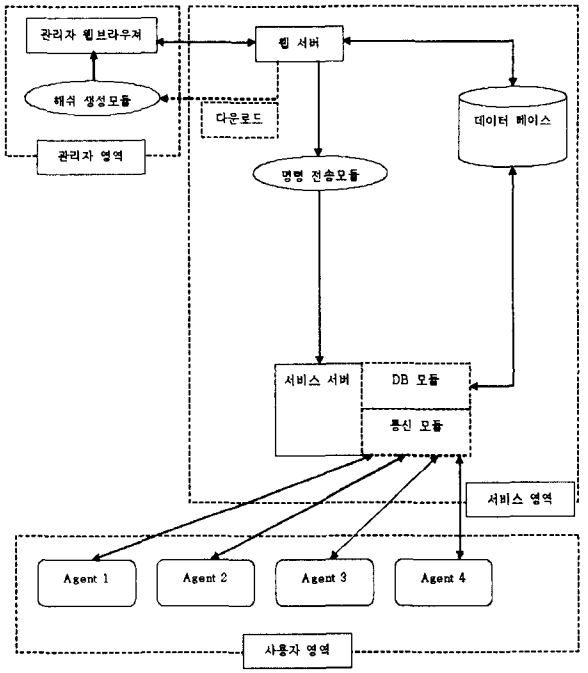
특히, 네트워크 두절과 같은 상황에서도 에이전트는 자체 DB에 해당 탐지 결과를 임시 저장하고 있다가 관리도구와의 통신이 재개되면 즉시 해당 정보를 자동으로 관리서버에 전송하도록 함으로써 한번 탐지된 악성실행코드는 이후 실행시간이 아닌 1차 탐지 단계에서 즉시 필터링하도록 하고 있다.



[그림 2] 악성행위 필터링 수행

3.3 관리서버

관리서버는 에이전트를 업데이트하거나 탐지된 악성실행코드를 바탕으로 새로운 보안 관리 정책을 생성하여 개별 에이전트에 배포하는 역할을 수행하도록 하고 있다. 관리서버는 아래 [그림 3]과 같이 크게 관리자 영역 및 서비스 영역으로 구성되어 있다.



[그림 3] 서비스 다이어그램

관리자는 악성으로 의심되어 에이전트로부터 보고된 실행코드에 대한 이벤트 로그 및 위험도를 바탕으로 최종적으로 악성여부를 판단한다. 악성으로 판단된 실행코드는 알려진 악성실행코드 데이터베이스에 저장하고 동시에 각 에이전트들에게도 해당 패턴 정보 및 보안 정책을 배포하여 추후 같은 실행코드가 재실행되었을 경우 실행코드 실행 이전에 알려진 악성실행코드 데이터베이스를 기반으로 하는 1차 탐지 모듈에서 즉시 탐지하도록 한다.

또한, 관리도구에서는 2차 보안정책에서 사용되는 행위별 위험도를 정의하여 에이전트에게 전송하며, 사용자로부터 보고된 이벤트 로그를 기반으로 해당 포인트에 대한 재조정 등을 수행한다. 이러한 과정을 통해 행위에 대한 위험도는 보다 최적화된 점수를 가지게 되며, 악성실행코드에 대한 패턴 정보, 행위 위험도 정보, 1차 및 2차 보안 정책 등은 실시간으로 관리도구와 에이전트가 공유하도록 설계되어 있다.

4. 결론

실행코드 내의 악의적인 코드 삽입으로 인한 다양한 피해 사례가 지속적으로 보도 되고 있고, 이에 대응할 수 있는 방안이 아직 미비한 실정으로 그 대응책이 시급히 요구되고 있다. 이러한 대응책으로 실행코드 자체의 보안 기능을 강화하는 등의 여러 가지 방법들이 제안되고 있으나 완전한 해결책이 되지 못하고 있는 실정이다.

또한, 각종 상용 제품들은 탐지 기술 자체가 주요한 기술 노하우로서 기술 노출을 꺼리고 있기 때문에 관련 기술을 파악하기도 쉽지 않은 상황이다.

		내용
관리 서버	정책 생성	- 탐지 보고된 패턴 정보를 기반으로 한 보안 정책 생성
	웹 기반 관리	- 관리자 인증 - 로그 정보 조회, 삭제 - 에이전트 관리
	DB 관리	- 악성실행코드 패턴 정보 관리 - 1차 보안 정책 관리 - 2차 보안 정책 관리
	중앙정책관리	- 파일 매니터 관리 - 프로세스 매니저 관리 - 레지스트리 매니저 관리 - 악성실행코드 추가
	탐지 결과 처리	- 탐지 결과 추가/보류 - 탐지 통계 그래프

[그림 4] 관리서버 기능

이에 본 논문에서는 행위 기반의 위험도 산정을 통해 해당 실행코드의 위험도 산정 엔진을 개발하였다. 또한, 이를 기반으로 기존 알려진 악성실행코드뿐만 아니라 알려지지 않은 악성실행코드를 실행시간에 탐지하기 위한 에이전트와 분산된 에이전트들로부터 탐지 정보를 전송받고 이를 관리할 수 있는 관리서버를 설계하였다.

본 제안 방식에서는 악성실행코드 탐지/제거와 해당 보안정책 생성/관리를 분리시킴으로써 탐지 효율성, 관리 편리성, 정책 일관성 및 신속한 보안 정책 생성/배포 등의 효과를 가져올 수 있다. 또한, 위험도 산정을 통한 악성 행위 탐지 기법을 확대 적용함으로써 바이러스 및 웜 등의 다른 악성코드 탐지에도 적용할 수 있을 것이다.

참고문헌

[1] H.Thimbleby, S.Anderson, P.Cairns, "A framework for modeling Trojans and computer virus infection", Computer Journal, Vol. 41 No.7, pp444 ~ 458, 1999

[2] Matthew G.Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J.Stolfo. "Data Mining Methods for detecion of New Malicious Executables" in Proceedings of IEEE Symposium on Security and Privacy. Oakland, CA : May 2001

[3] Roser A. Grimes, "Malicious Mobile Code", O'Relly, 2001

[4] "Security Tradeoffs : Java vs. ActiveX", http://www.cs.princeton.edu/sip/faq/java-vs-activex.html, Princeton Univ.

[5] InterScan AppletTrap data sheet, http://www.antivirus.com/products/isat/datasheets.asp?datasheet=isap

[6] SurfinGate overview : http://www.finjan.com/product_detail2.cfm?product_id=5&type=description

[7] SurfinShield overview : http://www.finjan.com/product_detail2.cfm?product_id=5&type=description

[8] Symantec사, "Mail-Gear Web Client User's Guide", http://www.symantec.com

[9] Finjan사, http://www.finjan.com

[10] 오형근,배병철,김은영,박중길 "정책기반의 새로운 악성실행코드 탐지 기법 설계", WISC2002,pp297-311,2002