실시간 메일 모니터링 시스템

*조미옥[0]　*최경희　**정기현
*아주 대 학 교　정보 및 컴퓨터 공학부, **아주대학교 전자 전기 공학부
jjomi@ccsys.ajou.ac.kr, {khchoi, khchung}@madang.ajou.ac.kr

Realtime mail monitoring system

*Mi-Ok Jo[0]　*Kyung-Hee Choi　**Gi-Hyun Jung
*The Professional Graduate School for Information&Communication Technology,
Ajou University,
** Division of Electrical & Electronics Engineering Ajou University

요　약

메일의 프로토콜 및 메일 서버의 개발이 메일이 송수신 되는 네트워크 환경의 변화와 다양한 기능의 지원을 중점으로 이뤄지고 있어서, 보안 요소는 아직 개발이 미약한 상태이다. 안전한 메일 환경 및 시스템을 구축하기 위해서는 메일을 통한 개인 정보 및 기밀 사항의 유출을 막고, 메일을 통한 공격 및 메일 바이러스 등의 검사가 이뤄져야 한다. 본 논문에서는 메일 서버의 부담을 줄이기 위해서, 메일 서버 독립적으로 구현되는 실시간 메일을 모니터링 시스템에 대해서 논의하고자 한다.

1. 서론

메일이 처음 제안되고, 사용되던 시기에는 네트워크 상에 해커나, 바이러스 등의 위험한 요소가 없었다. 따라서 메일을 주고 받는 프로토콜이나, 메일 서버 프로그램들에 보안적인 요소가 거의 고려되지 않았었다.[1] 하지만 네트워크의 상황이 변하게 되면서 메일을 통해 개인 정보나 기밀 문서가 누출되는 경우가 빈번하게 발생되고 있다. 또 트로이 목마 공격, 버퍼 오버플로우를 사용한 DOS 공격과 같은 메일을 통한 공격 및 메일 바이러스도 그 종류가 급격하게 늘어나고 있다.[2]

이런 문제를 해결하기 위해서 최근 들어 메일 검사 프로그램과 암호화를 응용한 방식의 프로토콜 및 어플리케이션들이 많이 소개되고 있다. Secure mail은 공개키-개인키를 사용해 메일을 암호화해서 전송하는 방식으로 메일을 통한 개인 정보의 누출을 막는 것을 목적으로 하고 있다.[3] 하지만 아직까지는 특정 업체의 클라이언트를 사용하는 사람들에게만 서비스의 제공이 가능하다. procmail은 사용자의 우편함에 저장한 메일들을 대상으로 메일 서버 관리자나 또는 개인 사용자가 정한 규칙에 따라서 메일을 강력하게 필터링하는 툴이다. 하지만 프로그램의 사용법과 필터링 조건을 만드는 방식이 매우 복잡할 뿐만 아니라, 일단 메일 서버에 저장된 다음에 메일을 검사를 하므로, 해당 서버의 유저에게로 수신되는 메일만 검사가 가능하다는 단점이 있다.[4]

위의 방식들의 가장 큰 문제점은 모두 메일 서버에 통합되어 구현되어 있다는 것이다. 현재의 메일서버 프로그램들은 보안적인 요소가 거의 고려되지 않았음에도 불구하고, 다양한 형태와 구조의 메일들을 지원하기 위해서 구조가 매우 복잡해지고, 사이즈도 많이 커졌다. 따라서 메일 서버에 메일의 필터링이나, 암호화를 응용한 보안 기능을 추가하기 위해서는 상당부분을 수정하고, 재구성해야만 한다. 또, 메일 서버의 메일 처리 부담이 더욱 가중되고, 사용자는 특정 메일 서버를 사용해야만 보안성을 보장 받을 수 있다는 제한점이 생긴다.

따라서 본 논문에서는 메일 서버의 부담을 줄이고, 사용자가 메일 서버의 사용에 제한을 받지 않도록 하기 위해서 메일 서버 독립적으로 구현되는 실시간 메일 모니터링 시스템을 제안하고자 한다. 또 기존의 메일 서버 프로그램을 수정을 최소화 시킬 수 있다.

2장에서는 관련 연구로 메일의 전송과정과, 사용자와 메일 서버, 그리고 망을 고려한 메일의 전달 경로에 대해서 살펴본다. 3장에서는 시스템의 구성과, 구현 방법, 또 메일 패킷을 필터링하는 조건에 대해서 살펴본다.

2. 관련연구

2.1 메일의 전송 과정

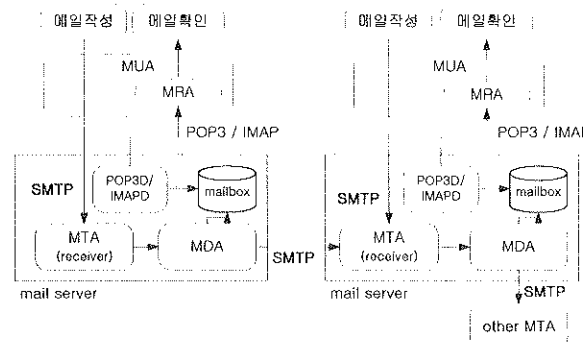

그림 1 메일의 전송 과정

메일을 전송하기 위해서는 사용자가 일반적으로 MUA에 통합되어 있는 메일 작성 프로그램을 사용하거나, 직접 메일 서버의 MUA에서 메일을 작성해야 한다.[5] MUA(Mail User Agent)는 사용자가 메일을 송수신할 때 사용하는 클라이언트 프로그램으로, 마이크로 소프트사의 아웃룩 익스프레스나 리눅스나, 유닉스에서 사용하는 mail 프로그램를 예로 들 수 있다. MUA는 사용자가 작성한 메일을 메일 서버의 MTA로 전송한다. MTA(Mail Transfer Agent)는 인터넷 상에 있는 하나의 컴퓨터로부터 다른 컴퓨터(수신자측의 메일 서버)로 메일을 전달하는 프로그램이다. MTA는 수신된 메일의 수신자가 로컬 메일 서버의 사용자인지, 아니면 다른 메일 서버의 사용자인지를 판단한다. 만약 로컬 사용자인 경우에는 해당 MDA를 통해서 로컬 파일시스템에 있는 사용자의 우편함에 메일을 저장한다. MDA(Mail Delivery Agent)는 메시지를 사용자의 우편함에 쓰기 위해 MTA가 사용하는 프로그램이다. 그리고 로컬 사용자가 아닌 경우에는 별도의 MDA를 사용해서 다른 MTA로 메일을 전달한다. 메일을 수신 받은 수신자의 MTA는 다시 로컬 메일 서버의 사용자인지를 판단 한 후에, MDA를 통해서 파일 시스템의 사용자 우편함에 메일을 저장한다. MTA와 MDA는 sendmail[6]이나

qmail[7]처럼 일반적으로 하나의 메일 서버 어플리케이션에 통합되어 있다.사용자 MUA에서 메일 서버로 메일을 전송할 때와, 메일의 수신자가 로컬 사용자가 아니어서 다른 메일 서버에 있는 MTA로 메일을 전송할 때, SMTP[8]를 사용한다.

사용자가 자신의 메일을 확인 하기 위해서는 MUA에 포함되어 있는 MRA를 사용해야 한다. MRA(Mail Retrieval Agent)는 원격지 서버에 있는 우편함으로부터 사용자의 MUA로 메시지를 가져오는 서비스를 제공하는 프로그램이다. MRA가 메일 서버에 있는 POP3데몬에 접속하면, POP3 데몬은 사용자의 우편함에서 메일을 읽어서, MRA에게 전달한다. 사용자의 우편함으로부터 메일을 읽어올 때는 POP3(Post Office Protocol Version 3)를 주로 사용하지만, 요즘에는 많은 네트워크 지향적인 옵션을 제공하는 IMAP(Internet Message Access Protocol)의 사용이 선호되고 있다.

2.2 메일의 전달 경로에 따른 분류

앞 절에서 살펴본 메일의 전송과정은 망의 구성은 고려하지 않고, 같은 망 안에 사용자와 메일 서버가 존재하는 것을 전제로 하고 있다. 하지만 실제로는 사용자가 서로 다른 망에 있는 메일 서버를 사용해서 메일을 주고 받는 것이 일반적이다. 따라서 실제 메일의 전송과정을 살펴보기 위해서는 망의 구성도 고려되어야 한다.

메일의 전송과정에서 성립되는 SMTP연결을 기준으로, 서로 다른 망 사이의 연결과, 사용자와 메일 서버 사이의 연결에 따라서 메일의 전달 경로를 분류한다.

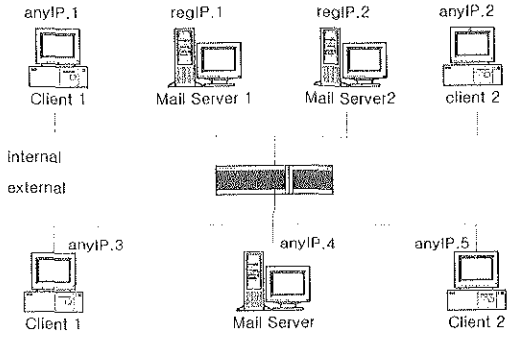

그림 2 네트워크 시스템의 구성 예

망의 구성이 고려된 메일의 전달 과정을 살펴보기 위해서 그림과 같은 구성을 예로 들었다. 내부 망에는 내부 사용자들이 사용할 메일 서버가 2개와, 각각 메일을 주고 받을 수 있는 2개의 사용자 메일 클라이언트 호스트가 있다. 외부 망에도 메일을 주고 받을 수 있는 외부 클라이언트 호스트가 2개 있고, 이 들이 사용하는 메일 서버가 있다. 내부 망의 메일 서버는 메일 보안 시스템에 IP가 등록되어서 관리될 것이라는 의미에서 regIP 1,2를 사용한다. 다른 시스템들은 어떤 IP를 사용할지 알 수 없기 때문에 anyIP로 명시했다. 단, anyIP는 등록되어 있는 내부 IP와 동일한 IP가 존재해서는 안 된다.

SMTP연결이 성립되기 위해서는 SMTP 클라이언트에서 SMTP 서버에 연결을 요청해야 한다. 연결이 성립되면 SMTP 클라이언트에서 SMTP 서버에 메일을 전송하기 위한 과정을 수행하기 위해서 단계별로 request를 보낸다. SMTP 서버에서는 각각의 request에 대해서 반드시 response를 보내야 한다. request와 response 패킷의 IP 주소와 port를 기준으로 SMTP 연결과정에서 서버와 클라이언트를 구분한다. SMTP 서버는 25번 포트를 사용한다.

2.2.1 external network에서 들어오는 메일 (외부사용자(1) ➔ 외부 메일 서버 ➔ 내부 메일 서버(1) ➔ 내부 사용자(1))

가) 외부사용자(1)과 외부 메일서버 사이에 연결되는 SMTP연결이 성립된다. 하지만 외부 네트워크 망 내에서의 연결에 대해서는 메일을 검사하지 않는다.

나) 외부 메일서버가 SMTP client가 되고, 내부 메일서버(1)이 SMTP server가 되어 SMTP 연결이 성립된다.

SMTP client(외부메일서버)		SMTP server(내부 메일서버)	
IP	port	IP	port
anyIP4	any	regIP1	25

2.2.2 외부 망에서 들어와서 내부 망의 메일 서버들 사이에서 포워딩 되는 메일 (외부사용자(1) ➔ 외부 메일서버 ➔ 내부 메일서버(1) ➔ 내부 메일서버(2) ➔ 내부사용자(2))

가) 외부사용자(1)과 외부 메일서버 사이에 연결되는 SMTP연결에 대해서는 메일을 검사하지 않는다.

나) 외부 메일서버가 SMTP 클라이언트가 되고, 내부 메일 서버(1)이 SMTP 서버로 설정되어 SMTP 연결이 성립된다. 2.2.1의 나)번과 같다.

다) 내부 메일서버(1)이 SMTP 클라이언트, 내부 메일서버(2)가 SMTP 서버 역할을 맡아서 SMTP 연결이 성립된다.

SMTP client(내부메일서버 1)		SMTP server(내부메일서버 2)	
IP	port	IP	port
regIP1	any	regIP2	25

2.2.3 내부 망 안에서 전송되는 메일 (내부사용자(1) ➔ 내부 메일서버(1) ➔ 내부사용자(2))

가) 내부 사용자의 MUA와 내부 메일서버(1) 사이에 SMTP연결이 이루어진다. 내부 사용자(1)이 SMTP client가 되고, 내부 메일서버(1)이 SMTP server가 된다.

SMTP client(내부 사용자 1)		SMTP server(내부 메일서버)	
IP	port	IP	port
anyIP1	any	regIP1	25

2.2.4 내부 망 안의 메일 서버들 간에 포워딩 되는 메일 (내부 사용자(1) ➔ 내부 메일서버(1) ➔ 내부 메일서버(2) ➔ 내부 사용자(2))

가) 내부 사용자(1)과 내부메일서버(1)사이의 SMTP 연결은 2.2.3의 가)번의 연결과 같다.

나) 내부메일서버(1)과 내부메일서버(2) 사이의 SMTP연결은 2.2.2의 다)번의 연결과 같다.

2.2.5 내부 망의 메일 서버에서 외부 망의 메일 서버로 전송되는 메일 (내부 사용자(1) ➔ 내부 메일서버(1) ➔ 외부 메일서버 ➔ 외부 사용자(1))

가) 내부 사용자와 내부 메일서버(1) 사이에 SMTP 연결이 이루어진다. 2.2.3의 가)번 연결과 같다.

나) 내부 메일서버(1)과 외부 메일서버 사이에 SMTP 연결이 이루어진다. 내부 메일서버가 SMTP client가 되고, 외부메일서버가 SMTP server가 된다.

SMTP client(내부메일서버)		SMTP server(외부메일서버)	
IP	port	IP	port
regIP1	any	anyIP4	25

2.2.6 내부 망의 메일 서버들 간에 포워딩 된 후 외부 망의 메일 서버로 전송되는 메일 (내부 사용자(1) ➔ 내부메일서버(1) ➔ 내부메일서버(2) ➔ 외부메일서버 ➔ 외부 사용자)

가) 내부 사용자(1)과 내부메일서버(1) 사이에 SMTP 연결이 이루어진다. 2.2.3의 가)번 연결과 같다.

나) 내부메일서버(1)과 내부메일서버(2) 사이에 SMTP 연결이 이루어진다. 2.2.2의 다)번 연결과 같다.

다) 내부메일서버(2)와 외부메일서버 사이에 SMTP 연결이 이루어진다. 2.2.5의 나)번에서 성립되는 내부메일서버(1)을 내부메일서버(2)로 바꾼 것과 같다. 하지만 SMTP client가 register된 IP를 가지고, SMTP server가 any IP를 가지는 상황으로 본다면 같은 연결로 간주될 수 있다.

3. 구현

3.1 시스템의 구성

메일 모니터링 시스템을 구성하기 위해서는, 먼저 망 안으로 들어오거나, 망을 지나가는 모든 패킷을 잡아야 한다. 네트워크 상의 모든 패킷들을 잡는 것은 별도의 디바이스나, 방화벽 루틴을 사용해서 구현할 수 있다. 또는 pcap 라이브러리를 쓸 수도 있다.

두번째로 패킷을 서비스 별로 분류한다. 로컬 메일 서버를 통해서 들어오는 모든 메일을 검사하기 위해서는 먼저 다른 메일 서버의 MDA로부터 SMTP를 통해서 들어오는 메일을 검사해야 한다. 또한 로컬 사용자가 메일의 송신을 위해서 MUA를 통해 전달하는 메일도 검사되어야 한다. 이 때도 마찬가지로 SMTP를 통해서 전달된다. 따라서 잡은

패킷의 목적지나, 출발지 주소의 포트가 SMTP인 경우에는 메일의 전달과 관련된 패킷이라고 간주할 수 있다.

다음으로 메일을 검사하기 위해서, SMTP 패킷들을 메일 별로 재구성 한다. 2.2절의 시스템 구성 예제의 망과 같이 여러 개의 메일 서버가 존재하는 환경에서는 같은 망 내에서는 메일서버 사이에 동시에 여러 개의 메일이 송수신 될 수 있다. 또는 내부 망의 메일 서버는 한 개만 존재하더라도, 내부 망의 여러 사용자들이 동시에 메일을 전송하거나, 외부 망으로부터 동시에 여러 메일이 들어올 수도 있다. 따라서 동시에 여러 개의 메일에 해당하는 SMTP 패킷이 잡힐 수 있기 때문에, 각각의 메일들을 구분할 수 있어야 한다. 메일별로 패킷을 모아서, 재구성된 메일은 디스크 상에 저장된다.

마지막으로 저장된 메일은 메일을 검사하고, 검사 결과에 따라 처리될 수 있는 별도의 프로그램으로 전달된다. 이 프로그램에서는 메일을 여러 조건에 걸쳐서 검사하고, 검사 결과에 따라서 메일이 서버로 전달되는 것을 중단 시킬 수도 있고, 메일 서버에 경고 메시지를 보낼 수도 있다.

본 논문에서는 SMTP 패킷들로부터 메일을 재구성해서 디스크에 저장하는 과정까지를 살펴본다.

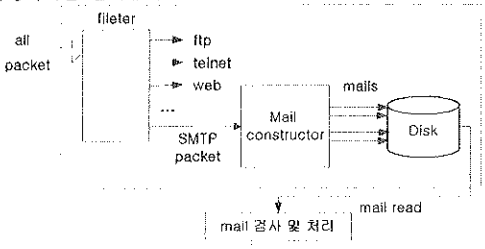

그림 3 전체 메일 보안 시스템의 구성

3.2 SMTP 패킷의 선택 조건

내부 망으로 송수신 되는 메일들을 검사하기 위해서 내부 망의 메일 서버의 IP 주소들은 메일 보안 시스템에 등록되어서 관리되어진다. 따라서 망에서 캡쳐된 SMTP 패킷의 출발지 IP가 등록되어 있는 IP 목록에 속한다면, 내부 메일 서버에서 나오는 패킷이다. 목록에 없는 IP 주소라면 외부 망의 메일 서버나, 내부 망의 사용자들에 의해서 전달되어지는 패킷이다.

메일을 재구성 하기 위해서는 SMTP의 서버에서 오는 패킷들과, SMTP 클라이언트에서 오는 패킷들을 각각 구분해서 메일의 전송 과정을 추적할 수 있어야 한다. SMTP 클라이언트에서 보내는 request 패킷은 등록되어 관리되어지는 메일 서버의 IP 리스트에 속해있는 한 IP와 25번 포트를 목적지 주소로 하고 있다. 마찬가지로 SMTP 서버에서 보내는 response 패킷은 등록된 메일 서버의 IP와 25번 포트를 출발지의 주소로 가지고 있다.

다음으로 여러 개의 메일이 동시에 SMTP를 사용해서 전송될 수도 있으므로, 각각의 메일을 구분할 수 있어야 한다. 먼저 메일 전송을 요청하는 SMTP 클라이언트의 IP와 port를 사용해서 한 메일 서버에 접속한 클라이언트들을 구분할 수 있다. 또한 각각의 메일 서버는 메일 서버 자신의 IP와 port-25번-을 사용해서 구분할 수 있다.따라서 메일들은 각각의 연결에서 성립되어 있는 SMTP 서버의 IP와 port, 그리고 SMTP 클라이언트의 IP와 port의 쌍으로 구분될 수 있다.

구현하고자 하는 메일 보안 시스템의 목적은 내부 메일 서버를 통해서 송수신 되는 메일로 국한되므로, 외부의 유저가 외부의 메일서버와 연결하는 SMTP 연결에 대해서는 검사하지 않는다.

망의 구성을 고려한 메일의 전달과정을 살펴보면, 하나의 메일에 대해서 내부 망 안에서만 2번 이상의 SMTP 연결이 성립되는 경로가 생긴다. 내부 메일 서버간에 포워딩 되는 경우와, 내부의 사용자가 작성한 메일이 외부의 메일서버로 전달되는 경우이다. 따라서 SMTP 연결이 성립될 때마다, 전송되는 메일 패킷을 모두 재구성하게 되면, 같은 메일을 2번 이상 재구성하는 경우가 생길 수 있다. 같은 메일을 중복해서 재구성하고 검사하는 것은 메일 검사 작업 시간의 증가와, 시스템 자원의 낭비를 가져온다. 따라서 2.2에서 살펴본 모든 메일의 전달 경로를 포함해서 내부 메일 서버를 경유하는 모든 메일을 포함하면서, 같은 메일에 대해서는 한번만 검사할 수 있는 SMTP 패킷의 검사 조건을 결정해야 한다.

다음의 조건처럼 request와 response의 IP와 port의 쌍으로 메일을 구분해서 SMTP 패킷을 필터링하는 경우, 모든 메일의 전달 경로에 대해서 한 번씩 만 메일을 구성해서 저장할 수 있다.

request : any IP · {reg. IP}/ any port ➔ reg. IP / 25 port
response : reg. IP / 25 port ➔ any IP · {reg. IP} / any port

3.3 메일의 전송에 따른 SMTP 상태변화

마지막으로 메일을 재구성하기 위해서는 성립되어 있는 SMTP 연결에서 전달되어지는 SMTP request 명령의 상태 변화와 response의 타입을 살펴봐야 한다. request와 response의 상태 변화에 따라서 메일의 전송 과정을 추적할 수 있으며, 메일 정보들을 순서에 맞게 저장할 수 있다. 다음은 SMTP의 request의 명령과, response의 타입에 따른 오토마타이다.

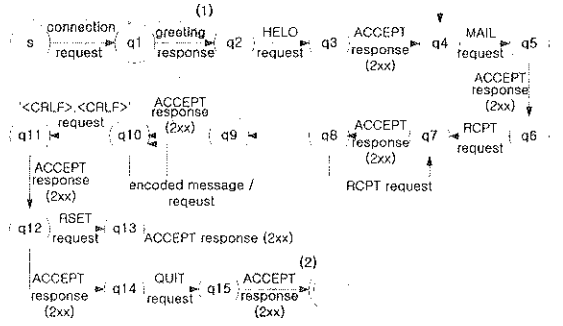

그림 4 메일의 전송에 따른 SMTP 상태 변화 오토마타

각각의 단계에서 처리하고자 하는 메시지 외의 메시지가 들어오면 현재 단계까지의 처리사항을 삭제하고, 메일 구성 작업을 종료 시킨다. 또한 ACCEPT response외에 영구적인 거부나, 일시적인 거부의 response가 발생하면 마찬가지로 메일 구성 작업을 종료 시킨다.

(1)의 과정은 SMTP 클라이언트에서 접속을 시도하고, 서버에서 허용하는 과정이다. 하지만 실제로 메일을 전송하기 위한 시작 단계는 클라이언트에서 보내는 HELO request부터 이므로, (1)의 과정은 생략이 가능하다. (2)의 과정은 SMTP의 연결을 종료하기 위한 과정이다. 하지만 근래의 많은 SMTP 클라이언트 프로그램들은 메일 서버에 QUIT request를 보낸 후에 응답을 기다리지 않고, 메일 송신 과정을 종료 시킨다. 따라서 QUIT request에 대한 응답에 대한 처리 상태는 생략이 가능하다.[9]

4. 결론 및 향후 과제

실시간 메일 모니터링 시스템을 구성하기 위해서, 망의 구성을 고려한 메일의 전달 경로 분석과, SMTP의 분석을 통해서 캡처한 패킷들을 필터링 하고, 메일을 재구성하는 과정을 살펴봤다.

하지만 현재까지 논의된 SMTP 패킷의 분류조건, 메일의 구성 작업은 SMTP를 기반으로 정리되어있다. 하지만 요즘의 많은 MTA 및 MDA에서 ESMTP(Extended Simple Mail Transfer Protocol)를 사용해서 메일을 송수신하고 있다. 따라서 좀더 일반적으로 적용시킬 수 있는 메일 구성 프로그램을 만들기 위해서 ESMTP를 지원하고자 한다.

5. 참고 문헌

[1]. Dove Sill, " Life with qmail" , http://qmail.org/ November 1999
[2]. " Enhanced E-Mail security with procmail" , ftp://ftp.rubyriver.com/pub/jhardin/antispam/procmail-security.html
[3]. secure mail, http://www.securemail.co.kr
[4]. procmail, http://www.procmail.org
[5]. David Wood, " Programming Internet Email" , O' reilly, June 1999, pp.18-20
[6]. sendmail8.11.0, http://www.sendmail.org
[7]. Dan Bernstein, qmail-1.03, http://qmail.org/
[8]. Jonathan B. Postel, " Simple Mail Transfer Protocol" in RFC 821, August 1982
[9]. D.J. Bernstein, " SMTP:Simple Mail Transfer Protocol" , http://cry.yp.to/smtp.html