DoS 공격에 강한 인증 프로토콜의 설계

김민현*, 전동호**, 김순자*

*경북대학교, 전자공학과, **경북대학교 정보보호학과

Design of Authentication Protocol with DoS Resistance

Min-hyun Kim*, Dong-ho Jeon**, Soon-ja Kim*

*Department of Electronics Engineering Kyungpook National Univ.

**Department of Information security Kyungpook National Univ.

요 약

인터넷을 통한 서비스의 이용이 증가하면서 서비스 자체에 대한 보장이 중요해지고 있다. 서비스를 방해하는 공격 중에 DoS(denial of service) 공격이 있다. 이 공격을 막기 위해서는 DoS 공격에 강한 인증 프로토콜이 필요하다. 본 논문에서는 인증 프로토콜 상에서 DoS공격의 취약점을 분석하고, 이 공격에 강한 인증 프로토콜을 설계하기 위한 방안으로 Stateless, 사전검증, 접속제한 등을 제시한다. 그리고 실제 인증 프로토콜에 이것들을 적용해서 DoS 공격에 강한 인증 프로토콜을 설계한다.

I. 서론

인터넷은 생활전반에 활용되고 있으며 인터넷을 통하여 물건을 사고 계약을 하고, 정보를 교환한다. 특히 실시간 증권 거래, 금융 업무 등 서비스의 이용이 증가함에 따라 서비스 이용 자체에 대한 보장이 중요하게 되었다.

서비스 이용을 방해하는 공격 방법에 DoS 공격이 있다. DoS 공격은 서버의 자원을 고갈 시켜 다른 사람이 서비스를 못 받도록 하는 공격이다. 자원이란 하드용량, 메모리 등 컴퓨터나 시스템을 작동하는데 꼭 필요한 요소들을 말한다. 클라이언트들이 서버에 접속할 때마다 연결상태 유지에 필요한 자원들이 할당되는데 제한치를 넘기게 될 때 더 이상 서비스를 제공하지 못하는 상태가 된다. 이런 점을 이용하여 악의적인 공격자가 거짓으로 많은 연결을 시도하여 자원을 다 소비해 버리는 공격 방법이 DoS(denial of service) 공격이다[1].

이 공격의 특징은 간단한 공격 방법이지만 그 효과가 크다는 것이다. 그리고 공격의 모든 과정이 자동화 된 툴들이 나오고 있기 때문에 전문 지식이 없는 사람이라도 이를 이용해서 쉽게 공격을 할 수 있다[2]. 즉 언제라도 누구나 쉽게 공격할 수 있으며 따라서 항상 이런 공격을 당할 위험이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 프로토콜 상에서의 DoS 공격에 대한 취약점과 해결방법을 제안하고, 3장에서는 그것을 실제 인증 프로토콜에 적용하고, 마지막으로 4장에서 결론을 맺는다.

II.기존 프로토콜 분석

1.프로토콜상의 취약점

다음은 일반적인 프로토콜의 진행과정을 나타낸 것이다.

$M \rightarrow V$: Msg_1

서버에 $state_1$저장

$M \leftarrow V$: Msg_2

서버에 $state_1$저장된 상태유지

$M \rightarrow V$: Msg_3

서버에 $state_2$저장

그림 1: 일반적인 프로토콜의 진행 과정

그림에서 M은 사용자를 V는 서비스 제공자를 *Msg*는 프로토콜 상에서 주고받는 메시지를 뜻한다. 프로토콜이 진행됨에 따라 서버(V)는 프로토콜의 진행 과정에 필요한 클라이언트의 identity, 프로토콜의 진행 상황, 중간에 필요한 파라미터의 값 등의 상태 변수(state)를 저장하며 프로토콜이 진행된다[3]. 악의의 공격자가 많은 프로토콜 시작을 요청한 뒤 그 연결을 완전히 끝내지 않는 상태로 놓아둔다면 자원을 소비하게 될 것이고 다른 합법적인 클라이언트들이 서비스를 받지 못하게 된다.

근본적으로 이런 일이 발생 할 수 있는 원인은 처음 프로토콜을 시작할 때 상대에 대한 정확한 인증 없이 처음부터 익명의 사용자에게 자원을 할당하는데 있다. 따라서 이 문제는 클라이언트가 인증이 된다면 누가 DoS 공격을 하는지 확실히 알 수 있고 그 문제에 대해 적절히 대응할 수 있는 것이다. 많은 프로토콜에서는 프로토콜을 시작할 때 인증부터 하고 있다. 하지만 이 인증 프로토콜 또한 앞에서 말한 것처럼 프로토콜을 시작하는 부분에서 인증되지 않은 상대에게 자원을 할당할 뿐만 아니라, 인증 프로토콜과 같은 암호 프로토콜들은 일반적인 프로토콜에 비해 실행하는데 더 많은 계산량과 메모리가 사용되기 때문에 이런 DoS 공격에 대해서는 더욱 취약하다고 할 수 있다. 그러므로 DoS 공격에 강한 인증 프로토콜이 있다면 이것을 통해서 프로토콜 상에서 DoS 공격 문제를 해결할 수 있다.

2. 해결방안

1) Stateless protocol

근본적으로는 DoS 공격의 원인은 인증되지 않은 공격자에게 자원을 할당하는데 그 원인이 있다. 그러므로 할당되는 자원들을 없앰으로써 DoS 공격을 근본적으로 막을 수 있다. 서버에서 할당되는 자원을 저장하지 않고 그것을 원래의 메시지와 함께 주고받는 식인 Stateless 프로토콜로 이것을 구현할 수 있다[3].

$$M \rightarrow V: \quad Msg_1$$
$$M \leftarrow V: \quad Msg_2, state_1$$
$$M \rightarrow V: \quad Msg_3, state_1$$
$$M \leftarrow V: \quad Msg_4, state_2$$

그림 2: Stateless 프로토콜

여기서 프로토콜의 환경 정보가 보통의 메시지의 무결성을 보장하기 위해서 다음 식과 같이 Keyed 해시함수를 이용하여 서명을 대신한다.

$$MAC_{K_s}(T_s, state) = \{state, h_{K_s}(T_s, state)\} \quad (1)$$

K_S는 서버의 비밀키를 나타낸다. 서명값은 공개키 방식이 아닌 서버의 비밀키를 이용한 해시함수를 사용함으로써 계산량과 수행 시간을 줄일 수 있다. T_s 는 서버의 타임 스탬프(time stamp)로서 재전송 공격을 막는 기능을 한다

2) 사전 검증단계

DoS 공격을 계속한다고 의심이 되는 클라이언트에 대해서 프로토콜 진행에 앞서 사전 검증의 단계를 부여한다. 서버가 접속하는 클라이언트에게 질의(puzzle)를 하고 클라이언트는 그것을 풀어서 검증이 될 때만 다음 단계의 과정으로 넘어가도록 한다. 일방적으로 서버 쪽에서 자원을 할당하는 구조를 어느 정도 클라이언트에게 자신의 리소스를 먼저 할당하도록 하는 방식으로 바꾼다[4,5].

puzzle은 다음과 같이 구성된다.

$$h(ID_M, N_S, N_M, X) = 00\cdots\cdots00Y$$

ID_M은 클라이언트의 identity이고 h는 해시함수를 N_s 와 N_c 는 각각 서버와 클라이언트가 생성하는 랜덤 변수를 나타낸다. Y는 X를 뺀 나머지인자로 해시한 값에서 k만큼의 비트를 뺀 나머지이다. '0'의 개수는 문제의 난이도 k를 나타내는 것으로 서버 쪽에서 이것을 정함으로써 문제의 어렵고 쉬움을 결정할 수 있다. 평소에는 k가 0으로 잡혀 있어 계산 없이 진행된다. 서버는 단지 문제의 난이도 k와 N_s만 결정하면 되고, 클라이언트마다 ID_M과 N_c가 다르므로 각각의 클라이언트에게 마다 문제는 다른 문제가 된다. 클라이언트는 받은 값에서 전사공격(brute force)방법으로 X를 구한다. 해시함수를 사용하기 때문에 거의 모든 하드웨어에 적용이 가능하고 계산이 다른 것에 비해 빠르다.

3) 접속제한

공격자가 무수히 많은 접속을 할 가능성이 있다. 악의적인 과도한 접속은 통신망의 대역폭을 차지하게 되어 다른 이용자의 서비스 이용을 막는 것을 방지한다. 그러므로 접속의 한도를 적당히

주어 필요 이상의 접속 시도는 막도록 한다.

III. 프로토콜 설계

표1은 본 논문에 이용된 기호와 그 에 대한 설명을 나타낸다.

표 1: 사용된 기호와 그에 대한 설명

기호	설명
M, V	사용자, 서비스 제공자
ID_M	M의 아이덴터티(identity)
k, l	보안 수준 변수, 접근제한 변수
N_V, N_M	사전검증에 사용되는 랜덤변수
$state_V$	V의 프로토콜 진행에 필요한 설정값
$Cert_E$, T_E	E 객체의 인증서, 타임스템프
K_V	서비스 제공자의 비밀키
K_{MV}	M과 V사이에 만들어지는 세션키
r_V	V에서만드는 랜덤변수
$h()$	해시함수
$E_K()$	키 값이 K인 대칭키 암호함수
$Sig_M()$	M의 서명값
α_T, IV, chd	지불 프로토콜에 사용되는 변수
X	사전검증단계에서 클라이언트가 서버에게 보내는 값
m, v	M과 V의 비밀키
MAC_{K_V}	2장 2절에서 정의한 (1) 식

다음 그림은 일반적인 AsPeCT프로토콜에 대해 나타낸 것이다.

$M \rightarrow V: g^m, ID_M$

$M \leftarrow V: r_V, h(K_{MV}, r_V), chd, T_V, Cert_V$

$M \rightarrow V: E_{K_{MV}}[Sig_M\{h(g^m, g^v, r_V, ID_V, chd, T_V, \alpha_T, IV)\}, Cert_M, \alpha_T, IV]$

$sessionkey: K_{MV} = h((g^m)^v, r_V)$

그림 3: AsPeCT프로토콜

AsPeCT 프로토콜은 공개키 인증 기반으로 하여 상호인증과 프로토콜 내에 지불과 관련된 두 부분으로 이루어지는 프로토콜로 서비스 이용에 대한 사용자 인증과 부인 방지 등의 보안 특성을 가진다[6,7]. 인증 후 지불 프로토콜의 수행이라는 일반적인 서비스 수행 형태를 가지므로 이 프로토콜을 기반으로 설계하였다.

프로토콜의 설계 목표는 다음과 같다.

- 기존의 프로토콜에 많은 수정 없이 DoS 공격에 강한 프로토콜을 설계한다.
- 다른 프로토콜 적용을 쉽게 한다.
- 기존의 하드웨어에서도 돌아갈 수 있도록 한다.
- 연산량을 최소로 한다.

1. 제안된 프로토콜

제안된 프로토콜은 다음과 같다.

(1) $M \rightarrow V: g^m, ID_M$

(2) $M \leftarrow V: r_V, h(K_{MV}, r_V), chd, T_V, Cert_V, MAC_{K_V}(T_V, state_V)$

(3) $M \rightarrow V: E_{K_{MV}}[Sig_M\{h(g^m, g^v, r_V, ID_V, chd, T_V, \alpha_T, IV)\}, Cert_M, \alpha_T, IV], X, N_C, MAC_{K_V}(T_V, state_V)$

$state_V: =\{ID_M, K_{MV}, T_V, k, N_S, l\}$

$sessionkey: K_{MV} = h((g^m)^v, r_V)$

그림 4: 제안한 인증 프로토콜

(2)과정에서 클라이언트의 요청을 받았을 때 프로토콜에 필요한 세션키, 타임 스템프(time stamp), N_C, ,접근 제한 변수, 보안 수준 변수 등의 설정데이터들을 서버에 직접 저장하지 않고 서명하고 암호화해서 직접 클라이언트에게 보냄으로써 서버는 stateless 상태의 프로토콜을 유지할 수 있게 된다. (3) 과정은 클라이언트가 확실히 세션키를 가졌는지 인증되는 단계로 프로토콜 설정 변수로 사전 인증단계에서 계산된 X와 계산에 필요한 랜덤 변수 N_C 가 더 붙여서 보내진다. K_V 는 사전검증에 사용되는 비밀키로써 주기적으로 새로 생성하여 사용한다.

평소에는 보안 수준변수가 0으로 잡혀 있어 계산 없이 프로토콜이 진행된다. 보안 정책에 따라 클라이언트가 공격자로 의심된다면 보안 수준변수를 높여서 클라이언트가 먼저 사전검증 단계를 해

결한 인증 프로토콜 단계로 넘어가게 한다. 사전 검증이 있는 경우 (3)에서 넘어온 메시지에서 먼저 X와 N_C 로 검증한 후 나머지 파라미터를 이용하여 상호 인증을 마무리한다. 같은 패킷이 연속적으로 전송될 경우가 있는데 이때마다 접속제한 변수를 계속 증가 시켜 기준 이상 넘을 경우 다음 과정으로 넘어가지 않고 그 패킷을 폐기시킨다.

이상의 과정을 다 거친 후 인증이 완료되면 지금까지 저장하지 않았던 환경 설정들을 저장하고 다음 단계의 프로토콜을 실행한다.

2. 프로토콜 분석

인증이 완전히 이루어지기 전까지 부분적으로 stateless 프로토콜을 사용하였고 시스템상에 필요에 따라 파라미터 속에 들어있는 랜덤변수와 보안변수들을 이용하여 사전검증작업을 적용할 수 도 있다. 접근 제한 변수로 너무 많은 접속하는 것을 제한할 수 있다.

기존의 프로토콜에서 $Sign_{K_S}(state_V)$ 만을 첨가하는 형태여서 기존의 프로토콜들에서 수정이 용이하다. 그리고 이 함수에 사용되는 해시함수는 거의 모든 종류의 하드웨어에서 쉽게 구현될 수 있다. 프로토콜 설정변수를 다른 패킷과 보내는 것 외에는 기존의 프로토콜에서 변화 된 것이 없으므로 기존의 프로토콜의 성질을 그대로 가지면서 DoS에 강한 프로토콜이 됨을 알 수 있다.

또한 공격에 의해 시스템이 다운되었다가 다시 복귀할 때에도 프로토콜 환경 정보를 가진 패킷을 이용하여 바로 다음 단계로 넘어갈 수 있기 때문에 보통의 프로토콜에 비해 그 복귀가 빠르다는 장점을 가진다.

IV. 결론

기존에 프로토콜을 설계할 때는 보통 DoS 공격에 대한 고려를 거의 하지 않았었다. 하지만 서비스 자체에 대한 중요성이 강조되면서 이런 점을 꼭 고려하여 프로토콜을 설계해야 한다.

프로토콜의 처음 시작 부분에서 자원의 인증이 확실히 되지 않은 사용자에게 할당되는 것이 DoS 공격을 당할 수 있는 취약점이다. 본 논문에서는 그것을 해결할 수 있는 방법으로 자원 할당 이전에 사전 검증을 하고 접속을 제한하는 방법 등을 제안하였고 실제 인증 프로토콜에 그것을 적용하여 설계하였다.

최근 DoS 공격 방법이 점점 발달하고 다양해지고 있으므로 보다 효과적이고 다양한 공격을 막을 방법들이 연구되어야 한다.

참고문헌

[1] CERT Coordination Center,"Denial of Service Attack," http://www.cert.org/tech_tips/denial_of_service.html

[2] CERT Coordination Cenrter,"Trends in Denial of Service Attack Technology," October 2001

[3] T.Aura and, P.Nikander, "Stateless Protocol," In ICICS'97,LNCS 1334. Springer-Verlag,1997

[4] T. Aura, and P.Nikander, J.Leiwo, "DOS-resistant authentication with Client puzzles", In Proc. Security Protocols Workshop, 2000

[5] J. Leiwo, "Towards Network Denial of Service Resistant Protocol,"SEC-2000, pp. 301-310, 2000

[6] G. Horn, B. Preneel, "Authentication and Payment in Future Mobile System," ESORICS '98, LNCS 1485, Springer-Verlag, 1998

[7] K.M. Matin and C.J. Mitchel, "Evaluation of authentication protocols for mobile environment value-added service,"draft, 1998

[8] J. Mirkovic, J. Martin and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms," Computer Science Department Univ. of California, Los Angeles Technical report #020018

[9] T. Aura. "Authorization and availability - aspects of open network security." Research Report A64, Helsinki Univ. of Tech., Dep. of Computer Science and Engineering, Lab. for Theoretical Computer Science, Espoo, Finland, Nov. 2000. Doctoral dissertation.

[10] B. Todd,"Distributed Denial of Service Attack," http://www.opensourcefirewall.com/ddos_whitepaper_copy.html, 18 Feb. 2000

[11] P. Eronen,"Denial of service in public key protocols,"Helsinki Univ. of Technology's Seminar on Network Security,course Tik-110.501, fall 2000