CSP/FDR을 이용한 RADIUS 프로토콜 분석[1)]

김일곤°, 최진영
고려대학교 컴퓨터학과
{igkim°, choi}@formal.korea.ac.kr,

The Analysis of RADIUS protocol using CSP/FDR

Il-Gon Kim°, Jin-Young Choi,
Dept of Computer Science & Engineering, Korea University

요약

무선 인터넷의 활성화와 더불어, 사용자 인증(Authentication), 권한 부여(Accounting) 그리고 자원 사용(Accounting)의 세가지 AAA 서비스를 효율적으로 제공하기 위해 RADIUS와 같은 AAA 프로토콜들이 사용되고 있다. 본 논문에서는 Casper와 CSP를 이용하여 RADIUS 프로토콜을 수행 동작을 명세하고, 모델 체킹 도구인 FDR을 사용하여 RADIUS 프로토콜의 안전성을 분석하고자 하였다.

1. 서론

무선 인터넷의 활성화와 더불어, 사용자인증(Authentication), 권한 부여(Accounting) 그리고 자원 사용(Accounting)의 세가지 AAA 서비스를 효율적으로 제공하기 위해 RADIUS[1][2]와 DIAMETER[3]와 같은 프로토콜들이 제안되고 있다. RADIUS 프로토콜은 처음 IETF의 AAA Working Group[4]에 의해 처음 제안되어졌고, Livingston Enterprise에 의해 최초로 개발되어졌으며, 상용 제품 뿐만 아니라 프리웨어 제품도 인터넷에서 다운 받을 수 있을 만큼, 매우 범용적으로 사용되어 오고 있는 인증 프로토콜 이다. 하지만 몇 가지 보안상 문제점들이 밝혀지고[5][6], 또 다양한 통신 환경에서 사용자 인증을 지원하기 위해 DIAMETER 프로토콜이 제안되고 있다. RADIUS와 같은 보안 프로토콜의 안전성을 분석하는 일은 사용자와 개발자 모두에게 안전성과 신뢰성을 제공할 수 있는 매우 중요한 과제이다. 따라서, 보안 프로토콜 구현하기 전에 설계 단계에서부터 해당 프로토콜의 안전성을 분석하기 위한 다양한 연구가 진행되어 왔다.

이런 연구는 크게 정리 증명과 모델체킹 기법으로 나누어 볼 수 있다. 첫번째 방법의 경우, BAN[7], GNY[8]와 같은 보안 로직을 이용하여 정해진 규칙에 따라 상호 호스트간의 신뢰성을 증명하게 된다. 두번째 방법의 경우, 해당 프로토콜의 인증 동작을 정형 명세 언어로 설계 한 후, 다양한 보안 속성을 만족시키는지 체크하게 된다. ESTELLE, Murphi, NRL Protocol Analyser[9]와 FDR[10][11]은 위와 같은 방법을 이용하게 된다. 특히 FDR을 이용한 모델 체킹 기법은 보안 프로토콜의 안전성을 분석하기 위해 널리 사용되어오고 있는 방법으로, 그 효율성을 인증 받고 있다.

본 논문에서는 FDR을 이용하여 RADIUS 프로토콜의 문제점을 찾아낼 뿐만 아니라, 이와 같은 수학적 이론 모델에서 발견된 보안 문제점과 실제 구현 모델에서 발생할 수 있는 취약점을 비교하고자 한다.

1) 본 연구는 과학기술부 원자력연구개발사업중 원전계측제어시스템 연구개발 사업 위탁연구 과제로 수행되었음

본 논문의 나머지 부분은 다음과 같이 구성되어 있다. 제 2장에서는 RADIUS 프로토콜에 대해 간략히 소개하고, 제 3장에서는 프로토콜을 명세하고 검증하기 위한 CASPER[12], CSP[13] 언어와 FDR 도구에 대해 각각 설명하고, 제 3장에서는 정형 검증 기법을 이용해 RADIUS 프로토콜을 분석한 결과를 보여주고, 마지막으로 제 5장에서는 결론 및 향후 연구 방향을 제시하고자 한다.

2. RADIUS(Remote Authentication Dial In User Service)

RADIUS 프로토콜에서 사용되는 메시지는 기본적으로 TCP 대신 UDP 프로토콜을 사용하며, Access-Request, Access-Accept, Access-Reject, Accounting-Request 등과 같은 패킷 형태로 전달되게 된다. 또한 RADIUS 프로토콜은 다양한 인증 방식을 지원하며, 가장 일반적으로 사용되는 인증 방식은 PAP(Password Authentication Protocol)와 CHAP(Challenge Handshake Protocol)로 구분할 수 있다.

PAP와 CHAP의 가장 큰 차이점은 PAP는 사용자 인증을 위해 Access-Request 패킷에 사용자 이름과 패스워드를 함께 전송하지만, CHAP의 경우에는 사용자 패스워드만 전송한다는 점이다. RADIUS 제품의 인증방식은 각각의 밴더의 특성에 따라 약간의 차이점이 존재 하기 때문에, 본 논문에서는RADIUS의 RFC 문서와 FreeRADIUS 제품을 기준으로 하였다.

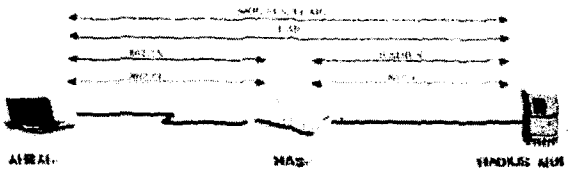

그림 1. RADIUS 프로토콜의 통신 환경

그림 1은 RADIUS 프로토콜의 통신 환경을 보여 주고 있다.

3. Casper, CSP 와 FDR

3.1 Casper(A Compile for the Analysis of Security Protocols)

기존의 CSP를 이용하여 보안 프로토콜의 동작을 명세하고 분석하는 방식은 매우 복잡하여 CSP 명세 전문가 조차도 사소한 실수나 애러를 야기시킴으로써, 보다 정확한 분석을 어렵게 만들었었다. 이에 따라 보안 프로토콜을 보다 쉽게 명세할 수 있고, 자동으로 CSP 명세 소스를 생성할 수 있도록 개발된 도구가 바로 Casper 이다. Casper를 이용하여 보안 프로토콜에서 사용되는 각종 키 타입, 동작 절차, 보안 속성, 공격자 모델등을 8개 영역으로 나누어 명세할 수 있다.

3.2 CSP(Communicating Sequential Process)

CSP는 프로세스 알제브라 언어로서, 병렬성을 갖는 통신 프로토콜의 동작을 효율적으로 명세하기 위해 제작되어졌다. 처음에는 일반적인 통신 프로토콜과 제어 시스템을 명세하기 위해 사용되어졌지만, 점차 보안 프로토콜을 명세하기 위한 영역으로도 확대되어 오고 있다. CSP에서 제공하는 pure synchronization(|||)과 Interleaving parallelism(||) 개념을 사용하여 분산 시스템 환경에서 동작하는 클라이언트 서버, 공격자 모델을 정형적으로 표현할 수 있다는 장점을 갖고 있다. 예를 들면, 분산시스템 환경에서 동작하는 보안 시스템은 다음과 같이 간략히 표현될 수 있다.

SYSTEM = CLIENT1 ||| CLIENT2||| SERVER ||
INTRUDER

3.3 FDR(Failure Divergence Refinements)

FDR은 모델체킹 도구로서, CSP 언어로 구현된 보안 모델이 safety, authentication과 같은 보안 속성을 만족시키는지 체크하게 되며, 만일 해당 속성을 만족시키지 않을 경우 반례를 보여주어, 어떤 공격 시나리오가 가능한지 분석하도록 도와준다. 기본적으로 trace, failure, failure and divergence 이 세가지 동치성 검사 방법을 제공하며, trace는 시스템의 safety, faiure는 deadlock, failure and divergence는 livelock을 검사하기 위해 각각 사용되어 진다.

4 RADIUS 프로토콜 명세 및 분석

4.1 RADIUS 프로토콜 명세

본 논문에서는 RADIUS의 PAP 인증방식을 기준으로 프로토콜의 동작을 명세하였다. 그림 2는 PAP 인증 방식에 대한 간략한 자연어 명세를 보여 주고 있다.

1. User -> NAS : 사용자 이름과 패스워드 전송
2. NAS -> RADIUS : Access-Request 패킷 전송
3. RADIUS -> NAS : Access-Request 패킷 분석후, 적합한 사용자인 경우 Access-Accept, 그렇지 않은 경우, Access-Reject 패킷 전송
4. NAS -> User : RADIUS 서버에 대한 접속 허가

그림 2. PAP의 자연어 명세

그림 3은 위에서 언급한 PAP 인증 방식을 Casper로 명세한 부분으로, 8개의 섹션 영역중에서 #free variable과 #protocol description, #intruder information 만 보여주고 있다.

```
#Free variables
A, B : Agent
S: Server
rauth : Nonce
ServerKey : Agent -> ServerKeys
passwd, shared : SessionKey
f : HashFunction
InverseKeys = (f,f),(passwd,passwd),(shared,shared)

#Protocol description
0.    -> A : B
1. A -> S : A, passwd
2. S -> B : S, rauth, passwd(+)f(rauth, shared)
3. B -> S : B, f(rauth,shared)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Alice, Bob, Mallory, Sam
Km}
```

그림 3. Casper를 이용한 PAP 명세

A, S, B는 각각 사용자 클라이언트, NAS(Network Access Point), RADIUS 서버를 나타낸다. 또한 rauth는 NAS에서 RADIUS 서버에 인증 요청을 하는 Reqeust Authentication 정보로 난수로 표현된다. passwd는 /etc/raddb/users에 저장되어 있는 키로 세 호스트들이 모두 함께 공유하고 있는 패스워드를 의미한다. f는 MD5 해쉬함수를 나타내며, (+)는 배타 논리합 연산자를 나타내고 있다.

4.2 RADIUS 프로토콜 검증

FDR을 이용하여 RADIUS 프로토콜의 안전성을 검증하기 위해서는, RADIUS 프로토콜인 만족해야할 보안 속성을 명시해 주어야 한다. RADIUS 만족해야할 보안 속성은 크게 두가지로 분류할 수 있다. 첫째는 비밀성이고 둘째는 인증이다. RADIUS 프로토콜에서 만족해야할 비밀성을 다음과 같이 5부분으로 나타내 보았다. Secret(A, passwd, [S])

Secret(S, rauth, [B])
Secret(B, rauth, [S])
Secret(S, shared, [B])
Secret(B, shared, [S])

위에서 첫번째 표현식에 대한 의미를 살펴보면, "S는 rauth 정보를 B하고만 알고있다고 믿는다" 고 해석할 수 있다. 나머지 4개의 표현식도 이와 유사한 의미를 갖으므로 추가 설명은 생략하도록 하겠다. 인증의 경우는 다음과 같이 2개 부분으로 나타내 보았다.

Agreement(S,B,[rauth,shared,passwd])
Agreement(B,S,[rauth,shared,passwd])

마찬가지로, 위에서 첫번째 표현식에 대한 의미를 살펴보면, "S는 rauth, shared의 정보를 통해 B에 인증을 받는다"고 해석 할 수 있다. 나머지 부분도 마찬가지로 해석된다.

FDR 모델 체커 도구를 이용해, 위에서 언급한 비밀성, 인증 속성을 RADIUS 프로토콜이 정확히 만족하고 있는지 확인해 보았다. 그 결과 비밀성, 인증 요구사항이 만족되지 않았음을 볼 수 있었다. 다음은 FDR 도구에서 보여주고 있는 두 가지 반례를 나타내고 있다.

비밀성
env.Alice.(Env0,Bob,<>)
signal.Claim_Secret.Alice.Passwd.{Sam}
leak.Passwd

인증
env.Alice.(Env0,Bob,<>)
intercept.Alice.Sam.(Msg1,Sq.<Alice,Passwd>,<>)
fake.Mallory.Sam.(Msg1,Sq.<Mallory,Passwd>,<>)
intercept.Sam.Bob.(Msg2,Sq.<Sam,Rauth,Xor.(Hash.(f,<Rauth,Shared>),Passwd)>,<>)
signal.Commit2.SERVER_role.Sam.Bob.Rauth.Shared.Passwd

첫번째, 비밀성 요구사항에 대한 반례를 분석해 보면, 일반 사용자가 NAS에 사용자 패스워드를 전송할 때 암호화되지 않은 통신 채널을 사용하여 패스워드 스니핑이 가능하다는 사실을 알 수 있다. 이 문제점을 해결하기 위해서는 그림 1에서 보는 바와 같이, EAP[13]와 같은 암호화된 통신 채널을 사용하여 사용자와 NAS간의 통신 신뢰성을 향상시키면 된다. 두번째, 인증 요구사항에 대한 반례를 분석해 보면, 다음과 같은 공격 시나리오를 알 수 있다.

1. 사용자 → 공격자(NAS) : passwd
2. NAS → 공격자(RADIUS) : rauth, passwd (+) f(rauth, shared)
3. 공격자(RADIUS) → NAS : f(rauth,shared)

이러한 공격 형태가 발생하는 이유는 앞에서 언급한 바와 같이 사용자 패스워드 정보가 유출되기 때문이다. 즉 passwd를 암호화하여 전송하여 공격자에게 패스워드 정보가 유출되지 않는다면, 이런 문제점을 발생하지 않게 된다. passwd가 암호화되어 전송되는 경우에 대해 검증한 결과 위와 같은 공격은 발생하지 않음을 확인 할 수 있었다.

5. 결론 및 향후 연구 방향

본 논문에서는 Casper, CSP/FDR을 이용하여 RADIUS 프로토콜의 안전성을 분석해 보았다. 분석해 본 결과, 다른 기타 보안 프로토콜에서와 마찬가지로, 각 호스트간에 안전한 통신 채널을 형성하는 것이 무엇보다 중요한 보안 사항임을 확인할 수 있었다. 따라서 RADIUS의 보안성을 향상시키기 위해서는 EAP와 같은 보안 프로토콜을 사용하여 우선적으로 무선 통신만의 안전성을 향상시키는 것이 보다 사용자의 안전성을 향상시키는 방법일 것이다. 향후 연구방향으로는 FDR과 같은 모델 체킹을 이용한 분석 방법은 보안 프로토콜 디자인상에서의 보안 문제점에 중점을 두고 분석하기 때문에, man-in-the-middle 공격을 쉽게 찾아 주지만, 보안 알고리즘의 취약점에 대해서는 분석해내지는 않는다. 실제 RADIUS 제품에서는 MD5 해쉬 함수의 취약점으로 인해 대신 SHA-1 함수를 사용할 것을 권고하고 있다. 따라서, 보다 다양한 형태의 취약점을 찾아내기 위해서는 여러 가지 공격 패턴을 찾아 낼 수 있는 방법을 추가시켜야 할 것이다.

6. 참고문헌

[1] RFC 2138, Remote Authentication Dial In User Service(RADIUS).
[2] Jonatban Hassell, *O'Reilly RADIUS book*, 2002
[3] Pat R. Calhoun, "DIAMETER Base Protocol", draft-calhoun-diameter-strong-crypto-03.txt IEFT work in progress. Apr. 2000
[4] AAA Working Group Internet-Draft Category Standards Track <draft-ietf-aaa-diameter-12.txt>
[5] Joshua Hill, An Analysis of the RADIUS Authentication Protocol, InfoGard Laboratories
[6] RADIUS Protocol and Implementation Weakness, BUGTRAQ.
[7] M. Abadi, M. Burrows, and R. Needham. A Logic of Authentication. In Proceeding of the Royal Society, Series A, 426, 1871, pages 233-271, December 1989
[8] Li Gong, Roger Needham, Raphael Yahalom, Reasoning about Belief in Cryptographic Protocols, *Proceedings 1990 IEEE Symposium on Research in Security and Privacy.*
[9] Philip E. Varner, Formal Methods as an Environmental Catalyst for Emergent Security in System Design and Construction, December 12, 2002.
[10] Gavin Lowe, Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR.
[11] Formal Systems(Europe) Ltd. Failure Divergence Refinement-FDR2 User Manual, Aug. 1999.
[12] G. Lowe. Casper: A compiler for the analysis of security protocols. 10th IEEE Computer Security Foundations Workshop, 1997.
[13] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, 1985.