Kerberos 개요 - 개방형 네트워크 시스템용 인증 서비스

목차

소개

개방형 네트워크 컴퓨팅 환경에서는 워크스테이션을 신뢰할 수 없어 네트워크 서비스에 대한 사용자를 올바르게 식별할 수 없습니다.Kerberos는 신뢰할 수 있는 서드파티 인증 서비스를 사용하여 사용자의 ID를 확인하는 대체 접근 방식을 제공합니다.이 백서에서는 MIT의 Project Athena에 대해 구현된 Kerberos 인증 모델에 대한 개요를 제공합니다.클라이언트, 서버 및 Kerberos에서 인증을 얻기 위해 사용하는 프로토콜에 대해 설명합니다.또한 필요한 데이터베이스의 관리 및 복제에 대해서도 설명합니다.사용자, 프로그래머 및 관리자가 볼 수 있는 Kerberos 뷰에 대해 설명합니다.마지막으로, 더 큰 Athena 그림에서 Kerberos의 역할은 사용자 인증에 현재 Kerberos를 사용하는 애플리케이션 목록과 함께 제공됩니다.Kerberos 인증을 Sun Network File System에 추가하는 것은 Kerberos를 기존 애플리케이션과 통합하는 사례 연구라고 설명합니다.

Kerberos 작성자

- Jennifer G. Steiner, Project Athena, Massachusetts Institute of Technology, 케임브리지, MA 02139, steiner@ATHENA.MIT.EDU
- 클리포드 뉴먼, 컴퓨터 과학부, FR-35, 워싱턴 대학교, 시애틀, 98195, bcn@CS.WASHINGTON.EDU.Clifford Neuman은 Kerberos의 설계 및 초기 구현 단계에서 Project Athena 직원의 일원이었습니다.
- Jeffrey I. Schiller, Project Athena, Massachusetts Institute of Technology, Cambridge, MA 02139, jis@ATHENA.MIT.EDU

Kerberos 소개

이 문서에서는 Miller 및 Neuman에서 설계한 인증 시스템인 Kerberos에 대해 간략하게 설명합니다.MIT의 Project Athena에서 이를 사용한 경험에 대해 설명합니다.동기에 대한 섹션에서 오픈 네트워크에 새로운 인증 모델이 필요한 이유와 요구 사항에 대해 설명합니다.Kerberos란 무엇입니까?이 섹션에서는 Kerberos 소프트웨어의 구성 요소를 나열하고 인증 서비스 제공 시 상호 작용하는 방법을 설명합니다.Kerberos Names 섹션에서 Kerberos 이름 지정 체계에 대해 설명합니다.

Kerberos 작동 방식은 Kerberos 인증의 구성 요소(티켓 및 인증자)를 나타냅니다.이렇게 하면 두 인증 프로토콜에 대한 토론이 이루어집니다.Kerberos에 대한 사용자의 초기 인증(로그인하는 것과 유사), 잠재적 소비자와 잠재적 네트워크 서비스 생산자의 상호 인증을 위한 프로토콜.

Kerberos에는 클라이언트에 대한 정보 데이터베이스가 필요합니다.[Kerberos 데이터베이스] 섹션에서는 데이터베이스, 데이터베이스 관리 및 수정 프로토콜에 대해 설명합니다.Kerberos From the Outside Looking In 섹션에서는 사용자, 애플리케이션 프로그래머 및 관리자에 대한 Kerberos 인터페이스에 대해 설명합니다.Bigger Picture(더 큰 그림) 섹션에서 Project Athena Kerberos가 Athena 환경의 나머지 영역에 어떻게 부합하는지 설명합니다.또한 여러 Kerberos 인증 도메인 또는 영역의 상호 작용에 대해서도 설명합니다.이 경우 Project Athena Kerberos와 MIT의 Laboratory for Computer Science에서 실행되는 Kerberos의 관계는

Issues and Open Problems(문제 및 미해결 문제) 섹션에서 미해결 과제와 문제를 언급합니다.마지막 섹션에서는 Project Athena의 현재 Kerberos 상태를 보여줍니다.부록에서는 원격 파일 시스템에 액세스하려는 사용자를 인증하기 위해 네트워크 파일 서비스에 Kerberos를 적용하는 방법에 대해 자세히 설명합니다.

Kerberos 개념

본 백서에서는 애매하거나 독자에게 처음이거나 다른 곳에서 다르게 사용되는 용어를 사용합니다.아래에는 이러한 용어의 사용을 명시하고 있습니다.

사용자, 클라이언트, 서버—사용자는 프로그램 또는 서비스를 사용하는 사람을 의미합니다.고객이 뭔가를 사용하기도 하지만 반드시 사람은 아니다.프로그램일 수 있습니다.네트워크 애플리케이션은 두 부분으로 구성된 경우가 많습니다.한 컴퓨터에서 실행되어 원격 서비스를 요청하는 프로그램과 원격 시스템에서 실행되고 해당 서비스를 수행하는 다른 프로그램이를 애플리케이션의 클라이언트 측 및 서버 측이라고 합니다.클라이언트가 사용자를 대신하여 서버에 연결하는 경우가 많습니다.

사용자 또는 네트워크 서버인 Kerberos 시스템을 사용하는 각 엔티티는 Kerberos 서비스를 사용하기 때문에 하나의 의미에서는 클라이언트입니다.따라서 Kerberos 클라이언트와 다른 서비스의 클라이언트를 구분하기 위해 principal이라는 용어를 사용하여 해당 엔티티를 나타냅니다.Kerberos 주체는 사용자 또는 서버일 수 있습니다.(다음 섹션에서는 Kerberos 주체 이름을 설명합니다.)

Service vs. Server(서비스 대 서버) - 서비스를 수행할 일부 작업의 추상 사양으로 사용합니다.이러한 작업을 수행하는 프로세스를 서버라고 합니다.특정 시간에 지정된 서비스를 수행하는 여러 서버(일반적으로 다른 시스템에서 실행)가 있을 수 있습니다.예를 들어, Athena에는 각 시간 분할 시스템에서 실행되는 BSD UNIX 로그인 서버가 하나씩 있습니다.

Key, Private Key, Password(키, 개인 키, 비밀번호) - Kerberos는 개인 키 암호화를 사용합니다.각 Kerberos 주체에 많은 수의 개인 키가 할당됩니다. 이 개인 키는 해당 주체 및 Kerberos에만 알려져 있습니다.사용자의 경우 개인 키는 사용자의 비밀번호에 적용되는 단방향 기능의 결과입니다.우리는 키를 개인 키의 약어로 사용합니다.

자격 증명—안타깝게도 이 단어는 Sun Network File System과 Kerberos 시스템 모두에 대해 특별한 의미를 갖습니다.NFS 자격 증명 또는 Kerberos 자격 증명을 의미하는지 명시적으로 설명하며, 그렇지 않으면 이 용어가 일반 영어 의미로 사용됩니다.

Master and Slave(마스터 및 슬레이브) - 둘 이상의 시스템에서 Kerberos 인증 소프트웨어를 실행할 수 있습니다.그러나 항상 Kerberos 데이터베이스의 명확한 복제본은 하나만 있습니다.이 데이터베이스를 저장하는 시스템을 마스터 시스템이나 마스터라고 합니다.다른 시스템은 Kerberos 데이터베이스의 읽기 전용 복사본을 가질 수 있으며 이를 슬레이브라고 합니다.

Kerberos 의 동인

네트워크로 연결되지 않은 개인용 컴퓨팅 환경에서는 개인 컴퓨터를 물리적으로 보호함으로써 리소스와 정보를 보호할 수 있습니다.시간을 절약하는 컴퓨팅 환경에서 운영 체제는 사용자를 서로 보호하고 리소스를 제어합니다.각 사용자가 읽거나 수정할 수 있는 항목을 결정하려면 시간 분할 시스템에서 각 사용자를 식별해야 합니다.이는 사용자가 로그인할 때 수행됩니다.

여러 개별 컴퓨터에서 서비스를 필요로 하는 사용자의 네트워크에서는 액세스 제어를 위해 세 가지 접근 방식이 있습니다.사용자가 로그인한 시스템에 의존하여 무단 액세스를 방지할 수 있습니다.호스트가 자신의 ID를 증명하도록 요청할 수 있지만, 호스트의 말을 신뢰할 수 있습니다.또는 사용자가 필요한 각 서비스에 대해 자신의 신원을 증명해야 할 수 있습니다.

모든 장비가 엄격한 통제 하에 있는 폐쇄적인 환경에서는 첫 번째 접근 방식을 사용할 수 있습니다.조직이 네트워크를 통해 통신하는 모든 호스트를 제어할 경우 이는 합리적인 방법입니다.

좀 더 개방적인 환경에서는 조직 제어 하에 있는 호스트만 선택적으로 신뢰할 수 있습니다.이 경우 각 호스트가 ID를 입증해야 합니다.rlogin 및 rsh 프로그램은 이 방식을 사용합니다.이러한 프로토콜에서는 연결이 설정된 인터넷 주소를 확인하여 인증이 수행됩니다.

Athena 환경에서는 조직의 제어에 속하지 않는 호스트의 요청을 수락할 수 있어야 합니다.사용자는 워크스테이션을 완벽하게 제어할 수 있습니다.이를 재부팅하거나 독립적으로 실행하거나 자체 테이프를 부팅할 수도 있습니다.따라서 세 번째 접근 방식이 필요합니다.사용자는 원하는 각 서비스에 대한 자신의 신원을 증명해야 합니다.서버도 ID를 증명해야 합니다.네트워크 서버를 실행하는 호스트를 물리적으로 보호하는 것은 충분하지 않습니다.네트워크의 다른 사용자가 지정된 서버로 가장할 수 있습니다.

Cisco 환경에서는 식별 메커니즘에 몇 가지 요구 사항이 있습니다.첫째, 안전해야 합니다.잠재적인 공격자가 취약한 링크가 될 인증 메커니즘을 찾지 못할 정도로 이를 회피하기 어려울 수 있습니다.네트워크를 보고 있는 사용자는 다른 사용자를 가장하는 데 필요한 정보를 가져올 수 없습니다.둘째, 믿을 만합니다.많은 서비스에 대한 액세스는 인증 서비스에 따라 달라집니다.신뢰할 수 없다면 서비스 시스템 전체가 그렇지 않을 것입니다.셋째, 투명해야 한다.사용자는 인증이 발생하는 것을 인식하지 않는 것이 좋습니다.마지막으로, 확장 가능해야 합니다.많은 시스템이 Athena 호스트와 통신할 수 있습니다.이러한 모든 기능이 Cisco 메커니즘을 지원하는 것은 아니지만, Software가 Cisco를 지원한다고 해서 고장이 나면 안 됩니다.

Kerberos는 위의 요구 사항을 충족하기 위한 노력의 결과입니다.사용자가 워크스테이션으로 이동하면 로그인됩니다.사용자가 알 수 있는 한, 이 초기 식별은 로그인 세션 동안 필요한 모든 네트워크 서버에 자신의 ID를 입증하기에 충분합니다.Kerberos의 보안은 여러 인증 서버의 보안에 의존하지만 사용자가 로그인하는 시스템이나 사용할 최종 서버의 보안에 의존하지 않습니다.인증 서버는 네트워크 전체에 분산된 서버에 자신의 ID를 증명할 수 있는 적절한 인증 사용자를 제공합니다.

인증은 안전한 네트워크 환경을 위한 기본 구성 요소입니다.예를 들어, 서버가 특정 클라이언트 ID에 대해 알고 있는 경우 서비스를 제공할 것인지, 사용자에게 특별한 권한을 부여해야 하는지 여부, 서비스에 대한 BOM을 받을 사람 등을 결정할 수 있습니다.다시 말해, 권한 부여 및 계정 체계는 Kerberos가 제공하는 인증 위에 구축될 수 있으며, 이로 인해 개인 컴퓨터 또는 시간 분할 시스템과 동일한 보안이 이루어집니다.

Kerberos란?

Kerberos는 Needham 및 Schroeder가 제시하는 모델을 기반으로 하는 신뢰할 수 있는 타사 인증 서비스입니다.각 클라이언트는 서로 다른 각 클라이언트의 ID에 대한 Kerberos의 판단을 정확하게 인식한다는 점에서 신뢰합니다.재생 탐지를 돕기 위해 원본 모델에 타임스탬프(현재 날짜 및 시간을 나타내는 큰 숫자)가 추가되었습니다.메시지가 네트워크에서 도난되고 나중에 다시 전송될 때 재생됩니다.재생 및 기타 인증 문제에 대한 자세한 내용은 Voydock 및 Kent를 참조하십시오.

Kerberos는 무엇을 합니까?

Kerberos는 클라이언트 및 개인 키의 데이터베이스를 유지합니다.개인 키는 Kerberos 및 해당 키가 속한 클라이언트에만 알려진 큰 수입니다.클라이언트가 사용자이면 암호화된 비밀번호입니다.인증이 필요한 네트워크 서비스는 Kerberos에 등록되며, 이러한 서비스를 사용하려는 클라이언트도 마찬가지입니다.개인 키는 등록 시 협상됩니다.

Kerberos는 이러한 개인 키를 알고 있기 때문에 메시지를 생성하여 한 클라이언트에 다른 클라이언트가 실제로 자신이 주장하는 대상임을 확신시킬 수 있습니다.또한 Kerberos는 세션 키라고 하는 임시 개인 키를 생성하며, 이는 두 개의 클라이언트에 제공되지만 다른 클라이언트에는 제공되지 않습니다.세션 키를 사용하여 두 당사자 간의 메시지를 암호화할 수 있습니다.

Kerberos는 세 가지 고유한 보호 수준을 제공합니다.애플리케이션 프로그래머는 애플리케이션의 요구 사항에 따라 적절한 항목을 결정합니다.예를 들어, 일부 애플리케이션은 네트워크 연결 시작 시 신뢰성을 설정하기만 하면 되며, 지정된 네트워크 주소의 추가 메시지가 인증된 당사자로부터

시작된다고 가정할 수 있습니다.Cisco의 인증된 네트워크 파일 시스템은 이러한 수준의 보안을 사용합니다.

다른 애플리케이션에는 각 메시지에 대한 인증이 필요하지만 메시지 내용이 공개되었는지 여부는 중요하지 않습니다.이러한 경우 Kerberos는 안전한 메시지를 제공합니다.그러나 더 높은 수준의 보안은 개인 메시지에서 제공됩니다. 각 메시지는 인증될 뿐만 아니라 암호화됩니다.예를 들어 Kerberos 서버 자체에서 네트워크를 통해 비밀번호를 보내는 데 개인 메시지를 사용합니다.

Kerberos 소프트웨어 구성 요소

Athena 구현은 다음과 같은 여러 모듈로 구성됩니다.

- Kerberos 애플리케이션 라이브러리
- 암호화 라이브러리
- 데이터베이스 라이브러리
- 데이터베이스 관리 프로그램
- 관리 서버
- 인증 서버
- db 전파 소프트웨어
- 사용자 프로그램
- 애플리케이션

Kerberos 애플리케이션 라이브러리는 애플리케이션 클라이언트 및 애플리케이션 서버에 대한 인터페이스를 제공합니다.여기에는 인증 요청을 만들거나 읽는 루틴, 안전 또는 개인 메시지를 만드는 루틴도 포함됩니다.

Kerberos의 암호화는 데이터 암호화 표준인 DES를 기반으로 합니다.암호화 라이브러리는 이러한 루틴을 구현합니다.속도와 보안 사이의 절충과 함께 여러 암호화 방법이 제공됩니다.전파 CBC 모드라고 하는 DES CBC(Cipher Block Chaining) 모드에 대한 확장도 제공됩니다.CBC에서 오류는 현재 암호 블록을 통해서만 전파되지만 PCBC에서는 메시지 전체에 오류가 전파됩니다.이렇게 하면 오류가 발생할 경우 전체 메시지가 일부만 발생하는 것이 아니라 무용지물이 됩니다.암호화 라이브러리는 독립적인 모듈이며 다른 DES 구현 또는 다른 암호화 라이브러리로 교체할 수 있습니다.

또 다른 교체 가능한 모듈은 데이터베이스 관리 시스템입니다.Ingres가 원래 사용되었지만 데이터베이스 라이브러리의 현재 Athena 구현에서는 ndbm을 사용합니다.다른 데이터베이스 관리 라이브러리도 사용할 수 있습니다.

Kerberos 데이터베이스 요구 사항은 간단합니다.사용자 이름, 개인 키, 보안 주체 만료 날짜 및 일부 관리 정보가 포함된 레코드가 각 사용자에 대해 보관됩니다.(만료 날짜는 항목이 더 이상 유효하지 않은 날짜입니다.일반적으로 등록 시 향후 몇 년 안에 완료됩니다.)

실제 이름, 전화 번호 등과 같은 기타 사용자 정보는 다른 서버인 Hesiod 이름 서버에 의해 보관됩니다.이렇게 하면 암호와 같은 민감한 정보를 매우 높은 보안 조치를 사용하여 Kerberos에서 처리할 수 있습니다.Hesiod에서 보관하던 중요하지 않은 정보는 다르게 처리되지만,예를 들어 네트워크를 통해 암호화되지 않은 상태로 전송될 수 있습니다.

Kerberos 서버는 데이터베이스 라이브러리를 사용하며, 데이터베이스 관리를 위한 툴도 사용합니다.

관리 서버(또는 KDBM 서버)는 데이터베이스에 대한 읽기-쓰기 네트워크 인터페이스를 제공합니다.프로그램의 클라이언트 부분은 네트워크의 모든 시스템에서 실행될 수 있습니다.그러나 데이터베

이스를 변경하려면 서버 측에서 Kerberos 데이터베이스를 사용하는 컴퓨터에서 실행해야 합니다.

반면 인증 서버(또는 Kerberos 서버)는 Kerberos 데이터베이스에서 읽기 전용 작업(주체 인증 및 세션 키 생성)을 수행합니다.이 서버는 Kerberos 데이터베이스를 수정하지 않으므로 마스터 Kerberos 데이터베이스의 읽기 전용 복사본을 포함하는 컴퓨터에서 실행될 수 있습니다.

데이터베이스 전파 소프트웨어는 Kerberos 데이터베이스의 복제를 관리합니다.각 시스템에서 실행 중인 인증 서버의 복사본을 사용하여 여러 시스템에 데이터베이스 복사본을 가질 수 있습니다.이러한 각 슬레이브 시스템은 지정된 간격으로 마스터 시스템에서 Kerberos 데이터베이스의 업데이트를 수신합니다.

마지막으로, Kerberos에 로그인하고, Kerberos 비밀번호를 변경하고, Kerberos 티켓을 표시 또는 삭제하는 최종 사용자 프로그램이 있습니다(티켓은 나중에 설명됨).

Kerberos 이름

엔티티의 이름을 지정하는 것도 인증의 일부입니다.인증 프로세스는 클라이언트가 요청에 명명된 클라이언트인지 확인하는 것입니다.이름은 무엇으로 구성됩니까?Kerberos에서 사용자 및 서버 모두 이름이 지정됩니다.인증 서버에 관한한 이는 동일합니다.이름은 기본 이름, 인스턴스 및 영역으로 구성되며 name.instance@realm으로 표시됩니다.

기본 이름은 사용자 또는 서비스의 이름입니다.이 인스턴스는 기본 이름의 변형을 구분하는 데 사용됩니다.사용자의 경우 인스턴스에 "root" 또는 "admin" 인스턴스와 같은 특수 권한이 있을 수 있습니다.Athena 환경의 서비스의 경우 일반적으로 인스턴스는 서버가 실행되는 시스템의 이름입니다.예를 들어, rlogin 서비스는 다른 호스트에 다른 인스턴스를 가지고 있습니다.rlogin.priam은 priam이라는 호스트의 rlogin 서버입니다.Kerberos 티켓은 명명된 단일 서버에만 적합합니다.따라서 동일한 서비스의 다른 인스턴스에 액세스하려면 별도의 티켓이 필요합니다.영역은 인증 데이터를 유지 관리하는 관리 엔티티의 이름입니다.예를 들어, 서로 다른 기관이 각각 다른 데이터베이스를 포함하는 자체 Kerberos 시스템을 가질 수 있습니다.Kerberos 영역은 서로 다릅니다.(영역은 Interaction with Other Kerberos에서 자세히 설명합니다.)

Kerberos 작동 방식

이 섹션에서는 Kerberos 인증 프로토콜에 대해 설명합니다.위에서 언급한 대로 Kerberos 인증 모델은 Needham 및 Schroeder 키 배포 프로토콜을 기반으로 합니다.사용자가 서비스를 요청할 경우 해당 ID를 설정해야 합니다.이렇게 하려면 티켓이 서버에 표시되고, 티켓이 원래 사용자에게 발급되었지만 도난되지 않았다는 증거가 표시됩니다.Kerberos를 통한 인증에는 세 가지 단계가 있습니다.첫 번째 단계에서는 사용자가 다른 서비스에 대한 액세스를 요청하는 데 사용할 자격 증명을 가져옵니다.두 번째 단계에서는 사용자가 특정 서비스에 대한 인증을 요청합니다.최종 단계에서는 최종 서버에 이러한 자격 증명을 표시합니다.

Kerberos 자격 증명

Kerberos 인증 모델에는 두 가지 유형의 자격 증명이 사용됩니다.티켓 및 인증자입니다.둘 다 개인 키 암호화를 기반으로 하지만 다른 키를 사용하여 암호화됩니다.티켓은 인증 서버와 최종 서버 간에 티켓이 발급된 사람의 ID를 안전하게 전달하는 데 사용됩니다.티켓은 또한 표를 사용하는 사람이 그것을 발행한 사람과 동일한지 확인하기 위해 사용할 수 있는 정보를 전달합니다.인증자는 추가 정보를 포함합니다. 이는 티켓의 정보와 비교했을 때 티켓을 제시하는 클라이언트가 티켓이 발급된 클라이언트와 동일함을 증명합니다.

티켓은 단일 서버 및 단일 클라이언트에 적합합니다.서버 이름, 클라이언트 이름, 클라이언트의 인터넷 주소, 타임스탬프, 수명 및 임의의 세션 키가 포함됩니다.이 정보는 티켓을 사용할 서버의 키를 사용하여 암호화됩니다.티켓이 발급되면 지정된 클라이언트가 티켓이 만료될 때까지 명명된 서버에 액세스하기 위해 여러 번 사용할 수 있습니다.티켓은 서버 키에서 암호화되므로 사용자가 티켓을 수정할 필요 없이 티켓을 서버에 전달할 수 있습니다.

티켓과 달리 인증자는 한 번만 사용할 수 있습니다.클라이언트가 서비스를 사용하려는 때마다 새 서비스를 생성해야 합니다.클라이언트가 인증자 자체를 빌드할 수 있으므로 문제가 발생하지 않습니다.인증자는 클라이언트 이름, 워크스테이션의 IP 주소 및 현재 워크스테이션 시간을 포함합니다.인증자는 티켓의 일부인 세션 키로 암호화됩니다.

초기 Kerberos 티켓 가져오기

사용자가 워크스테이션으로 이동하면 자신의 신원을 입증할 수 있는 정보는 한 가지뿐입니다.사용자 비밀번호입니다.인증 서버와의 초기 교환은 비밀번호가 손상될 가능성을 최소화하도록 설계되며, 동시에 사용자가 해당 비밀번호를 알지 못한 채 자신을 올바르게 인증하도록 허용하지 않습니다.로그인 프로세스는 시간 분할 시스템에 로그인하는 것과 동일하게 사용자에게 표시됩니다.하지만 그 이면에는 꽤 다르다.

사용자에게 사용자 이름을 묻는 메시지가 표시됩니다.입력한 후에는 사용자의 이름과 티켓 부여 서비스라고 하는 특수 서비스의 이름을 포함하는 인증 서버로 요청이 전송됩니다.

인증 서버는 클라이언트에 대해 알고 있는지 확인합니다.이 경우 클라이언트와 티켓 부여 서버 간에 나중에 사용할 임의 세션 키를 생성합니다.그런 다음 클라이언트 이름, 티켓 부여 서버 이름, 현재 시간, 티켓 수명, 클라이언트 IP 주소, 방금 만든 임의 세션 키가 들어 있는 티켓 부여 서버에 대한 티켓을 만듭니다.이는 모두 티켓 부여 서버 및 인증 서버에만 알려진 키로 암호화됩니다.

그런 다음 인증 서버는 무작위 세션 키의 복사본과 일부 추가 정보와 함께 티켓을 클라이언트로 다시 전송합니다.이 응답은 클라이언트의 개인 키로 암호화되며, Kerberos 및 클라이언트에만 알려지며 사용자의 비밀번호로 파생됩니다.

클라이언트가 응답을 수신하면 사용자에게 비밀번호를 묻는 메시지가 표시됩니다.비밀번호는 DES 키로 변환되며 인증 서버의 응답을 해독하는 데 사용됩니다.티켓 및 세션 키, 기타 일부 정보는 나중에 사용할 수 있도록 저장되며 사용자의 암호와 DES 키가 메모리에서 지워집니다.

교환이 완료되면 워크스테이션은 티켓 부여 티켓 수명 동안 사용자의 신원을 확인하는 데 사용할 수 있는 정보를 보유합니다.워크스테이션의 소프트웨어가 이전에 변조되지 않은 경우 티켓의 수명 이후에도 다른 사용자가 사용자를 가장할 수 있는 정보가 없습니다.

Kerberos 서비스 요청

잠시 동안 사용자가 원하는 서버에 대한 티켓을 이미 가지고 있는 것으로 가정하겠습니다.서버에 액세스하기 위해 응용 프로그램은 클라이언트의 이름과 IP 주소 및 현재 시간을 포함하는 인증자를 구축합니다.그런 다음 인증자는 서버 티켓과 함께 수신된 세션 키로 암호화됩니다.그러면 클라이언트는 개별 애플리케이션에 의해 정의된 방식으로 인증자와 함께 티켓을 서버로 전송합니다.

서버에서 인증자 및 티켓을 수신하면 서버는 티켓을 해독하고, 티켓에 포함된 세션 키를 사용하여 인증자를 해독하고, 티켓의 정보와 인증자의 정보를 비교하고, 요청을 받은 IP 주소, 현재 시간을 비교합니다.모든 항목이 일치하면 요청을 계속 진행할 수 있습니다.

시계는 몇 분 내에 와 동기화된 것으로 간주된다.요청의 시간이 너무 미래나 과거일 경우 서버는 요청을 이전 요청을 재생하려는 시도로 간주합니다.또한 서버는 유효한 타임스탬프가 있는 모든 과거

요청을 추적할 수 있습니다.다시 재생 공격을 돋우기 위해 이미 수신한 것과 동일한 티켓 및 타임스탬프로 받은 요청은 폐기될 수 있습니다.

마지막으로, 클라이언트가 서버의 ID를 증명하도록 지정한 경우 서버는 인증자에게 전송된 클라이언트의 타임스탬프에 하나를 추가하고 세션 키의 결과를 암호화하고 결과를 클라이언트로 다시 보냅니다.

이 Exchange가 끝날 때 서버는 Kerberos에 따르면 클라이언트가 Kerberos라고 하는 것이 확실합니다.상호 인증이 발생할 경우 클라이언트는 서버가 인증되었음을 확인합니다.또한 클라이언트와 서버는 다른 사람이 모르는 키를 공유하며, 해당 키로 암호화된 합리적으로 최근 메시지가 다른 사람과 함께 시작되었다고 간주할 수 있습니다.

Kerberos 서버 티켓 가져오기

티켓은 단일 서버에만 적합합니다.따라서 클라이언트가 사용하려는 각 서비스에 대해 별도의 티켓을 얻어야 합니다.개별 서버 티켓은 티켓 부여 서비스에서 구할 수 있습니다.Ticket-Granting Service 자체는 서비스이므로 이전 섹션에서 설명한 서비스 액세스 프로토콜을 사용합니다.

아직 요청되지 않은 티켓이 필요한 프로그램은 티켓 부여 서버로 요청을 보냅니다.요청에는 이전 섹션에 설명된 대로 작성된 티켓 부여 티켓 및 인증자와 함께 티켓이 요청된 서버의 이름이 포함됩니다.

그러면 티켓 부여 서버는 위에 설명된 대로 인증자 및 티켓 부여 티켓을 확인합니다.유효한 경우, 티켓 부여 서버는 클라이언트와 새 서버 간에 사용할 새 임의 세션 키를 생성합니다.그런 다음 클라이언트의 이름, 서버 이름, 현재 시간, 클라이언트의 IP 주소 및 방금 생성한 새 세션 키가 포함된 새 서버에 대한 티켓을 작성합니다.새 티켓의 수명은 티켓 부여 티켓의 남은 수명 및 서비스의 기본값입니다.

그러면 티켓 부여 서버는 세션 키 및 기타 정보와 함께 티켓을 클라이언트로 다시 보냅니다.그러나 이번에는 Ticket-Granting Ticket의 일부인 세션 키에서 회신이 암호화됩니다.이렇게 하면 사용자가 비밀번호를 다시 입력할 필요가 없습니다.

Kerberos 데이터베이스

지금까지 Kerberos 데이터베이스에 대한 읽기 전용 액세스가 필요한 작업에 대해 설명했습니다.이러한 작업은 마스터 및 슬레이브 시스템에서 모두 실행할 수 있는 인증 서비스에 의해 수행됩니다.

이 섹션에서는 데이터베이스에 대한 쓰기 액세스가 필요한 작업에 대해 설명합니다.이러한 작업은 KDBM(Kerberos Database Management Service)이라는 관리 서비스에서 수행합니다. 현재 구현에서는 마스터 Kerberos 데이터베이스만 변경할 수 있다고 명시하고 있습니다.슬레이브 복사본은 읽기 전용입니다.따라서 KDBM 서버는 마스터 Kerberos 시스템에서만 실행될 수 있습니다.

인증(슬레이브에서는)이 계속 발생할 수 있지만 마스터 시스템이 중단된 경우 관리 요청을 처리할 수 없습니다.관리 요청이 드물게 발생하므로, 당사의 경험에서 이것은 문제를 나타내지 않았습니다.

KDBM은 사용자의 비밀번호 변경 요청을 처리합니다.네트워크를 통해 KDBM에 요청을 보내는 이 프로그램의 클라이언트 측은 kpasswd 프로그램입니다.또한 KDBM은 데이터베이스에 주도자를 추가할 수 있는 Kerberos 관리자의 요청과 기존 주도자의 비밀번호를 변경할 수 있습니다.네트워크를 통해 KDBM에 요청을 전송하는 관리 프로그램의 클라이언트 측면은 kadmin 프로그램입니다.

KDBM 서버

KDBM 서버는 데이터베이스에 주도자를 추가하거나 기존 주도자의 비밀번호를 변경하라는 요청을 수락합니다.이 서비스는 티켓 부여 서비스에서 티켓을 발행하지 않는다는 점에서 고유합니다.대신 인증 서비스 자체를 사용해야 합니다(티켓 부여 티켓을 얻는 데 사용되는 것과 동일한 서비스). 이는 사용자가 비밀번호를 입력해야 하는 것을 목적으로 합니다.그렇지 않은 경우 사용자가 워크스테이션을 무인 상태로 방치하면, 행인이 걸어 올라가 암호를 변경할 수 있습니다. 이는 방지되어야 합니다.마찬가지로 관리자가 자신의 워크스테이션을 보호하지 않은 상태로 방치한 경우, 행인이 시스템의 비밀번호를 변경할 수 있습니다.

KDBM 서버가 요청을 받으면 변경 요청자의 인증된 주체 이름과 요청 대상의 주체 이름을 비교하여 권한을 부여합니다.동일한 경우 요청이 허용됩니다.KDBM 서버가 동일하지 않으면 KDBM 서버는 액세스 제어 목록(마스터 Kerberos 시스템의 파일에 저장)을 협의합니다. 요청자의 주체 이름이 이 파일에 있는 경우 요청이 허용되며, 그렇지 않으면 요청이 거부됩니다.

일반적으로 NULL 인스턴스(기본 인스턴스)가 있는 이름은 액세스 제어 목록 파일에 나타나지 않습니다.대신 admin 인스턴스가 사용됩니다.따라서 사용자가 Kerberos의 관리자가 되려면 해당 사용자 이름에 대한 관리자 인스턴스를 생성하고 액세스 제어 목록에 추가해야 합니다.이 규칙을 사용하면 관리자가 Kerberos 관리에 다른 비밀번호를 사용할 수 있으며, 그런 다음 일반 로그인에 사용할 수 있습니다.

KDBM 프로그램에 대한 모든 요청(허용 또는 거부)이 기록됩니다.

kadmin 및 kpasswd 프로그램

Kerberos 관리자는 kadmin 프로그램을 사용하여 데이터베이스에 주도자를 추가하거나 기존 주도자의 비밀번호를 변경합니다.관리자가 kadmin 프로그램을 호출할 때 관리자 인스턴스 이름의 비밀번호를 입력해야 합니다.이 비밀번호는 KDBM 서버의 티켓을 가져오는 데 사용됩니다.

사용자는 kpasswd 프로그램을 사용하여 Kerberos 비밀번호를 변경할 수 있습니다.프로그램을 호출할 때 이전 비밀번호를 입력해야 합니다.이 비밀번호는 KDBM 서버의 티켓을 가져오는 데 사용됩니다.

Kerberos 데이터베이스 복제

각 Kerberos 영역에는 인증 데이터베이스의 마스터 복사본을 포함하는 마스터 Kerberos 시스템이 있습니다.시스템의 다른 슬레이브 시스템에 데이터베이스의 추가 읽기 전용 복사본을 가질 수 있습니다.데이터베이스 복제본을 여러 개 만들 경우 일반적으로 다음과 같은 이점을 얻을 수 있습니다.더 우수한 가용성과 성능을 제공합니다.마스터 시스템이 다운된 경우 슬레이브 시스템 중 하나에서 인증을 계속 수행할 수 있습니다.여러 시스템 중 하나에서 인증을 수행하는 기능을 사용하면 마스터 시스템에서 병목 현상이 발생할 가능성이 줄어듭니다.

데이터베이스의 여러 복제본을 유지하면 데이터 일관성 문제가 발생합니다.우리는 아주 간단한 방법으로도 불일치를 처리할 수 있다는 것을 발견했다.마스터 데이터베이스는 매시간마다 덤프됩니다.데이터베이스가 전체적으로 슬레이브 시스템으로 전송되고, 그러면 데이터베이스가 자체 데이터베이스를 업데이트합니다.마스터 호스트의 프로그램(kprop)은 각 슬레이브 시스템에서 실행되는 kpropd라는 피어 프로그램으로 업데이트를 전송합니다.첫 번째 kprop은 보내려는 새 데이터베이스의 체크섬을 보냅니다.체크섬은 마스터 및 슬레이브 Kerberos 시스템이 모두 소유하는 Kerberos 마스터 데이터베이스 키로 암호화됩니다.그런 다음 데이터는 네트워크를 통해 슬레이브 시스템의 kpropd로 전송됩니다.슬레이브 전달 서버는 수신한 데이터의 체크섬을 계산하고, 마스터에서 전송한 체크섬과 일치하면 새 정보를 사용하여 슬레이브의 데이터베이스를 업데이트합니다.

Kerberos 데이터베이스의 모든 비밀번호는 마스터 데이터베이스 키로 암호화됩니다. 따라서 네트워크를 통해 마스터에서 슬레이브로 전달된 정보는 엿듣는 사람에게 유용하지 않습니다.그러나 마스터 호스트의 정보만 슬레이브에서 수락하고 데이터의 변조를 탐지하여 체크섬을 탐지해야 합니다.

외부의 Kerberos

이 섹션에서는 Kerberos를 실제 관점에서, 사용자가 먼저 확인한 다음 애플리케이션 프로그래머의 관점에서, 마지막으로 Kerberos 관리자의 작업을 통해 설명합니다.

Kerberos 사용자의 눈 모양

모든 작업이 제대로 진행되면 Kerberos 가 있는 것을 거의 알 수 없습니다.UNIX 구현에서 티켓 부여 티켓은 로그인 프로세스의 일부로 Kerberos에서 가져옵니다.사용자의 Kerberos 비밀번호 변경은 passwd 프로그램의 일부입니다.또한 사용자가 로그아웃하면 Kerberos 티켓이 자동으로 제거됩니다.

사용자의 로그인 세션이 티켓 부여 티켓의 수명(현재 8시간)보다 오래 지속되는 경우, 다음에 Kerberos 인증 애플리케이션이 실행될 때 오류가 발생하므로 사용자는 Kerberos가 있음을 알 수 있습니다.해당 Kerberos 티켓이 만료되었습니다.그 시점에서 사용자는 Kinit 프로그램을 실행하여 Ticket-Granting Server에 대한 새 티켓을 얻을 수 있습니다.로그인할 때와 마찬가지로 비밀번호를 입력해야 비밀번호를 얻을 수 있습니다.호기심에서 klist 명령을 실행하는 사용자는 Kerberos 인증이 필요한 서비스에 대해 자신을 대신하여 자동으로 얻은 모든 티켓에 놀랄수도 있습니다.

프로그래머 관점의 Kerberos

Kerberos 애플리케이션을 쓰는 프로그래머는 클라이언트와 서버 측으로 구성된 기존 네트워크 애플리케이션에 인증을 추가하는 경우가 많습니다.이 프로세스를 "Kerberos" 프로그램이라고 합니다.Kerberos는 일반적으로 초기 서비스 요청 시 인증을 수행하기 위해 Kerberos 라이브러리에 대한 호출을 수행합니다.또한 애플리케이션 클라이언트와 애플리케이션 서버 간에 전송되는 메시지와 데이터를 암호화하기 위해 DES 라이브러리에 대한 호출이 필요할 수 있습니다.

가장 일반적으로 사용되는 라이브러리 함수는 클라이언트 쪽에 있는 krb_mk_req와 서버 쪽에 있는 krb_rd_req입니다.krb_mk_req 루틴은 요청된 대상 서버의 이름, 인스턴스 및 영역, 전송할 데이터의 체크섬 등의 매개 변수로 사용됩니다.그런 다음 클라이언트는 네트워크를 통해 krb_mk_req 호출이 반환한 메시지를 애플리케이션의 서버 측에 전송합니다.서버가 이 메시지를 받으면 라이브러리 루틴 krb_rd_req를 호출합니다.그 일상은 발신자의 협의를 받은 신원에 대한 진위성에 대한 판단을 반환 한다.

응용 프로그램에서 클라이언트와 서버 간에 전송되는 메시지를 암호화할 것을 요구할 경우 krb_mk_priv(krb_rd_priv)로 라이브러리 호출을 수행하여 양쪽이 현재 공유하는 세션 키의 메시지를 암호화(암호 해독)할 수 있습니다.

Kerberos 관리자의 작업

Kerberos 관리자의 작업은 데이터베이스를 초기화하는 프로그램을 실행하는 것으로 시작합니다.다른 프로그램을 실행하여 데이터베이스에 필수 주도자(예: Kerberos 관리자 이름)를 관리자 인스턴스에 등록해야 합니다.Kerberos 인증 서버와 관리 서버를 시작해야 합니다.슬레이브 데이터베이스가 있는 경우 관리자는 마스터에서 슬레이브로 데이터베이스 업데이트를 전파하는 프로그램을 주기적으로 해제하도록 정렬해야 합니다.

이러한 초기 단계를 수행한 후 관리자는 kadmin 프로그램을 사용하여 네트워크를 통해 데이터베이스를 조작합니다.이 프로그램을 통해 새 주도자를 추가하고 비밀번호를 변경할 수 있습니다.

특히 새 Kerberos 애플리케이션이 시스템에 추가되면 Kerberos 관리자는 몇 가지 단계를 수행하여 작업을 수행해야 합니다.서버는 데이터베이스에 등록되어 개인 키를 할당해야 합니다(일반적으로 자동으로 생성된 임의 키임). 그런 다음 일부 데이터(서버 키 포함)를 데이터베이스에서 추출하여 서버 시스템의 파일에 설치해야 합니다.기본 파일은 /etc/srvtab입니다.서버에서 호출한 krb_rd_req 라이브러리 루틴(이전 섹션 참조)은 해당 파일의 정보를 사용하여 서버의 개인 키로 암호화된 메시지를 해독합니다./etc/srvtab 파일은 터미널에 입력된 비밀번호로 서버를 인증하여 사용자를 인증합니다.

또한 Kerberos 관리자는 Kerberos 시스템이 물리적으로 안전하며 마스터 데이터베이스의 백업을 유지 관리하는 것이 좋습니다.

더 큰 Kerberos 그림

이 섹션에서는 다른 네트워크 서비스 및 애플리케이션에서 Kerberos를 사용하는 방법, 원격 Kerberos 영역과 상호 작용하는 방법 등 Athena 환경에 Kerberos가 어떻게 적용되는지 설명합니다 .아테나 환경에 대한 자세한 설명은 G.W. Treese를 참조하십시오.

다른 네트워크 서비스의 Kerberos 사용

여러 네트워크 애플리케이션이 Kerberos를 사용하도록 수정되었습니다.rlogin 및 rsh 명령은 먼저 Kerberos를 사용하여 인증하려고 시도합니다.유효한 Kerberos 티켓을 가진 사용자는 .rhosts 파일을 설정하지 않고도 다른 Athena 시스템에 로그인할 수 있습니다.Kerberos 인증이 실패하면 프로그램은 일반적인 권한 부여 방법(이 경우 .rhosts 파일)으로 돌아갑니다.

"우체국"에서 전자 메일을 검색하려는 사용자를 인증하기 위해 Kerberos를 사용하도록 우체국 프로토콜을 수정했습니다. Zephyr라는 메시지 전달 프로그램은 최근 Athena에서 개발되었으며 인증에도 Kerberos를 사용합니다.

레지스터라는 새 사용자를 등록하는 프로그램은 SMS(Service Management System)와 Kerberos를 모두 사용합니다.SMS에서 이름 및 MIT 식별 번호와 같은 새로운 Athena 사용자가 입력한 정보가 유효한지 여부를 결정합니다.그런 다음 Kerberos로 확인하여 요청된 사용자 이름이 고유한지 확인합니다.모두 정상적으로 작동하면 사용자 이름과 비밀번호를 포함하는 Kerberos 데이터베이스에 새 항목이 생성됩니다.

Kerberos를 사용하여 Sun의 네트워크 파일 시스템을 보호하는 방법에 대한 자세한 내용은 부록을 참조하십시오.

다른 Kerberos와의 상호 작용

서로 다른 관리 조직에서 사용자 인증에 Kerberos를 사용할 것으로 예상됩니다.또한 많은 경우 한 조직의 사용자가 다른 조직의 서비스를 사용하기를 원할 것입니다.Kerberos는 여러 관리 도메인을 지원합니다.Kerberos의 이름 사양은 영역이라는 필드를 포함합니다.이 필드는 사용자를 인증할 관리 도메인의 이름을 포함합니다.

서비스는 일반적으로 단일 영역에 등록되며 해당 영역에 대해 인증 서버에서 발급한 자격 증명만 수락합니다.일반적으로 사용자는 단일 영역(로컬 영역)에 등록되지만, 로컬 영역에서 제공하는 인증 강도에 따라 다른 영역(원격 영역)에서 발급한 인증서를 얻을 수 있습니다.원격 영역에서 유효한 자격 증명은 사용자가 원래 인증된 영역을 나타냅니다.원격 영역의 서비스는 필요한 보안 수준 및

사용자를 처음 인증한 영역의 신뢰 수준에 따라 해당 자격 증명을 적용할지 여부를 선택할 수 있습니다.

교차 영역 인증을 수행하려면 각 영역 쌍의 관리자가 영역 간에 공유할 키를 선택해야 합니다.로컬 영역의 사용자는 원격 영역의 티켓 부여 서버에 대해 로컬 인증 서버에서 티켓 부여 티켓을 요청할 수 있습니다.이 티켓을 사용하면 원격 티켓 부여 서버는 요청이 자체 영역에서 온 것이 아님을 인식하며 이전에 교환한 키를 사용하여 티켓 부여 티켓을 해독합니다.그런 다음 클라이언트의 영역 필드에 클라이언트가 원래 인증된 영역의 이름이 포함되어 있다는 점을 제외하고 평소와 같이 티켓을 발행합니다.

이러한 접근 방식은 원하는 서비스를 통해 영역에 도달할 때까지 일련의 영역을 통해 자신을 인증할 수 있도록 확장할 수 있습니다.그러나 이렇게 하려면 사용자가 인증된 초기 영역의 이름뿐만 아니라 가져온 전체 경로를 기록해야 합니다.이러한 상황에서, A는 B가 C가 사용자가 너무-소라고 말한다고 말한다고 합니다.이 문은 경로에 있는 모든 사람이 신뢰받는 경우에만 신뢰할 수 있습니다.

Kerberos 문제 및 열린 문제

Kerberos 인증 메커니즘과 관련된 여러 문제 및 열린 문제가 있습니다.문제 중에는 티켓의 올바른 수명을 결정하는 방법, 프록시를 허용하는 방법 및 워크스테이션 무결성을 보장하는 방법이 있습니다.

티켓 수명의 문제는 보안과 편의성 사이에서 적절한 거래를 선택하는 문제이다.티켓의 수명이 길면 티켓과 관련 세션 키를 도난당하거나 잘못 배치하면 더 오랜 기간 사용할 수 있습니다.사용자가 공용 워크스테이션에서 로그아웃하는 것을 잊은 경우 이러한 정보를 도난당할 수 있습니다.또는 사용자가 여러 사용자를 허용하는 시스템에서 인증된 경우 루트에 액세스할 수 있는 다른 사용자가 도난된 티켓을 사용하는 데 필요한 정보를 찾을 수 있습니다.그러나 티켓 수명이 짧을 경우, 만료 되면 사용자는 비밀번호를 다시 입력해야 하는 새 티켓을 받아야 한다는 문제가 있습니다.

열린 문제는 프록시 문제입니다.인증된 사용자가 서버를 대신하여 다른 네트워크 서비스를 인수하도록 허용하려면 어떻게 해야 합니까?중요한 예로는 파일 서버에서 보호된 파일에 직접 액세스할 수 있는 서비스를 사용하는 것입니다.이 문제의 또 다른 예는 인증 전달이라고 부르는 것입니다.사용자가 워크스테이션에 로그인하고 원격 호스트에 로그인하는 경우 원격 호스트에서 프로그램을 실행하는 동안 사용자가 로컬로 사용할 수 있는 동일한 서비스에 액세스할 수 있으면 좋습니다.이 문제를 어렵게 만드는 것은 사용자가 원격 호스트를 신뢰하지 않을 수 있으므로 모든 경우 인증 전달이 바람직하지 않습니다.현재 이 문제에 대한 해결책이 없습니다.

Athena 환경에서 중요한 또 다른 문제는 워크스테이션에서 실행되는 소프트웨어의 무결성을 보장하는 방법입니다.개인 워크스테이션의 경우에는 사용 중인 사용자가 이를 제어할 수 있으므로 이 문제는 그다지 큰 문제가 아닙니다.그러나 공용 워크스테이션에서 다른 사용자가 사용자의 암호를 저장하기 위해 로그인 프로그램을 수정했을 수 있습니다.현재 우리 환경에서 사용 가능한 유일한 솔루션은 사람들이 공용 워크스테이션에서 실행되는 소프트웨어를 수정하는 것을 어렵게 만드는 것입니다.더 나은 솔루션을 위해서는 사용자의 키가 신뢰할 수 있는 시스템을 벗어나지 않아야 합니다.이러한 작업을 수행할 수 있는 한 가지 방법은 사용자가 인증 프로토콜에 필요한 암호화를 수행할 수 있는 스마트카드를 보유하고 있다면 됩니다.

Kerberos 상태

Kerberos의 프로토타입 버전이 1986년 9월에 생산되었습니다.1987년 1월 이후 Kerberos는 사용자 5,000명, 워크스테이션 650개, 서버 65개를 인증하는 유일한 수단이었습니다.또한 Kerberos는 현재 일부 Athena의 시간 분할 시스템에서 액세스를 제어하기 위해 .rhosts 파일 대신 사용되고 있습니다.

Kerberos 승인

처음에 Kerberos는 Steve Miller와 Clifford Neuman이 Jeff Schiller 및 Jerry Saltzer의 제안으로 설계했습니다.그 이후로, 많은 다른 사람들이 이 프로젝트에 참여했습니다.그 중에는 짐 아스트네스, 밥 볼드윈, 존 바바, 리처드 바쉬, 짐 블룸, 빌 브라이언트, 마크 콜란, 롭 프렌치, 댄 게어, 존 쿠비아토비치, 존 맥키, 밥 맥키, 브라이언 머피, 존 오스틀런드 켄 라번, 존 로흐리스, 마이크 소머필드, 빌 머렛, Win Treese와 Stan Zanarotti입니다.

우리는 Dan Geer, Kathy Lieben, Josh Lubarr, Ken Raeburn, Jerry Saltzer, Ed Steiner, Robert van Renesse, 그리고 Win Treese에게 감사하며 이 보고서의 초기 초안을 훨씬 더 많이 개선했습니다.

제드린스키, J.T. 콜, W.E. 소머펠트, "제피르 알림 시스템", Usenix 회의 절차 (1988년 겨울).

M.A. Rosenstein, D.E. Geer, P.J. Levine, Usenix Conference Prospections (1988년 겨울).

R. Sandberg, D. Goldberg, S. Kle만, D. Walsh, B. Lyon, "Design and Implementation of the Sun Network Filesystem", Usenix Conference Procedures (1985년 여름).

부록:SUN의 NFS(Network File System)에 Kerberos 애플리케이션

Project Athena 워크스테이션 시스템의 주요 구성 요소는 사용자의 워크스테이션과 개인 파일 스토리지(홈 디렉토리) 간의 네트워크 상호 작용입니다. 모든 개인 스토리지는 이 용도로 사용되는 컴퓨터 집합(현재 VAX 11/750)에 상주합니다.이를 통해 공개적으로 사용 가능한 UNIX 워크스테이션에서 서비스를 제공할 수 있습니다.사용자가 공개적으로 사용 가능한 이러한 워크스테이션 중 하나에 로그인한 다음 로컬 상주 비밀번호 파일에 대해 이름과 비밀번호를 검증하면 Kerberos를 사용하여 해당 워크스테이션의 신뢰성을 확인합니다.로그인 프로그램은 사용자 이름(UNIX 시스템의 경우)을 묻는 메시지를 표시합니다. 이 사용자 이름은 Kerberos 티켓 부여 티켓을 가져오는 데 사용됩니다.로그인 프로그램은 비밀번호를 사용하여 티켓 해독을 위한 DES 키를 생성합니다.암호 해독에 성공하면 사용자의 홈 디렉토리는 Hesiod 이름 지정 서비스를 참조하여 NFS를 통해 마운트됩니다.그런 다음 로그인 프로그램이 사용자의 셸로 제어권을 전환하며, 홈 디렉토리가 워크스테이션에 "연결됨"이 되었으므로 기존의 사용자별 사용자 지정 파일을 실행할 수 있습니다.Hesiod 서비스는 로컬 암호 파일에 항목을 구성하는 데에도 사용됩니다.(이는 /etc/passwd에서 정보를 조회하는 프로그램의 혜택을 위한 것입니다.)

원격 파일 서비스 제공을 위한 여러 옵션 중에서 Sun의 Network File System을 선택했습니다.그러나 이 시스템은 우리의 요구사항에 꼭 맞지 않는다.NFS는 모든 워크스테이션이 파일 서버의 관점에서 볼 때 두 가지 범주로 분류된다고 가정합니다.신뢰할 수 있습니다.신뢰할 수 없는 시스템에서는 파일에 전혀 액세스할 수 없으며 신뢰할 수 있습니다.신뢰할 수 있는 시스템은 완전히 신뢰할 수 있습니다.신뢰할 수 있는 시스템은 친숙한 관리로 관리된다고 가정합니다.특히 신뢰할 수 있는 워크스테이션에서 파일 서비스 시스템의 유효한 사용자로 가장하여 시스템의 모든 파일에 액세스할 수 있습니다.("root"가 소유한 파일만 면제됩니다.)

Cisco 환경에서는 워크스테이션의 관리(기존의 UNIX 시스템 관리 방식)가 현재 사용 중인 사용자에게 제공됩니다.워크스테이션의 루트 비밀번호는 전혀 숨기지 않습니다. 사용자가 시스템과 동일한 물리적 위치에 있고 모든 콘솔 기능에 액세스할 수 있다는 사실을 매우 잘 알고 있기 때문입니다.따라서 신뢰의 NFS 해석에서 워크스테이션을 신뢰할 수 없습니다.환경에 적절한 액세스 제어를 허용하려면 기본 NFS 소프트웨어를 일부 수정하고 Kerberos를 스키마에 통합해야 했습니다.

Kerberos 수정되지 않은 NFS

(위스콘신 대학교에서) 시작한 NFS의 구현에서 각 NFS 요청에 포함된 데이터의 형태로 인증이 제공되었습니다(NFS 용어에서 "자격 증명"이라고 함). 이 자격 증명에는 요청자의 UID(고유 사용자 식별자)에 대한 정보와 요청자 구성원 자격의 GID(그룹 식별자) 목록이 포함됩니다.그런 다음 NFS 서버에서 액세스 확인을 위해 이 정보를 사용합니다.신뢰할 수 있는 워크스테이션과 신뢰할 수 없는 워크스테이션의 차이점은 NFS 서버에서 해당 자격 증명을 수락하는지 여부입니다.

Kerberos 수정 NFS

Cisco 환경에서 NFS 서버는 자격 증명이 워크스테이션 사용자의 UID를 나타내고 다른 어떤 것도 나타내지 않는 경우에만 워크스테이션의 자격 증명을 수락해야 합니다.

한 가지 분명한 해결책은 자격 증명의 특성을 단순한 UID 및 GID 표시기에서 완전한 Kerberos 인증 데이터로 변경하는 것입니다.그러나 이 솔루션이 채택되면 상당한 성능 페널티가 지불됩니다.모든 디스크 읽기 및 쓰기 작업을 포함하여 모든 NFS 작업에서 자격 증명이 교환됩니다.각 디스크 트랜잭션에 Kerberos 인증을 포함하면 트랜잭션당 전체 암호화(소프트웨어로 수행)가 상당히 많이 추가되며, Envelope 계산에 따르면 허용되지 않는 성능을 제공할 수 있습니다.또한 커널 주소 공간에 Kerberos 라이브러리 루틴을 배치해야 합니다.

아래에 설명된 하이브리드 접근 방식이 필요합니다.기본 개념은 클라이언트 워크스테이션에서 수신한 NFS 서버 맵 자격 증명을 서버 시스템의 유효한(또는 다를 수 있음) 자격 증명으로 가져오는 것입니다.이 매핑은 각 NFS 트랜잭션에서 서버의 커널에서 수행되며 유효한 커널 자격 증명 매핑을 설정하기 전에 Kerberos 조정 인증을 수행하는 사용자 수준 프로세스에 의해 "마운트" 시간에 설정됩니다.

이를 구현하기 위해 커널에 새 시스템 호출을 추가했습니다. 커널은 클라이언트 워크스테이션에서 서버에 사용할 수 있는 자격 증명으로 들어오는 자격 증명을 매핑하는 매핑 기능을 제어할 수 있도록 합니다(클라이언트 시스템이 아닌 서버 시스템에서만 필요). 기본 매핑 기능은 튜플을 매핑합니다.

`<CLIENT-IP-ADDRESS, UID-ON-CLIENT>`
서버 시스템의 유효한 NFS 자격 증명으로 연결합니다.CLIENT-IP-ADDRESS는 클라이언트 시스템에서 제공하는 NFS 요청 패킷에서 추출됩니다.참고:UID-ON-CLIENT를 제외한 클라이언트 생성 자격 증명의 모든 정보가 삭제됩니다.

매핑이 없는 경우 서버는 구성된 방식에 따라 두 가지 방법 중 하나로 응답합니다.Cisco의 친숙한 구성에서는 권한 있는 액세스 권한이 없고 고유한 UID를 가진 사용자 "nobody"에 대한 자격 증명에 대한 매핑 불가능한 요청을 기본값으로 설정합니다.수신 NFS 자격 증명에 대해 유효한 매핑을 찾을 수 없는 경우 불편 서버에서 NFS 액세스 오류를 반환합니다.

새로운 시스템 호출은 커널 상주 맵에서 항목을 추가하고 삭제하는 데 사용됩니다.또한 서버 시스템의 특정 UID에 매핑되는 모든 항목을 플러시하거나 지정된 CLIENT-IP-ADDRESS에서 모든 항목을 플러시할 수 있습니다.

새 트랜잭션 유형인 Kerberos 인증 매핑 요청을 허용하도록 마운트 데몬(서버 시스템의 NFS 마운트 요청을 처리)을 수정했습니다.기본적으로 마운트 프로세스의 일부로, 클라이언트 시스템은 워크스테이션에서 Kerberos 인증자와 해당 UID-ON-CLIENT(Kerberos 인증자에서 암호화됨)의 표시를 제공합니다.서버의 mount 데몬은 Kerberos 주체 이름을 로컬 사용자 이름으로 변환합니다.그러면 이 사용자 이름은 특수 파일에서 조회되어 사용자의 UID 및 GID 목록을 생성합니다.효율성을 위해 이 파일은 사용자 이름을 키로 하는 ndbm 데이터베이스 파일입니다.이 정보에서 NFS 자격 증명이 생성되어 이 요청에 대한 <CLIENT-IP-ADDRESS, CLIENT-UID> 튜플의 유효한 매핑으로 커널에 전달됩니다.

마운트 해제 시 요청이 마운트 데몬에 전송되어 커널에서 이전에 추가한 매핑을 제거합니다.또한 로그아웃 시 요청을 보내 해당 서버의 현재 사용자에 대한 모든 매핑을 무효화함으로써 다음 사용자가 워크스테이션을 사용할 수 있게 되기 전에 남아 있는 모든 매핑을 정리합니다(그렇지 않아도 됨).

수정된 NFS의 Kerberos 보안 영향

이 구현은 완전히 안전하지 않습니다.우선, 사용자 데이터는 여전히 암호화되지 않은 형태로 네트워크를 통해 전송되므로 가로채기가 가능합니다.낮은 수준의 트랜잭션당 인증은 요청 패킷에서 암호화되지 않은 상태로 제공된 <CLIENT-IP-ADDRESS, CLIENT-UID> 쌍을 기반으로 합니다.이 정보는 위조될 수 있으므로 보안이 침해될 수 있습니다.그러나 사용자가 자신의 파일(로그인하는 동안)을 적극적으로 사용하는 경우에만 유효한 매핑이 적용되므로 이 공격 유형은 해당 사용자가 로그인한 경우에만 제한됩니다.사용자가 로그인하지 않은 경우 IP 주소 위조에서는 파일에 대한 무단 액세스를 허용하지 않습니다.

Kerberos 참조

1. S.P. 밀러, B.C. 뉴먼, J.I. 쉴러, 그리고 J.H. 살처, 섹션 E.2.1:Kerberos Authentication and Authorization System, M.I.T. Project Athena, Cambridge, Massachusetts (1987년 12월 21일).
2. E. Balkovich, S.R. Lerman, R.P. Parmelee, "고등 교육 기관에서 컴퓨팅:The Athena Experience," Communications of the ACM, Vol. 28 (11), pp. 1214-1224, ACM (1985년 11월).
3. R.M. Needham 및 M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", ACM의 통신, Vol. 21(12), pp 993-999(1978년 12월).
4. V.L. Voydock and S.T. Kent, "Security Mechanism in High-Level Network Protocols," 컴퓨팅 설문조사, 15(2), ACM(1983년 6월).
5. National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standards Publication 46, Government Printing Office, Washington, DC(1977).
6. SP Dyer, "Hesiod", Usenix Conference Prospections (Winter, 1988).
7. W.J. Bryant, Kerberos Programmer's Tutorial, MIT Project Athena (준비 중).
8. W.J. Bryant, Kerberos 관리자 설명서, MIT Project Athena (준비 중).
9. G.W. Treese, "Berkeley Unix on 1000 Workstation:Athena Changes to 4.3BSD," in Usenix Conference Procedures (Winter, 1988).
10. C.A. DellaFera, M.W. Eichin, R.S. 프랑스, D.C. 제드린스키, J.T. 콜, W.E. Sommerfeld, "The Zephyr Notification System," Usenix Conference Procedures (1988년 겨울).
11. M.A. Rosenstein, D.E. Geer, P.J. Levine, Usenix Conference Prospections (1988년 겨울).
12. R. Sandberg, D. Goldberg, S. Kle만, D. Walsh, B. Lyon, "Design and Implementation of the Sun Network Filesystem", Usenix Conference Procedures (1985년 여름).

관련 정보

- Kerberos 지원 페이지
- 기술 지원 및 문서 – Cisco Systems