Kerberos im Überblick - Ein Authentifizierungsdienst für offene Netzwerksysteme

Inhalt

Einführung

In einer offenen Netzwerk-Computing-Umgebung kann es nicht vertrauenswürdig sein, dass eine Workstation die Benutzer der Netzwerkservices korrekt identifiziert. Kerberos bietet einen alternativen Ansatz, bei dem ein vertrauenswürdiger Drittanbieter-Authentifizierungsdienst verwendet wird, um die Identität von Benutzern zu überprüfen. Dieses Whitepaper gibt einen Überblick über das Kerberos-Authentifizierungsmodell, wie es für das MIT-Projekt Athena implementiert wurde. Es beschreibt die Protokolle, die von Clients, Servern und Kerberos verwendet werden, um eine Authentifizierung zu erreichen. Außerdem wird das Management und die Replikation der erforderlichen Datenbank beschrieben. Die vom Benutzer, Programmierer und Administrator sichtbaren Ansichten von Kerberos werden beschrieben. Schließlich wird die Rolle von Kerberos im größeren Athena-Bild gegeben, zusammen mit einer Liste von Anwendungen, die derzeit Kerberos für die Benutzerauthentifizierung verwenden. Wir beschreiben die Ergänzung der Kerberos-Authentifizierung zum Sun Network File System als Fallstudie zur Integration von Kerberos in eine vorhandene Anwendung.

Kerberos-Autoren

- Jennifer G. Steiner, Project Athena, Massachusetts Institute of Technology, Cambridge, MA 02139, steiner@ATHENA.MIT.EDU
- Clifford Neuman, Department of Computer Science, FR-35, University of Washington, Seattle, WA 98195, bcn@CS.WASHINGTON.EDU. Clifford Neuman war während der Design- und Implementierungsphase von Kerberos Mitglied des Projekts Athena.
- Jeffrey I. Schiller, Project Athena, Massachusetts Institute of Technology, Cambridge, MA 02139, jis@ATHENA.MIT.EDU

Einführung in Kerberos

Dieses Whitepaper gibt einen Überblick über Kerberos, ein von Miller und Neuman entwickeltes Authentifizierungssystem. für offene Netzwerk-Computing-Umgebungen und beschreibt unsere Erfahrungen mit deren Verwendung im MIT-Projekt Athena. Im Abschnitt zur Motivation erklären wir, warum ein neues Authentifizierungsmodell für offene Netzwerke erforderlich ist und welche Anforderungen darin bestehen. Das Was ist Kerberos? -Abschnitt listet die Komponenten der Kerberos-Software auf und beschreibt, wie sie bei der Bereitstellung des Authentifizierungsdienstes interagieren. Im Abschnitt Kerberos Names wird das Kerberos-Benennungsschema beschrieben.

Wie Kerberos arbeitet, zeigt die Bausteine der Kerberos-Authentifizierung - das Ticket und den Authentifizierer. Dies führt zu einer Diskussion der beiden Authentifizierungsprotokolle: die erste Authentifizierung eines Benutzers für Kerberos (analog zur Anmeldung) und das Protokoll für die gegenseitige Authentifizierung eines potenziellen Verbrauchers und eines potenziellen Herstellers eines Netzwerkdienstes.

Kerberos benötigt eine Datenbank mit Informationen über seine Clients. Im Abschnitt Kerberos-Datenbank werden die Datenbank, ihre Verwaltung und das Protokoll für ihre Änderung beschrieben. Der Abschnitt Kerberos Von der Außenansicht aus beschreibt die Kerberos-Schnittstelle für Benutzer, Anwendungsprogrammierer und Administratoren. Im Abschnitt The Bigger Picture (Das große Bild) wird beschrieben, wie das Projekt Athena Kerberos in die übrige Umgebung von Athena passt. Wir beschreiben auch die Interaktion verschiedener Kerberos-

Authentifizierungsdomänen oder -Bereiche. in unserem Fall die Beziehung zwischen dem Projekt Athena Kerberos und den Kerberos, die am Labor für Informatik des MIT ausgeführt werden.

Im Abschnitt "Probleme und offene Probleme" werden offene Probleme und noch ungelöste Probleme erwähnt. Der letzte Abschnitt gibt den aktuellen Status von Kerberos bei Project Athena. Im Anhang wird detailliert beschrieben, wie Kerberos auf einen Netzwerkdateidienst angewendet wird, um Benutzer zu authentifizieren, die Zugriff auf Remote-Dateisysteme erhalten möchten.

Kerberos-Konzepte

In diesem Whitepaper verwenden wir Begriffe, die möglicherweise mehrdeutig sind, für den Leser neu sind oder an anderer Stelle anders verwendet werden. Im Folgenden werden die Verwendung dieser Begriffe erläutert.

Benutzer, Client, Server: Ein Benutzer, der ein Programm oder einen Dienst verwendet. Ein Client verwendet auch etwas, ist aber nicht notwendigerweise eine Person. kann es ein Programm sein. Netzwerkanwendungen bestehen häufig aus zwei Teilen. ein Programm, das auf einem Computer ausgeführt wird und einen Remote-Dienst anfordert, und ein anderes Programm, das auf dem Remotecomputer ausgeführt wird und diesen Dienst ausführt. Diese werden jeweils als Client- bzw. Serverseite der Anwendung bezeichnet. Häufig kontaktiert ein Client einen Server im Namen eines Benutzers.

Jede Entität, die das Kerberos-System verwendet, sei es ein Benutzer oder ein Netzwerkserver, ist in gewissem Sinne ein Client, da sie den Kerberos-Dienst verwendet. Um Kerberos-Clients von Clients anderer Dienste zu unterscheiden, verwenden wir den Begriff Principal, um eine solche Einheit anzugeben. Beachten Sie, dass ein Kerberos-Principal entweder ein Benutzer oder ein Server sein kann. (Die Benennung der Kerberos-Prinzipale wird in einem späteren Abschnitt beschrieben.)

Service vs. Server - Wir verwenden Service als abstrakte Spezifikation einiger durchzuführender Aktionen. Ein Prozess, der diese Aktionen ausführt, wird als Server bezeichnet. Zu einem bestimmten Zeitpunkt können mehrere Server (die in der Regel auf verschiedenen Computern ausgeführt werden) einen bestimmten Dienst ausführen. In Athena gibt es beispielsweise einen BSD UNIX-Rlogin-Server, der auf jedem unserer Timesharing-Rechner ausgeführt wird.

Schlüssel, privater Schlüssel, Passwort: Kerberos verwendet Verschlüsselung mit privatem Schlüssel. Jedem Kerberos-Principal wird eine große Zahl zugewiesen, ihr privater Schlüssel, der nur diesem Principal und Kerberos bekannt ist. Bei einem Benutzer ist der private Schlüssel das Ergebnis einer einseitigen Funktion, die auf das Kennwort des Benutzers angewendet wird. Wir verwenden Schlüssel als Tastenkombination für privaten Schlüssel.

Anmeldeinformationen - Leider hat dieses Wort eine besondere Bedeutung sowohl für das Sun Network File System als auch für das Kerberos-System. Wir geben explizit an, ob es sich um NFS-Anmeldeinformationen oder Kerberos-Anmeldeinformationen handelt, andernfalls wird der Begriff im normalen englischen Sprachsinn verwendet.

Master und Slave: Es ist möglich, Kerberos-Authentifizierungssoftware auf mehr als einem Computer auszuführen. Allerdings gibt es immer nur eine endgültige Kopie der Kerberos-Datenbank. Der Computer, auf dem diese Datenbank gespeichert ist, wird als Master-Computer oder nur als Master bezeichnet. Andere Computer verfügen möglicherweise über schreibgeschützte Kopien der Kerberos-Datenbank, die als Slaves bezeichnet werden.

Motivation hinter Kerberos

In einer nicht vernetzten PC-Umgebung können Ressourcen und Informationen durch eine physische Sicherung des PCs geschützt werden. In einer zeitgleichen Computing-Umgebung schützt das Betriebssystem die Benutzer vor einander und steuert die Ressourcen. Um festzustellen, was jeder Benutzer lesen oder ändern kann, muss das Timesharing-System jeden Benutzer identifizieren. Dies geschieht, wenn sich der Benutzer anmeldet.

In einem Netzwerk von Benutzern, die Services von mehreren separaten Computern benötigen, gibt es drei Ansätze, die Sie für die Zugriffskontrolle verwenden können: Man kann nichts tun, indem man sich auf den Computer verlässt, an dem der Benutzer angemeldet ist, um nicht autorisierten Zugriff zu verhindern. Man kann verlangen, dass der Host seine Identität nachweist, aber dem Wort des Hosts, wer der Benutzer ist, vertrauen. oder den Benutzer auffordern, seine Identität für jeden erforderlichen Dienst nachzuweisen.

In einer geschlossenen Umgebung, in der alle Maschinen unter strenger Kontrolle stehen, kann man den ersten Ansatz verwenden. Wenn die Organisation alle Hosts kontrolliert, die über das Netzwerk kommunizieren, ist dies ein vernünftiger Ansatz.

In einer offeneren Umgebung kann man nur jenen Hosts unter organisatorischer Kontrolle vertrauen. In diesem Fall muss jeder Host seine Identität nachweisen. Die rlogin- und rsh-Programme verwenden diesen Ansatz. In diesen Protokollen erfolgt die Authentifizierung durch Überprüfung der Internetadresse, von der aus eine Verbindung hergestellt wurde.

Im Athena-Umfeld müssen wir Anfragen von Hosts, die nicht unter organisatorischer Kontrolle stehen, erfüllen können. Benutzer haben die vollständige Kontrolle über ihre Workstations: sie können sie neu starten, eigenständig hochfahren oder sogar von ihren eigenen Bändern booten. Daher muss der dritte Ansatz verfolgt werden. Der Benutzer muss seine Identität für jeden gewünschten Service nachweisen. Der Server muss auch seine Identität nachweisen. Die physische Sicherung des Hosts, auf dem ein Netzwerkserver ausgeführt wird, reicht nicht aus. jemand an einer anderen Stelle des Netzwerks kann sich als der angegebene Server tarnen.

Unsere Umgebung stellt mehrere Anforderungen an einen Identifikationsmechanismus. Erstens muss es sicher sein. Es muss schwierig genug sein, einen potenziellen Angreifer daran zu hindern, den Authentifizierungsmechanismus als Schwachstelle zu erkennen. Jemand, der das Netzwerk überwacht, sollte nicht in der Lage sein, die Informationen abzurufen, die erforderlich sind, um die Identität eines anderen Benutzers anzunehmen. Zweitens muss sie zuverlässig sein. Der Zugriff auf viele Dienste ist vom Authentifizierungsdienst abhängig. Wenn es nicht zuverlässig ist, wird das System der Dienstleistungen als Ganzes nicht sein. Drittens sollte sie transparent sein. Im Idealfall sollte der Benutzer nicht wissen, dass eine Authentifizierung stattfindet. Schließlich sollte es skalierbar sein. Viele Systeme können mit Athena-Hosts kommunizieren. Nicht alle werden unseren Mechanismus unterstützen, aber die Software sollte nicht kaputt gehen, wenn sie es tut.

Kerberos ist das Ergebnis unserer Arbeit, die oben genannten Anforderungen zu erfüllen. Wenn ein Benutzer zu einer Workstation geht, meldet er sich an. Soweit der Benutzer weiß, reicht diese erste Identifikation aus, um die Identität aller erforderlichen Netzwerkserver für die Dauer der Anmeldungssitzung nachzuweisen. Die Sicherheit von Kerberos beruht auf der Sicherheit mehrerer Authentifizierungsserver, jedoch nicht auf dem System, von dem aus sich Benutzer anmelden, und nicht auf der Sicherheit der Endserver, die verwendet werden. Der Authentifizierungsserver bietet einem ordnungsgemäß authentifizierten Benutzer die Möglichkeit, seine Identität gegenüber Servern zu überprüfen, die im Netzwerk verstreut sind.

Authentifizierung ist ein grundlegender Baustein für eine sichere Netzwerkumgebung. Wenn beispielsweise ein Server die Identität eines Clients kennt, kann er entscheiden, ob er den Dienst bereitstellen soll, ob ihm besondere Berechtigungen gewährt werden sollen, wer die Rechnung für den Dienst erhalten soll usw. Anders ausgedrückt: Autorisierungs- und Abrechnungsschemata können auf der von Kerberos bereitgestellten Authentifizierung aufbauen, was zu einer gleichwertigen Sicherheit wie der einzelne PC oder das Timesharing-System führt.

Was ist Kerberos?

Kerberos ist ein vertrauenswürdiger Drittanbieter-Authentifizierungsdienst, der auf dem von Needham und Schröder präsentierten Modell basiert. Es ist in dem Sinne vertraut, dass jeder seiner Kunden Kerberos's Urteilsvermögen in Bezug auf die Identität seiner anderen Kunden als zutreffend ansieht. Zeitstempel (große Zahlen, die das aktuelle Datum und die aktuelle Uhrzeit darstellen) wurden dem ursprünglichen Modell hinzugefügt, um die Erkennung von Wiederholungen zu erleichtern. Die Wiedergabe erfolgt, wenn eine Nachricht aus dem Netzwerk gestohlen und zu einem späteren Zeitpunkt erneut gesendet wird. Eine ausführlichere Beschreibung der Wiederholung und anderer Authentifizierungsprobleme finden Sie unter Voydock und Kent.

Was tut Kerberos?

Kerberos unterhält eine Datenbank seiner Clients und ihrer privaten Schlüssel. Der private Schlüssel ist eine große Zahl, die nur Kerberos und dem Client, zu dem er gehört, bekannt ist. Wenn der Client ein Benutzer ist, handelt es sich um ein verschlüsseltes Kennwort. Netzwerkdienste, die eine Authentifizierung erfordern, müssen bei Kerberos registriert werden, ebenso wie Clients, die diese Dienste nutzen möchten. Die privaten Schlüssel werden bei der Registrierung ausgehandelt.

Da Kerberos diese privaten Schlüssel kennt, kann es Nachrichten erstellen, die einen Kunden davon überzeugen, dass ein anderer wirklich der ist, der er vorgibt zu sein. Kerberos generiert auch temporäre private Schlüssel, so genannte Sitzungsschlüssel, die an zwei Clients und niemanden weitergegeben werden. Mit einem Sitzungsschlüssel können Nachrichten zwischen zwei Parteien verschlüsselt werden.

Kerberos bietet drei verschiedene Schutzstufen. Der Anwendungsprogrammierer bestimmt, welche entsprechend den Anforderungen der Anwendung geeignet ist. Beispielsweise erfordern einige Anwendungen nur, dass die Authentizität bei der Initiierung einer Netzwerkverbindung festgelegt wird. Außerdem kann davon ausgegangen werden, dass weitere Nachrichten aus einer bestimmten Netzwerkadresse vom authentifizierten Teilnehmer stammen. Unser authentifiziertes Netzwerk-Dateisystem nutzt diese Sicherheitsstufe.

Andere Anwendungen erfordern eine Authentifizierung jeder Nachricht, es ist jedoch egal, ob der Inhalt der Nachricht offen gelegt wird oder nicht. Für diese stellt Kerberos sichere Nachrichten bereit. Eine höhere Sicherheitsstufe wird jedoch durch private Nachrichten erreicht, bei denen jede Nachricht nicht nur authentifiziert, sondern auch verschlüsselt wird. Private Nachrichten werden beispielsweise vom Kerberos-Server selbst zum Senden von Passwörtern über das Netzwerk verwendet.

Kerberos-Softwarekomponenten

Die Athena-Implementierung umfasst mehrere Module:

- Kerberos-Anwendungsbibliothek
- Verschlüsselungsbibliothek
- Datenbankbibliothek
- Datenbankverwaltungsprogramme
- Administrationsserver
- Authentifizierungsserver
- DB-Verbreitungssoftware
- Benutzerprogramme
- Anwendungen

Die Kerberos-Anwendungsbibliothek stellt eine Schnittstelle für Anwendungs-Clients und Anwendungsserver bereit. Sie enthält u. a. Routinen zum Erstellen oder Lesen von Authentifizierungsanforderungen sowie die Routinen zum Erstellen sicherer oder privater Nachrichten.

Die Verschlüsselung in Kerberos basiert auf DES, dem Data Encryption Standard. Die Verschlüsselungsbibliothek implementiert diese Routinen. Es stehen verschiedene Verschlüsselungsmethoden zur Verfügung, die einen Kompromiss zwischen Geschwindigkeit und Sicherheit darstellen. Eine Erweiterung des DES Cipher Block Chaining (CBC)-Modus, der so genannte Propagating CBC-Modus, wird ebenfalls bereitgestellt. Im CBC wird ein Fehler nur über den aktuellen Codeblock verbreitet, während der Fehler im PCBC über die gesamte Nachricht verbreitet wird. Dadurch wird die gesamte Meldung bei Auftreten eines Fehlers nutzlos, nicht nur ein Teil davon. Die Verschlüsselungsbibliothek ist ein unabhängiges Modul und kann durch andere DES-Implementierungen oder eine andere Verschlüsselungsbibliothek ersetzt werden.

Ein weiteres austauschbares Modul ist das Datenbankmanagementsystem. Die aktuelle Athena-Implementierung der Datenbankbibliothek verwendet ndbm, obwohl Ingres ursprünglich verwendet wurde. Es können auch andere Bibliotheken für das Datenbankmanagement verwendet werden.

Die Kerberos-Datenbankanforderungen sind einfach. für jeden Hauptverpflichteten wird ein Datensatz mit dem Namen, dem privaten Schlüssel und dem Ablaufdatum des Hauptverpflichteten sowie einigen administrativen Informationen geführt. (Das Ablaufdatum ist das Datum, nach dem ein Eintrag nicht mehr gültig ist. Bei der Registrierung ist sie in der Regel auf ein paar Jahre in der Zukunft angesetzt.)

Andere Benutzerinformationen wie der tatsächliche Name, die Telefonnummer usw. werden von einem anderen Server, dem Hesiod-Namenserver, gespeichert. Auf diese Weise kann Kerberos sensible Informationen, insbesondere Kennwörter, mit relativ hohen Sicherheitsmaßnahmen behandeln. während die von Hesiod aufbewahrten nicht sensiblen Informationen anders behandelt werden; sie kann beispielsweise unverschlüsselt über das Netzwerk gesendet werden.

Die Kerberos-Server verwenden die Datenbankbibliothek ebenso wie die Tools für die Verwaltung der Datenbank.

Der Administrationsserver (oder KDBM-Server) stellt der Datenbank eine Lese- und Schreibnetzwerkschnittstelle zur Verfügung. Die Clientseite des Programms kann auf jedem Computer im Netzwerk ausgeführt werden. Die Serverseite muss jedoch auf dem Rechner ausgeführt werden, auf dem sich die Kerberos-Datenbank befindet, um Änderungen an der Datenbank vorzunehmen.

Der Authentifizierungsserver (oder Kerberos-Server) dagegen führt Lesevorgänge in der Kerberos-Datenbank durch, d. h. die Authentifizierung von Prinzipien und die Generierung von Sitzungsschlüsseln. Da dieser Server die Kerberos-Datenbank nicht ändert, kann er auf einem

Rechner ausgeführt werden, der eine schreibgeschützte Kopie der Kerberos-Master-Datenbank enthält.

Datenbankpropagierungs-Software verwaltet die Replikation der Kerberos-Datenbank. Es ist möglich, Kopien der Datenbank auf mehreren verschiedenen Computern zu haben, wobei auf jedem Rechner eine Kopie des Authentifizierungsservers ausgeführt wird. Jeder dieser Slave-Rechner erhält in bestimmten Abständen eine Aktualisierung der Kerberos-Datenbank vom Master-Rechner.

Schließlich gibt es Endbenutzerprogramme, um sich bei Kerberos anzumelden, ein Kerberos-Passwort zu ändern und Kerberos-Tickets anzuzeigen oder zu zerstören (Tickets werden später erklärt).

Kerberos-Namen

Teil der Authentifizierung einer Einheit ist die Benennung. Bei der Authentifizierung wird überprüft, ob der Client der in einer Anforderung benannte Client ist. Woraus besteht ein Name? In Kerberos werden Benutzer und Server benannt. Was den Authentifizierungsserver angeht, so sind sie gleichwertig. Ein Name besteht aus einem primären Namen, einer Instanz und einem Bereich, der als name.instance@realm ausgedrückt wird.

Der primäre Name ist der Name des Benutzers oder des Dienstes. Die Instanz wird verwendet, um zwischen Variationen des primären Namens zu unterscheiden. Für Benutzer kann eine Instanz besondere Berechtigungen beinhalten, z. B. die "root"- oder "admin"-Instanzen. Bei Diensten in der Athena-Umgebung ist die Instanz normalerweise der Name des Computers, auf dem der Server ausgeführt wird. Der Rlogin-Dienst verfügt beispielsweise über verschiedene Instanzen auf verschiedenen Hosts: rlogin.priam ist der Rlogin-Server auf dem Host namens priam. Ein Kerberos-Ticket ist nur für einen einzelnen benannten Server geeignet. Daher ist ein separates Ticket erforderlich, um Zugriff auf verschiedene Instanzen desselben Dienstes zu erhalten. Der Bereich ist der Name einer administrativen Entität, die Authentifizierungsdaten verwaltet. Zum Beispiel können verschiedene Institutionen jeweils ihre eigene Kerberos-Maschine haben, die eine andere Datenbank enthält. Sie haben verschiedene Kerberos-Realms. (Bereiche werden in der Interaktion mit anderen Kerberi genauer besprochen.)

Funktionsweise von Kerberos

In diesem Abschnitt werden die Kerberos-Authentifizierungsprotokolle beschrieben. Wie bereits erwähnt, basiert das Kerberos-Authentifizierungsmodell auf dem Schlüssel-Verteilungsprotokoll von Needham und Schröder. Wenn ein Benutzer einen Dienst anfordert, muss seine Identität festgelegt werden. Dazu wird dem Server ein Ticket präsentiert, zusammen mit dem Nachweis, dass das Ticket ursprünglich dem Benutzer ausgestellt wurde, nicht gestohlen. Die Authentifizierung über Kerberos erfolgt in drei Phasen. In der ersten Phase erhält der Benutzer Anmeldeinformationen, um den Zugriff auf andere Dienste anzufordern. In der zweiten Phase fordert der Benutzer Authentifizierung für einen bestimmten Dienst an. In der letzten Phase legt der Benutzer diese Anmeldeinformationen dem Endserver vor.

Kerberos-Anmeldeinformationen

Im Kerberos-Authentifizierungsmodell werden zwei Arten von Anmeldeinformationen verwendet: Tickets und Authentifizierer. Beide basieren auf der Verschlüsselung des privaten Schlüssels,

werden jedoch mit unterschiedlichen Schlüsseln verschlüsselt. Ein Ticket wird verwendet, um die Identität der Person, an die das Ticket ausgestellt wurde, sicher zwischen dem Authentifizierungsserver und dem Endserver weiterzugeben. Ein Ticket gibt auch Informationen weiter, mit denen sichergestellt werden kann, dass die Person, die das Ticket benutzt, die Person ist, der es ausgestellt wurde. Der Authentifizierer enthält die zusätzlichen Informationen, die im Vergleich zu den Angaben im Ticket belegen, dass der Kunde, der das Ticket präsentiert, die gleiche ist, für die das Ticket ausgestellt wurde.

Ein Ticket ist gut für einen einzelnen Server und einen einzigen Client. Sie enthält den Namen des Servers, den Namen des Clients, die Internetadresse des Clients, einen Zeitstempel, eine Lebensdauer und einen zufälligen Sitzungsschlüssel. Diese Informationen werden mit dem Schlüssel des Servers verschlüsselt, für den das Ticket verwendet wird. Sobald das Ticket ausgestellt wurde, kann es vom benannten Client mehrmals verwendet werden, um Zugriff auf den genannten Server zu erhalten, bis das Ticket abläuft. Da das Ticket im Schlüssel des Servers verschlüsselt ist, ist es sicher, dass der Benutzer das Ticket an den Server weiterleiten kann, ohne sich Sorgen über die Änderung des Tickets machen zu müssen.

Im Gegensatz zum Ticket kann der Authentifizierer nur einmal benutzt werden. Jedes Mal, wenn ein Client einen Dienst verwenden möchte, muss ein neuer generiert werden. Dies stellt kein Problem dar, da der Client den Authentifizierer selbst erstellen kann. Ein Authentifizierer enthält den Namen des Clients, die IP-Adresse der Workstation und die aktuelle Workstation-Zeit. Der Authentifizierer wird im Sitzungsschlüssel verschlüsselt, der Teil des Tickets ist.

Kerberos-Ticket anfordern

Wenn der Benutzer zu einer Workstation geht, kann nur eine einzige Information seine Identität belegen: das Kennwort des Benutzers. Der erste Austausch mit dem Authentifizierungsserver soll das Risiko einer Passwortkompromittierung minimieren und gleichzeitig einem Benutzer die korrekte Authentifizierung ohne Kenntnis des Passworts ermöglichen. Der Anmeldevorgang scheint für den Benutzer derselbe wie die Anmeldung bei einem Timesharing-System zu sein. Hinter den Kulissen ist es jedoch ganz anders.

Der Benutzer wird aufgefordert, seinen Benutzernamen einzugeben. Nach der Eingabe wird eine Anfrage an den Authentifizierungsserver gesendet, der den Benutzernamen und den Namen eines speziellen Dienstes, den Ticket-Gewährungsdienst, enthält.

Der Authentifizierungsserver überprüft, ob er über den Client Bescheid weiß. Wenn dies der Fall ist, wird ein zufälliger Sitzungsschlüssel generiert, der später zwischen dem Client und dem Ticket-Server verwendet wird. Anschließend wird ein Ticket für den Ticket-Server erstellt, der den Namen des Kunden, den Namen des Ticket-Server, die aktuelle Uhrzeit, die Lebensdauer des Tickets, die IP-Adresse des Kunden und den soeben erstellten zufälligen Sitzungsschlüssel enthält. Diese werden in einem Schlüssel verschlüsselt, der nur dem Ticket-Server und dem Authentifizierungsserver bekannt ist.

Der Authentifizierungsserver sendet dann das Ticket zusammen mit einer Kopie des zufälligen Sitzungsschlüssels und einigen zusätzlichen Informationen zurück an den Client. Diese Antwort wird im privaten Schlüssel des Clients verschlüsselt, der nur Kerberos und dem Client bekannt ist und vom Kennwort des Benutzers abgeleitet wird.

Nachdem der Kunde die Antwort erhalten hat, wird er um das Kennwort gebeten. Das Kennwort wird in einen DES-Schlüssel konvertiert und zum Entschlüsseln der Antwort vom Authentifizierungsserver verwendet. Das Ticket und der Sitzungsschlüssel werden zusammen mit

einigen anderen Informationen für die zukünftige Verwendung gespeichert, und das Benutzerkennwort und der DES-Schlüssel werden aus dem Speicher gelöscht.

Nachdem der Austausch abgeschlossen ist, verfügt der Arbeitsplatz über Informationen, mit denen er die Identität seines Benutzers für die Lebensdauer des Tickets nachweisen kann. Solange die Software auf dem Arbeitsplatzrechner noch nicht manipuliert war, existieren keine Informationen, die es anderen ermöglichen, die Identität des Benutzers über die Lebensdauer des Tickets hinaus zu übernehmen.

Kerberos-Service anfordern

Im Moment sollten wir so tun, als hätte der Benutzer bereits ein Ticket für den gewünschten Server. Um Zugriff auf den Server zu erhalten, erstellt die Anwendung einen Authentifizierer, der den Namen und die IP-Adresse des Clients sowie die aktuelle Uhrzeit enthält. Der Authentifizierer wird dann im Sitzungsschlüssel verschlüsselt, der mit dem Ticket für den Server empfangen wurde. Der Client sendet dann den Authentifizierer zusammen mit dem Ticket auf eine von der jeweiligen Anwendung definierte Weise an den Server.

Sobald der Authentifizierer und das Ticket beim Server eingegangen sind, entschlüsselt der Server das Ticket, verwendet den im Ticket enthaltenen Sitzungsschlüssel zur Entschlüsselung des Authentifizierers, vergleicht die Ticketinformationen mit den Informationen im Ticket im Authentifizierer, mit der IP-Adresse, von der die Anfrage empfangen wurde, und mit der aktuellen Uhrzeit. Wenn alles übereinstimmt, kann die Anforderung fortgesetzt werden.

Es wird davon ausgegangen, dass Uhren innerhalb von wenigen Minuten synchronisiert werden. Wenn die Zeit in der Anforderung zu weit in der Zukunft oder in der Vergangenheit liegt, behandelt der Server die Anforderung als Versuch, eine vorherige Anforderung erneut abzuspielen. Der Server ist auch berechtigt, alle vergangenen Anfragen mit noch gültigen Zeitstempeln zu verfolgen. Um weitere Angriffe zu verhindern, kann eine Anfrage, die mit demselben Ticket und Zeitstempel wie eine bereits erhaltene gesendet wurde, verworfen werden.

Wenn der Client schließlich angibt, dass der Server seine Identität auch überprüfen soll, fügt der Server dem Zeitstempel, den der Client im Authentifizierer gesendet hat, verschlüsselt das Ergebnis im Sitzungsschlüssel und sendet das Ergebnis zurück an den Client.

Am Ende dieses Austausches ist der Server sicher, dass, nach Kerberos, der Client ist, was er sagt. Bei gegenseitiger Authentifizierung ist der Client auch überzeugt, dass der Server authentisch ist. Darüber hinaus teilen der Client und der Server einen Schlüssel, den niemand sonst kennt, und können sicher davon ausgehen, dass eine in diesem Schlüssel verschlüsselte, relativ aktuelle Nachricht von der anderen Partei stammt.

Kerberos Server-Tickets herunterladen

Denken Sie daran, dass ein Ticket nur für einen einzelnen Server geeignet ist. Daher ist es notwendig, für jeden Dienst, den der Kunde nutzen möchte, ein separates Ticket zu erwerben. Tickets für einzelne Server erhalten Sie über den Ticketservice. Da der Ticket-Gewährungsdienst selbst ein Service ist, wird das im vorherigen Abschnitt beschriebene Service Access Protocol verwendet.

Wenn für ein Programm ein Ticket erforderlich ist, das noch nicht angefordert wurde, sendet es eine Anfrage an den Ticket-Erteilungs-Server. Die Anfrage enthält den Namen des Servers, für den ein Ticket angefordert wird, sowie das Ticket zur Ticketerstellung und einen Authentifizierer,

der wie im vorherigen Abschnitt beschrieben aufgebaut wurde.

Der Ticketvergabeserver prüft dann den Authentifizierer und das Ticket-Ticket wie oben beschrieben. Wenn gültig, generiert der Server, der Tickets gewährt, einen neuen zufälligen Sitzungsschlüssel, der zwischen dem Client und dem neuen Server verwendet wird. Anschließend wird ein Ticket für den neuen Server erstellt, das den Client-Namen, den Servernamen, die aktuelle Uhrzeit, die IP-Adresse des Clients und den neuen Sitzungsschlüssel enthält, den er gerade generiert hat. Die Lebensdauer des neuen Tickets ist das Minimum der Restlebensdauer für das Ticket-Ticket und der Standardwert für den Service.

Der Ticket-Server sendet dann das Ticket zusammen mit dem Sitzungsschlüssel und anderen Informationen an den Kunden zurück. Diesmal wird die Antwort jedoch in dem Sitzungsschlüssel verschlüsselt, der Teil des Ticket-Tickets war. Auf diese Weise muss der Benutzer sein Passwort nicht erneut eingeben.

Die Kerberos-Datenbank

Bis zu diesem Zeitpunkt haben wir über Vorgänge gesprochen, die schreibgeschützten Zugriff auf die Kerberos-Datenbank erfordern. Diese Vorgänge werden vom Authentifizierungsdienst ausgeführt, der sowohl auf Master- als auch auf Slave-Computern ausgeführt werden kann.

In diesem Abschnitt werden Vorgänge erläutert, die Schreibzugriff auf die Datenbank erfordern. Diese Operationen werden vom Administrationsdienst, dem Kerberos Database Management Service (KDBM), ausgeführt. Die aktuelle Implementierung sieht vor, dass Änderungen nur an der Kerberos-Master-Datenbank vorgenommen werden dürfen. Slave-Kopien sind schreibgeschützt. Daher kann der KDBM-Server nur auf dem Kerberos-Master ausgeführt werden.

Beachten Sie, dass die Authentifizierung zwar noch möglich ist (auf Slaves), Verwaltungsanfragen jedoch nicht bedient werden können, wenn der Master-Computer ausgefallen ist. Unserer Erfahrung nach ist dies kein Problem, da Verwaltungsanfragen selten sind.

Das KDBM bearbeitet Anfragen von Benutzern zum Ändern ihrer Kennwörter. Die Clientseite dieses Programms, das Anfragen über das Netzwerk an das KDBM sendet, ist das kpasswd-Programm. Das KDBM akzeptiert auch Anfragen von Kerberos-Administratoren, die der Datenbank Prinzipale hinzufügen und Kennwörter für vorhandene Prinzipale ändern können. Die Client-Seite des Administrationsprogramms, das auch Anfragen über das Netzwerk an das KDBM sendet, ist das kadmin-Programm.

Der KDBM-Server

Der KDBM-Server akzeptiert Anfragen zum Hinzufügen von Prinzipalen zur Datenbank oder zum Ändern der Kennwörter für vorhandene Prinzipale. Dieser Service ist insofern einzigartig, als der Ticketservice keine Tickets dafür ausstellt. Stattdessen muss der Authentifizierungsdienst selbst verwendet werden (derselbe Dienst, der für den Erhalt eines Tickets verwendet wird). Der Benutzer muss ein Kennwort eingeben. Wenn dies nicht der Fall ist, kann ein Benutzer, der seine Workstation unbeaufsichtigt verlassen hat, einen Passanten aufsuchen und sein Passwort für ihn ändern, was verhindert werden sollte. Wenn ein Administrator die Workstation nicht bewacht hat, kann ein Passwort im System geändert werden.

Wenn der KDBM-Server eine Anforderung empfängt, autorisiert er diese, indem er den authentifizierten Hauptnamen des Antragstellers der Änderung mit dem Hauptnamen des Ziels der

Anforderung vergleicht. Wenn sie identisch sind, ist die Anforderung zulässig. Wenn sie nicht identisch sind, ruft der KDBM-Server eine Zugriffskontrollliste ab (die in einer Datei auf dem Kerberos-Master-System gespeichert ist). Wenn der Hauptname des Antragstellers in dieser Datei gefunden wird, ist die Anforderung zulässig, ansonsten wird sie abgelehnt.

Standardmäßig werden Namen mit einer NULL-Instanz (der Standardinstanz) nicht in der Zugriffssteuerungslistendatei angezeigt. Stattdessen wird eine Admin-Instanz verwendet. Damit ein Benutzer Administrator von Kerberos wird, muss daher eine Admin-Instanz für diesen Benutzernamen erstellt und der Zugriffskontrollliste hinzugefügt werden. Diese Konvention ermöglicht es einem Administrator, ein anderes Kennwort für die Kerberos-Administration zu verwenden, als sie/er für die normale Anmeldung verwenden würde.

Alle Anfragen an das KDBM-Programm werden protokolliert, unabhängig davon, ob sie zugelassen oder abgelehnt werden.

Die kadmin- und kpasswd-Programme

Administratoren von Kerberos verwenden das Kadmin-Programm, um der Datenbank Prinzipale hinzuzufügen oder die Kennwörter vorhandener Prinzipale zu ändern. Ein Administrator muss das Kennwort für den Admin-Instanznamen eingeben, wenn er das kadmin-Programm aufruft. Dieses Kennwort wird verwendet, um ein Ticket für den KDBM-Server abzurufen.

Benutzer können ihre Kerberos-Passwörter mit dem kpasswd-Programm ändern. Sie müssen beim Aufruf des Programms ihr altes Kennwort eingeben. Dieses Kennwort wird verwendet, um ein Ticket für den KDBM-Server abzurufen.

Kerberos-Datenbankreplikation

Jeder Kerberos-Bereich verfügt über einen Kerberos-Master-Rechner, der die Master-Kopie der Authentifizierungsdatenbank enthält. Es ist möglich (wenn auch nicht notwendig), zusätzliche, schreibgeschützte Kopien der Datenbank auf Slave-Maschinen an anderen Stellen im System zu haben. Mehrere Kopien der Datenbank bieten die Vorteile, die üblicherweise für die Replikation genannt werden: höhere Verfügbarkeit und bessere Leistung. Wenn der Master-Rechner ausgefallen ist, kann die Authentifizierung auf einem der Slave-Systeme trotzdem erfolgen. Die Möglichkeit, die Authentifizierung auf einem von mehreren Computern durchzuführen, reduziert die Wahrscheinlichkeit eines Engpasses auf dem Master-Rechner.

Wenn Sie mehrere Kopien der Datenbank aufbewahren, tritt das Problem der Datenkonsistenz auf. Wir haben festgestellt, dass sehr einfache Methoden ausreichen, um mit Inkonsistenzen umzugehen. Die Master-Datenbank wird stündlich gelöscht. Die Datenbank wird in ihrer Gesamtheit an die Slave-Maschinen gesendet, die dann ihre eigenen Datenbanken aktualisieren. Ein Programm auf dem Master-Host, genannt kprop, sendet das Update an ein Peer-Programm namens kpropd, das auf jedem der Slave-Computer ausgeführt wird. First kprop sendet eine Prüfsumme der neuen Datenbank, die es zu senden beabsichtigt. Die Prüfsumme wird im Kerberos-Master-Datenbankschlüssel verschlüsselt, den sowohl der Master- als auch der Slave-Kerberos-Computer besitzen. Die Daten werden dann über das Netzwerk an den "kpropd" auf dem Slave-Rechner übertragen. Der Slave-Weiterleitungsserver berechnet eine Prüfsumme der empfangenen Daten. Wenn diese mit der vom Master gesendeten Prüfsumme übereinstimmen, werden die neuen Informationen zum Aktualisieren der Slave-Datenbank verwendet.

Alle Kennwörter in der Kerberos-Datenbank sind im Master-Datenbankschlüssel verschlüsselt.

Aus diesem Grund sind die Informationen, die von Master an Slave über das Netzwerk weitergegeben werden, für einen Lauschangler nicht hilfreich. Es ist jedoch wichtig, dass nur Informationen vom Master-Host von den Slaves akzeptiert werden und dass Manipulationen von Daten erkannt werden, also die Prüfsumme.

Kerberos von außen

In diesem Abschnitt wird Kerberos aus praktischer Sicht beschrieben, zunächst vom Benutzer, dann vom Programmierer der Anwendung und schließlich von den Aufgaben des Kerberos-Administrators.

Kerberos-Benutzeransicht

Wenn alles gut geht, wird der Benutzer kaum bemerken, dass Kerberos anwesend ist. Im Rahmen unserer UNIX-Implementierung erhalten Sie das Ticket zur Ticketgewährung bei Kerberos im Rahmen des Anmeldeprozesses. Die Änderung des Kerberos-Passworts eines Benutzers ist Teil des Programms passwd. Und Kerberos-Tickets werden automatisch zerstört, wenn sich ein Benutzer abmeldet.

Wenn die Anmeldesitzung des Benutzers länger als die Lebensdauer des Ticket-Erlaubnistickets dauert (derzeit 8 Stunden), bemerkt der Benutzer Kerberos' Anwesenheit, da eine von Kerberos authentifizierte Anwendung beim nächsten Ausführen fehlschlägt. Das Kerberos-Ticket dafür ist abgelaufen. An dieser Stelle kann der Benutzer das Kinit-Programm ausführen, um ein neues Ticket für den Ticket-Server zu erhalten. Wie bei der Anmeldung muss ein Passwort eingegeben werden, um es zu erhalten. Ein Benutzer, der den Befehl klist aus Neugier ausführt, kann von allen Tickets überrascht sein, die unbemerkt in ihrem Namen für Dienste abgerufen wurden, die eine Kerberos-Authentifizierung erfordern.

Kerberos aus Sicht des Programmierers

Ein Programmierer, der eine Kerberos-Anwendung schreibt, fügt einer bereits vorhandenen Netzwerkanwendung, die aus einer Client- und Serverseite besteht, häufig Authentifizierung hinzu. Wir nennen diesen Prozess "Kerberizing" ein Programm. Bei der Kerberisierung wird normalerweise die Kerberos-Bibliothek angerufen, um bei der ersten Dienstanforderung eine Authentifizierung durchzuführen. Dies kann auch Anrufe an die DES-Bibliothek umfassen, um Nachrichten und Daten zu verschlüsseln, die anschließend zwischen dem Anwendungs-Client und dem Anwendungsserver gesendet werden.

Die gebräuchlichsten Bibliotheksfunktionen sind krb_mk_req auf Client-Seite und krb_rd_req auf Server-Seite. Die krb_mk_req Routine nimmt als Parameter den Namen, die Instanz und den Bereich des Zielservers an, der angefordert wird, und möglicherweise eine Prüfsumme der zu sendenden Daten. Der Client sendet dann die vom Aufruf krb_mk_req zurückgegebene Nachricht über das Netzwerk an die Serverseite der Anwendung. Wenn der Server diese Meldung empfängt, ruft er die Bibliotheksroutine krb_rd_req auf. Die Routine gibt ein Urteil über die Authentizität der mutmaßlichen Identität des Absenders zurück.

Wenn die Anwendung vorschreibt, dass Nachrichten, die zwischen Client und Server gesendet werden, geheim sind, können Bibliotheksaufrufe an krb_mk_priv (krb_rd_priv) durchgeführt werden, um Nachrichten im Sitzungsschlüssel zu verschlüsseln (zu entschlüsseln), den beide Seiten nun gemeinsam nutzen.

Die Aufgabe des Kerberos-Administrators

Der Kerberos-Administrator beginnt mit der Ausführung eines Programms zum Initialisieren der Datenbank. Es muss ein anderes Programm ausgeführt werden, um grundlegende Prinzipien in der Datenbank zu registrieren, z. B. den Kerberos-Administratornamen mit einer Admin-Instanz. Der Kerberos-Authentifizierungsserver und der Administrationsserver müssen gestartet werden. Wenn Slave-Datenbanken vorhanden sind, muss der Administrator veranlassen, dass die Programme, die Datenbankaktualisierungen von Master an Slaves übertragen, regelmäßig gestartet werden.

Nachdem diese ersten Schritte ausgeführt wurden, bearbeitet der Administrator die Datenbank über das Netzwerk mithilfe des Programms kadmin. Mithilfe dieses Programms können neue Prinzipale hinzugefügt und Kennwörter geändert werden.

Insbesondere wenn dem System eine neue Kerberos-Anwendung hinzugefügt wird, muss der Kerberos-Administrator einige Schritte unternehmen, um sie zum Laufen zu bringen. Der Server muss in der Datenbank registriert und einem privaten Schlüssel zugewiesen werden (in der Regel ist dies ein automatisch generierter zufälliger Schlüssel). Anschließend müssen einige Daten (einschließlich des Serverschlüssels) aus der Datenbank extrahiert und in einer Datei auf dem Computer des Servers installiert werden. Die Standarddatei ist /etc/srvtab. Die vom Server benannte krb_rd_req-Bibliotheksroutine (siehe den vorherigen Abschnitt) verwendet die Informationen in dieser Datei, um Nachrichten zu entschlüsseln, die verschlüsselt im privaten Schlüssel des Servers gesendet wurden. Die Datei "/etc/srvtab" authentifiziert den Server als Kennwort, das an einem Terminal eingegeben wird, und authentifiziert den Benutzer.

Der Kerberos-Administrator muss außerdem sicherstellen, dass Kerberos-Computer physisch sicher sind und auch die Sicherung der Master-Datenbank beibehalten sollten.

Ein größeres Kerberos-Bild

In diesem Abschnitt wird beschrieben, wie Kerberos in die Athena-Umgebung passt, einschließlich der Verwendung durch andere Netzwerkdienste und -anwendungen, und wie es mit Kerberos-Fernrealms interagiert. Eine ausführlichere Beschreibung der Umgebung von Athena finden Sie unter G.W. Bäume.

Nutzung von Kerberos durch andere Netzwerkdienste

Mehrere Netzwerkanwendungen wurden für die Verwendung von Kerberos modifiziert. Die Befehle rlogin und rsh versuchen zunächst, sich mithilfe von Kerberos zu authentifizieren. Ein Benutzer mit gültigen Kerberos-Tickets kann sich auf einem anderen Athena-Rechner anmelden, ohne dass Rhosts-Dateien eingerichtet werden müssen. Wenn die Kerberos-Authentifizierung fehlschlägt, greifen die Programme auf ihre üblichen Autorisierungsmethoden zurück, in diesem Fall auf die Rhosts-Dateien.

Wir haben das Post Office Protocol dahingehend modifiziert, dass es Kerberos für die Authentifizierung von Benutzern verwendet, die ihre E-Mail von der "Post Office" abrufen möchten. Ein Nachrichtenübermittlungsprogramm namens Zephyr wurde kürzlich in Athena entwickelt und verwendet Kerberos auch für die Authentifizierung.

Das Programm zum Signieren neuer Benutzer, das so genannte Register, verwendet sowohl das Service Management System (SMS) als auch Kerberos. Über SMS wird bestimmt, ob die vom

potenziellen neuen Athena-Benutzer eingegebenen Informationen, wie Name und MIT-Identifikationsnummer, gültig sind. Anschließend überprüft er mit Kerberos, ob der angeforderte Benutzername eindeutig ist. Wenn alles gut geht, wird ein neuer Eintrag in die Kerberos-Datenbank gemacht, der den Benutzernamen und das Kennwort enthält.

Eine ausführliche Diskussion über die Verwendung von Kerberos zur Sicherung des Netzwerkdateisystems von Sun finden Sie im Anhang.

Interaktion mit anderen KerberiInnen

Es wird erwartet, dass verschiedene Verwaltungsorganisationen Kerberos für die Benutzerauthentifizierung verwenden möchten. Es wird auch erwartet, dass Benutzer in einem Unternehmen in vielen Fällen Services in einem anderen Unternehmen nutzen möchten. Kerberos unterstützt mehrere administrative Domänen. Die Spezifikation von Namen in Kerberos enthält ein Feld, das als Bereich bezeichnet wird. Dieses Feld enthält den Namen der administrativen Domäne, in der der Benutzer authentifiziert werden soll.

Services werden in der Regel in einem Bereich registriert und akzeptieren nur die von einem Authentifizierungsserver für diesen Bereich ausgegebenen Anmeldeinformationen. Ein Benutzer ist normalerweise in einem Bereich (dem lokalen Bereich) registriert, aber es ist ihm möglich, von einem anderen Bereich (dem Remote-Bereich) ausgegebene Anmeldeinformationen zu erhalten, indem er die vom lokalen Bereich bereitgestellte Authentifizierung unterstützt. Anmeldeinformationen, die in einem Remote-Bereich gültig sind, geben den Bereich an, in dem der Benutzer ursprünglich authentifiziert wurde. Die Dienste im Remote-Bereich können entscheiden, ob sie diese Anmeldeinformationen einhalten, abhängig von der erforderlichen Sicherheit und der Vertrauensebene in dem Bereich, der den Benutzer ursprünglich authentifiziert hat.

Um eine realitätsübergreifende Authentifizierung durchzuführen, müssen die Administratoren der einzelnen Realms einen Schlüssel auswählen, der von ihren Realms gemeinsam genutzt werden kann. Ein Benutzer im lokalen Bereich kann dann vom lokalen Authentifizierungsserver ein Ticket zur Ticketgewährung für den Server zur Ticketvergabe im Remote-Bereich anfordern. Wenn dieses Ticket verwendet wird, erkennt der Server, der das Ticket per Fernzugriff erteilt, dass die Anfrage nicht von seinem eigenen Bereich stammt, und verwendet den zuvor ausgetauschten Schlüssel, um das Ticket zu entschlüsseln. Anschließend wird wie gewohnt ein Ticket ausgegeben, mit der Ausnahme, dass das Bereichsfeld für den Client den Namen des Bereichs enthält, in dem der Client ursprünglich authentifiziert wurde.

Dieser Ansatz könnte erweitert werden, um es einem zu ermöglichen, sich über eine Reihe von Realms zu authentifizieren, bis der gewünschte Dienst in den Bereich gelangt. Um dies zu erreichen, muss jedoch der gesamte Pfad aufgezeichnet werden, der verwendet wurde, und nicht nur der Name des ursprünglichen Bereichs, in dem der Benutzer authentifiziert wurde. In einer solchen Situation ist dem Server nur bekannt, dass A sagt, dass C sagt, dass der Benutzer so und so ist. Diese Aussage kann nur dann vertrauenswürdig sein, wenn auch jeder auf dem Pfad vertraut ist.

Kerberos-Probleme und offene Probleme

Es gibt eine Reihe von Problemen und offenen Problemen im Zusammenhang mit dem Kerberos-Authentifizierungsmechanismus. Unter anderem geht es darum, die korrekte Lebensdauer eines Tickets zu bestimmen, wie Proxys zugelassen werden und wie die Integrität der Workstations

gewährleistet werden kann.

Das Problem der Lebensdauer eines Tickets besteht darin, den richtigen Kompromiss zwischen Sicherheit und Komfort zu wählen. Wenn die Lebensdauer eines Tickets lang ist, können Tickets und der zugehörige Sitzungsschlüssel für einen längeren Zeitraum gestohlen oder verlegt werden. Diese Informationen können gestohlen werden, wenn ein Benutzer vergessen hat, sich von einer öffentlichen Workstation abzumelden. Wenn ein Benutzer auf einem System authentifiziert wurde, das mehrere Benutzer zulässt, kann ein anderer Benutzer mit Zugriff auf das Stammverzeichnis möglicherweise die Informationen finden, die für die Verwendung gestohlener Tickets erforderlich sind. Das Problem bei einer kurzen Lebensdauer eines Tickets besteht jedoch darin, dass der Benutzer nach Ablauf dieses Tickets ein neues Ticket erwerben muss, bei dem der Benutzer das Kennwort erneut eingeben muss.

Ein offenes Problem ist das Proxy-Problem. Wie kann ein authentifizierter Benutzer einem Server den Erwerb anderer Netzwerkservices in seinem Namen ermöglichen? Ein Beispiel, in dem dies wichtig wäre, ist die Verwendung eines Diensts, der Zugriff auf geschützte Dateien direkt von einem Dateiserver erhält. Ein weiteres Beispiel für dieses Problem ist die so genannte Authentifizierungsweiterleitung. Wenn ein Benutzer bei einer Workstation angemeldet ist und sich bei einem Remotehost anmeldet, wäre es hilfreich, wenn der Benutzer auf dieselben lokal verfügbaren Dienste zugreifen könnte, während er ein Programm auf dem Remotehost ausführt. Dies wird dadurch erschwert, dass der Benutzer dem Remote-Host möglicherweise nicht traut, sodass eine Weiterleitung der Authentifizierung in allen Fällen nicht wünschenswert ist. Wir haben derzeit keine Lösung für dieses Problem.

Ein weiteres Problem, das in der Umgebung von Athena wichtig ist, ist, wie die Integrität der Software, die auf einer Workstation ausgeführt wird, gewährleistet werden kann. Dies ist nicht so sehr ein Problem auf privaten Workstations, da der Benutzer, der sie verwenden wird, die Kontrolle darüber hat. Auf öffentlichen Workstations ist es jedoch möglich, dass jemand das Anmeldeprogramm geändert hat, um das Kennwort des Benutzers zu speichern. Die einzige Lösung, die derzeit in unserer Umgebung zur Verfügung steht, ist es Menschen zu erschweren, Software zu modifizieren, die auf öffentlichen Workstations ausgeführt wird. Eine bessere Lösung würde erfordern, dass der Schlüssel des Benutzers niemals ein System verlässt, von dem der Benutzer weiß, dass es vertrauenswürdig ist. Dies ist beispielsweise möglich, wenn der Benutzer über eine Smartcard verfügt, die die im Authentifizierungsprotokoll erforderlichen Verschlüsselungen durchführen kann.

Kerberos-Status

Eine Prototypversion von Kerberos wurde im September 1986 in Betrieb genommen. Seit Januar 1987 ist Kerberos das einzige Mittel von Project Athena, um seine 5.000 Benutzer, 650 Workstations und 65 Server zu authentifizieren. Außerdem wird Kerberos nun anstelle von Rhosts-Dateien für die Zugriffskontrolle in mehreren Timesharing-Systemen von Athena verwendet.

Kerberos-Bestätigungen

Kerberos wurde ursprünglich von Steve Miller und Clifford Neuman mit Vorschlägen von Jeff Schiller und Jerry Saltzer entworfen. Seitdem sind zahlreiche andere Personen an dem Projekt beteiligt. darunter Jim Aspnes, Bob Baldwin, John Barba, Richard Basch, Jim Bloom, Bill Bryant, Mark Colan, Rob French, Dan Geer, John Kohl, John Kubiatowicz, Bob Mckie, Brian Murphy, John Ostlund Ken Raeburn, Chris Reed, Jon Rochlis, Mike Shanzer Bill. T'so, Win Treese und Stan

Zanarotti.

Wir sind Dan Geer, Kathy Lieben, Josh Lubarr, Ken Raeburn, Jerry Saltzer, Ed Steiner, Robbert van Renesse und Win Treese dankbar, deren Vorschläge frühere Entwürfe dieses Whitepapers deutlich verbessert haben.

Jedlinsky, J.T. Kohl und W.E. Sommerfeld, "The Zephyr Notification System", in Usenix Conference Proceedings (Winter, 1988).

Mittlere Rosenstein, D.E. Geer und P.J. Levine, in Usenix Conference Proceedings (Winter 1988).

R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh und B. Lyon, "Design and Implementation of the Sun Network Filesystem", in Usenix Conference Proceedings (Sommer 1985).

Anhang: Kerberos-Anwendung auf das Network File System (NFS) von SUN

Eine Schlüsselkomponente des Workstation-Systems von Project Athena ist die Verflechtung des Netzwerks zwischen der Workstation des Benutzers und seinem privaten Dateispeicher (Heimverzeichnis). Der gesamte private Speicher befindet sich auf einer Reihe von Computern (derzeit VAX 11/750), die für diesen Zweck vorgesehen sind. Auf diese Weise können wir Dienste auf öffentlich zugänglichen UNIX-Workstations anbieten. Wenn sich ein Benutzer bei einer dieser öffentlich zugänglichen Workstations anmeldet, anstatt seinen Namen und sein Kennwort anhand einer lokal installierten Kennwortdatei zu überprüfen, verwenden wir Kerberos, um die Authentizität des Benutzers zu bestimmen. Das Anmeldeprogramm fordert Sie zur Eingabe eines Benutzernamens auf (wie bei jedem UNIX-System). Dieser Benutzername wird verwendet, um ein Kerberos Ticket zu erhalten. Das Anmeldeprogramm verwendet das Kennwort, um einen DES-Schlüssel zur Entschlüsselung des Tickets zu generieren. Wenn die Entschlüsselung erfolgreich ist, befindet sich das Hauptverzeichnis des Benutzers, indem Sie den Hesiod-Benennungsdienst konsultieren und über NFS bereitgestellt werden. Das Anmeldeprogramm schaltet dann die Steuerung auf die Shell des Benutzers um, die dann die traditionellen benutzerspezifischen Anpassungsdateien ausführen kann, da das Hauptverzeichnis nun an die Workstation "angeschlossen" ist. Der Hesiod-Dienst wird auch verwendet, um einen Eintrag in der lokalen Kennwortdatei zu erstellen. (Dies ist zum Vorteil von Programmen, die Informationen in /etc/passwd nachschlagen.)

Aus mehreren Optionen für die Bereitstellung von Remote-Dateidiensten haben wir Sun's Network File System gewählt. Dieses System lässt sich jedoch nicht unbedingt mit unseren Bedürfnissen verknüpfen. NFS geht davon aus, dass alle Workstations in zwei Kategorien unterteilt werden (aus Sicht eines Dateiservers): vertrauenswürdig und nicht vertrauenswürdig. Nicht vertrauenswürdige Systeme können überhaupt nicht auf Dateien zugreifen. Vertrauenswürdige Systeme können dies tun. Vertrauenswürdige Systeme sind absolut vertrauenswürdig. Es wird davon ausgegangen, dass ein vertrauenswürdiges System durch freundliches Management verwaltet wird. Insbesondere ist es von einer vertrauenswürdigen Workstation möglich, sich als ein gültiger Benutzer des Dateidienstsystems zu tarnen und so Zugriff auf nahezu jede Datei auf dem System zu erhalten. (Nur Dateien im Besitz von "root" sind ausgenommen.)

In unserer Umgebung befindet sich die Verwaltung einer Workstation (im traditionellen Sinne des UNIX-Systemmanagements) in den Händen des Benutzers, der sie derzeit verwendet. Wir machen kein Geheimnis des Root-Passworts auf unseren Workstations, denn wir wissen, dass ein wirklich unfreundlicher Benutzer einbrechen kann, weil er/er am selben physischen Ort wie der

Rechner sitzt und Zugriff auf alle Konsolenfunktionen hat. Daher können wir unseren Workstations bei der NFS-Interpretation von Vertrauen nicht wirklich vertrauen. Um eine angemessene Zugriffskontrolle in unserer Umgebung zu ermöglichen, mussten wir einige Änderungen an der NFS-Basissoftware vornehmen und Kerberos in das Schema integrieren.

Kerberos, nicht modifiziert, NFS

Bei der Implementierung von NFS, mit der wir begonnen haben (von der University of Wisconsin), wurde die Authentifizierung in Form eines Datenpakets durchgeführt, das in jeder NFS-Anfrage enthalten ist (in der NFS-Terminologie als "Anmeldeinformationen" bezeichnet). Diese Anmeldeinformationen enthalten Informationen über die eindeutige Benutzer-ID (UID) des Antragstellers und eine Liste der Gruppen-IDs (GIDs) der Mitgliedschaft des Antragstellers. Diese Informationen werden dann vom NFS-Server für die Zugriffsprüfung verwendet. Der Unterschied zwischen einer vertrauenswürdigen und einer nicht vertrauenswürdigen Workstation besteht darin, ob ihre Anmeldeinformationen vom NFS-Server akzeptiert werden.

Kerberos hat NFS geändert

In unserer Umgebung müssen NFS-Server Anmeldeinformationen von einer Workstation nur dann akzeptieren, wenn die Anmeldeinformationen die UID des Benutzers der Workstation angeben, und keine andere.

Eine naheliegende Lösung wäre, die Art der Anmeldeinformationen von bloßen Indikatoren für UID und GIDs auf vollständig gebrannte, von Kerberos authentifizierte Daten zu ändern. Bei der Einführung dieser Lösung würde jedoch ein erhebliches Leistungseinbußen hingenommen. Anmeldeinformationen werden bei jedem NFS-Vorgang, einschließlich aller Lese- und Schreibaktivitäten der Festplatte, ausgetauscht. Eine Kerberos-Authentifizierung bei jeder Datenträgertransaktion würde eine ganze Reihe von vollverschlüsselten (softwarebasierten) Verschlüsselungen pro Transaktion hinzufügen und, wie wir aus Umschlagberechnungen hervorgehen, unannehmbare Leistung erbracht haben. (Es hätte auch erfordert, die Kerberos-Bibliotheksroutinen im Kernel-Adressbereich zu platzieren.)

Wir benötigten einen Hybridansatz, wie im Folgenden beschrieben. Die Grundidee besteht darin, die Anmeldeinformationen der NFS-Serverzuordnung von Client-Workstations an gültige (und möglicherweise unterschiedliche) Anmeldeinformationen des Serversystems zu übertragen. Diese Zuordnung wird im Kernel des Servers für jede NFS-Transaktion durchgeführt und wird zur "Mount"-Zeit durch einen Prozess auf Benutzerebene eingerichtet, der eine von Kerberos moderierte Authentifizierung durchführt, bevor eine gültige Kernel-Credential-Zuordnung erstellt wird.

Um dies zu implementieren, haben wir einen neuen Systemaufruf zum Kernel hinzugefügt (nur auf Serversystemen, nicht auf Client-Systemen erforderlich), der die Steuerung der Zuordnungsfunktion vorsieht, die eingehende Anmeldeinformationen von Client-Workstations zu Anmeldeinformationen zuordnet, die für den Server gültig sind (falls vorhanden). Die grundlegende Zuordnungsfunktion ordnet den Tupel zu:

```
<CLIENT-IP-ADDRESS, UID-ON-CLIENT>
```

auf eine gültige NFS-Anmeldeinformationen für das Serversystem. Die CLIENT-IP-ADDRESS wird aus dem vom Client-System bereitgestellten NFS-Anforderungspaket extrahiert. Hinweis: alle Informationen in den vom Client generierten Anmeldeinformationen außer UID-ON-CLIENT

werden verworfen.

Wenn keine Zuordnung vorhanden ist, reagiert der Server auf zwei Arten, je nachdem, ob er konfiguriert wurde. In unserer benutzerfreundlichen Konfiguration werden die unmappbaren Anfragen in die Anmeldeinformationen für den Benutzer "none" übernommen, der keinen privilegierten Zugriff hat und über eine eindeutige UID verfügt. Unfreundliche Server geben einen NFS-Zugriffsfehler zurück, wenn keine gültige Zuordnung für eingehende NFS-Anmeldeinformationen gefunden werden kann.

Unser neuer Systemaufruf wird verwendet, um Einträge aus der Kernelübersicht hinzuzufügen und zu löschen. Sie bietet außerdem die Möglichkeit, alle Einträge zu leeren, die einer bestimmten UID auf dem Serversystem zugeordnet sind, oder alle Einträge einer bestimmten CLIENT-IP-ADDRESS zu leeren.

Wir haben den mount-Daemon (der NFS-Bereitstellungsanfragen auf Serversystemen behandelt) so geändert, dass er einen neuen Transaktionstyp akzeptiert, die Kerberos-Authentifizierungsanfrage. Im Prinzip stellt das Client-System im Rahmen des Montageprozesses einen Kerberos-Authentifizierer sowie einen Hinweis auf ihren UID-ON-CLIENT (im Kerberos-Authentifizierer verschlüsselt) auf der Workstation bereit. Der Mount-Daemon des Servers konvertiert den Kerberos-Hauptnamen in einen lokalen Benutzernamen. Dieser Benutzername wird dann in einer speziellen Datei nachgeschlagen, um die UID- und GID-Liste des Benutzers zu erhalten. Aus Effizienzgründen ist diese Datei eine Datenbankdatei mit dem Benutzernamen als Schlüssel. Aus diesen Informationen wird eine NFS-Anmeldeinformationen erstellt und als gültige Zuordnung des <CLIENT-IP-ADDRESS, CLIENT-UID>-Tupels für diese Anforderung an den Kernel übergeben.

Zur unmount-Zeit wird eine Anforderung an den mount-Daemon gesendet, die zuvor hinzugefügte Zuordnung aus dem Kernel zu entfernen. Es ist auch möglich, eine Anforderung zur Abmeldezeit zu senden, um alle Zuordnungen für den aktuellen Benutzer auf dem betreffenden Server zu ungültig zu machen, und so alle verbleibenden Zuordnungen, die vorhanden sind (obwohl dies nicht der Fall sein sollte), zu bereinigen, bevor die Workstation für den nächsten Benutzer verfügbar gemacht wird.

Kerberos Security Implikationen des modifizierten NFS

Diese Implementierung ist nicht vollständig sicher. Zunächst einmal werden Benutzerdaten immer noch in unverschlüsselter und somit abnehmbarer Form über das Netzwerk gesendet. Die grundlegende, transaktionsbasierte Authentifizierung basiert auf einem <CLIENT-IP-ADDRESS, CLIENT-UID> Paar, das im Anforderungspaket unverschlüsselt bereitgestellt wird. Diese Informationen könnten gefälscht und somit die Sicherheit gefährdet werden. Es ist jedoch zu beachten, dass nur dann gültige Zuordnungen vorhanden sind, wenn ein Benutzer seine Dateien aktiv nutzt (d. h. wenn er angemeldet ist), und dass diese Form des Angriffs daher auf den Zeitpunkt beschränkt ist, an dem der betreffende Benutzer angemeldet ist. Wenn ein Benutzer nicht angemeldet ist, erlaubt die Fälschung von IP-Adressen keinen unbefugten Zugriff auf seine Dateien.

Kerberos-Referenzen

1. S.P. Miller, B.C. Neuman, J.I. Schiller und J.H. Saltzer, Abschnitt E.2.1: Kerberos Authentication and Authorization System, M.I.T. Projekt Athena, Cambridge, Massachusetts (21. Dezember 1987).

2. E. Balkovich, S.R. Lerman und R.P. Parmelee "Computing in der Hochschulbildung: The Athena Experience", Communications of the ACM, Vol. 28(11), S. 1214-1224, ACM (November 1985).
3. Mittlere Needham und M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", Communications of the ACM, Band 21(12), 993-99 (Dezember 1978).
4. V.L. Voydock und S.T. Kent, "Security Mechanismen in High-Level Network Protocols", Computing Surveys, Band 15(2), ACM (Juni 1983).
5. National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standards Publication 46, Government Printing Office, Washington, DC (1977).
6. SP Dyer, "Hesiod", in Usenix Conference Proceedings (Winter, 1988).
7. W.J. Bryant, Tutorial von Kerberos Programmierer, MIT Project Athena (In Vorbereitung).
8. W.J. Bryant, Kerberos Administrator's Manual, MIT Project Athena (In Vorbereitung).
9. G.W. Treese, "Berkeley Unix on 1000 Workstations: Athena wechselt zu 4.3BSD", in Usenix Conference Proceedings (Winter, 1988).
10. C.A. DellaFera, M.W. Eichin, R.S. Französisch, D.C. Jedlinsky, J.T. Kohl und W.E. Sommerfeld, "The Zephyr Notification System", in Usenix Conference Proceedings (Winter, 1988).
11. Mittlere Rosenstein, D.E. Geer und P.J. Levine, in Usenix Conference Proceedings (Winter 1988).
12. R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh und B. Lyon, "Design and Implementation of the Sun Network Filesystem", in Usenix Conference Proceedings (Sommer 1985).

Zugehörige Informationen

- Support-Seite für Kerberos
- Technischer Support und Dokumentation - Cisco Systems