P2P-Sicherheit

Georg Lukas
Seminar „Kommunikation in P2P-Netzen”
Universität Magdeburg

2004-03-14

Zusammenfassung

Dieses Dokument soll einen Überblick darüber liefern, welche Aspekte der Sicherheit beim Design und der Implementierung von Peer-to-Peer-Systemen relevant sind. Nach einer Vorstellung der Sicherheitsziele wird auf die möglichen Störungen und Störer eingegangen, um dann mit den Verfahren abzuschließen, die solche Störungen möglichst effektiv abwehren sollen. Dabei liegt der Schwerpunkt auf P2P-Systemen zur verteilten Datenspeicherung, mit einem kurzen Ausflug zum verteilten Rechnen. Es wird dagegen nicht darauf eingegangen, wie man sich als Benutzer bestehender P2P-Systeme vor Bugs und Angriffen schützt.

Inhaltsverzeichnis

1 Sicherheit

Der deutsche Begriff *Sicherheit* umfasst zwei Aspekte, die im Englischen besser mit *safety*, der Sicherheit vor Fehlern und Ausfällen, und *security*, also Sicherheit vor Angriffen beschrieben werden. Dabei ist *safety* eine unbedingte Voraussetzung für *security*, da ein Programm mit Bugs sich viel leichter angreifen lässt.

1.1 Sicherheitsziele

Ein Sicherheitskonzept muss die folgenden Ziele berücksichtigen, um erfolgreich zu sein:

- *Integrität* der Daten
- *Verfügbarkeit* von Daten und Diensten und deren *Verlässlichkeit*
- *Vertraulichkeit* von Kommunikation und Daten
- *Authentizität* der Kommunikationspartner oder von Dokumenten
- *Anonymität* von Teilnehmern

Allerdings ist die Umsetzung aller Ziele in vollem Maße nich möglich, da es zwangsweise zu Konflikten kommt. In einem P2P-Netz, dessen Knoten vollständig anonym sind, können zwischen den einzelnen Teilnehmern keine Vertrauensbeziehungen hergestellt werden – eine Überprüfung der Authentizität anderer Teilnehmer ist unmöglich.

Ein Netz, in dem es unmöglich ist, die Speicherposition der einzelnen Datenelemente zu bestimmen, macht es den Nutzern gleichzeitig sehr schwer, von ihnen gesuchte Objekte zu finden. Daher ist es notwendig, beim Design eines sicheren P2P-Systems die einzelnen Ziele gegeneinander abzuwägen und sich für den besten Kompromiss zu entscheiden.

1.2 Anonymität

Der Begriff *Anonymität* beschreibt die „Unmöglichkeit, aus einer Menge von Teilnehmern identifiziert zu werden"[5]. Man unterscheidet zwischen unterschiedlichen Arten der Anonymität, die jeweils unterschiedliche Teilnehmer-Rollen abdecken, so z.B. Autor-, Herausgeber- und Leser-Anonymität. Außerdem existiert Server-Anonymität, bei der man nicht nachvollziehen kann, auf welchem Server im Netz bestimmte Dokumente gelagert werden, und umgekehrt dazu Dokument-Anonymität, wo nicht feststellbar ist, welche Dokumente sich auf einem bestimmten Server befinden.

2 Störungen und Störer

Bei der Entwicklung eines sicheren P2P-Systems spielen die möglichen Bedrohungen eine große Rolle. Je nach Ausrichtung muss das Konzept von simplen Ausfällen bis zu Unterbindungsversuchen durch Großkonzerne oder Regierungen widerstehen können. Im folgenden werden Klassen von potentiellen Störern und ihre möglichen Angriffsziele bei einem P2P-Netz vorgestellt, gefolgt von den möglichen Angriffsebenen.

2.1 Störer-Klassen

Die einfachste und häufigste Art sind die *ungezielten Störungen*. Dazu zählen der Hardwareausfall eines Knotens oder die Unterbrechung einer Netzverbindung genau so wie Vandalismus oder Angriffe auf ein anderes System, die Kollateralschäden hinterlassen.

Zu den *laienhaften Angreifern* zählen Personen oder Gruppen, die zwar auf das P2P-System ausgerichtet Angriffe durchführen, aber nur über begrenzte Ressourcen verfügen. Sie können zum Teil umfangreiche Kenntnisse des Systems besitzen, sind aber nur in der Lage, punktuell Schaden azurichten.

Im Kontrast dazu stehen *professionelle Angreifer* wie große Konzerne oder Sicherheitsdienste. Diese können einen großen Etat und viele Ressourcen bündeln, um das Netz möglichst effektiv zu stören.

Staatliche *Behörden* bilden eine weitere Klasse. Diese können durch Gesetze, Verordnungen und polizeiliche Maßnahmen agieren und so den Austausch bestimmter Materialien oder die Verwendung von Diensten sperren. Auf diese Weise geht z.B. die chinesische Regierung gegen Freiheitsaktivisten vor.

2.2 Angriffsziele

Angriffe auf ein P2P-Netz dienen meist nicht bloß dazu, den Netzbetrieb zu stören, sondern sollen dem Angreifer Informationen beschaffen, oder Netzteilnehmer am Erhalt von Informationen hindern.

Am weitesten verbreitet sind Angriffe, die den Zugang zu Daten unterbinden sollen. Dabei geht es entweder darum, einer bestimmten Benutzergruppe (eines Providers oder in einem Land) den Zugang zu allen Daten des Netzes zu sperren, oder um die Unterbrechung des Zugangs aller Teilnehmer zu bestimmten Dokumenten, die ungewollte oder verbotene Inhalte haben.

Einen Schritt weiter gehen Angriffe, die dazu dienen, Daten komplett aus dem Netz zu entfernen - dazu muss der Angreifer meist zuerst herausfin-

den, welche Teilnehmer diese Daten vorhalten, und kann danach gegen sie vorgehen.

Subtiler ist die Manipulation von im Netz gespeicherten Daten. Dabei wird der Anschein erweckt, das ursprüngliche Dokument befände sich noch im Netz, es wird aber eine andere, abgeänderte Version verbreitet.

Manchmal genügt es dem Angreifer aber auch, bestimmte Informationen aus dem Netz zu bekommen – entweder Dokumente, zu denen er keine Autorisierung hat, oder Meta-Information über den Netzbetrieb – welche Knoten bestimmte Daten vorhalten oder anfordern. Gegen diese Knoten kann dann am P2P-Netz vorbei vorgegangen werden - so wie es die US-amerikanische – und neuerdings auch die deutsche – Musikindustrie mit ihren Klagen gegen Tauschbörsen-Benutzer demonstriert.

2.3 Angriffsebenen

Ein verteiltes Peer-to-Peer-Netz kann auf mehreren Ebenen angegriffen werden. Auf der *physikalischen Ebene* gehört dazu das Zerstören von Rechnern oder Netzwerkverbindungen, aber auch das Entwenden oder Beschlagnahmen von Datenträgern.

Auf der *Protokoll-Ebene* setzen Angriffe an, die die Kommunikation der Netzknoten miteinander stören. Dazu gehören Man-in-the-Middle-Attacken, wo der Angreifer sich in die Kommunikation zweier Stationen einklinkt und so nicht nur alles mitlesen, sondern auch gezielt Daten verfälschen und unterschlagen kann. Die meisten Protokoll-Angriffe sind aber speziell auf ein bestimmtes Netz und das ihm zugrundeliegende Protokoll ausgerichtet, als Beispiel sei GNUnet zu nennen, wo man einen Knoten durch Überlastung dazu bringen kann, die Adresse des nächsten Knotens auf dem Weg zu einem bestimmten Dokument preiszugeben. So ist es möglich, sich bis zu dem Teilnehmer vorzuarbeiten, auf dessen System sich bestimmte Daten befinden, und die Server-Anonymität zu unterwandern.

Schließlich sind auch Angriffe gegen die *Anwendung* selbst möglich. Dazu zählt die Ausnutzung von Sicherheitslücken (z.B. Buffer-Overflows) genau so wie das Fluten des Netzwerks mit sinnloses Suchanfragen oder Dokumenten.

3 Sicherheitskonzepte

Um ein P2P-Netz gegen Störungen und Angriffe abzusichern, existieren zahlreiche Maßnahmen, die im folgenden gegliedert nach ihrem primären Sicherheitsziel dargestellt werden.

3.1 Integrität

Die Datenintegrität wird bei Speicherung und Transport im allgemeinen mit kryptographischen Checksummenverfahren wie MD5 oder SHA-1 gesichert. Dabei werden aus Dateien beliebiger Länge relativ kurze (128 oder 160 Bit) eindeutige Hashwerte berechnet, und mit übertragen. Diese Hashwerte lassen sich auch als Index für die Dateien benutzen, aus denen sie ermittelt wurden, und werden z.B. im Freenet für Suche und Sortierung verwendet[6].

Im Idealfall sollte jede Station, die eine Datei anbietet, weitergibt oder empfängt, die Übereinstimmung der Daten mit der Checksumme überprüfen, um Fehler und Manipulationen so schnell wie möglich zu erkennen. Bei hoher Auslastung des Netzes sollte es aber ausreichen, wenn der Client beim Empfang der Daten diese Überprüfung vornimmt, und bei Abweichungen die Daten z.B. von einem anderen Anbieter anfordert.

Signiert der Autor die Daten (bzw. deren Checksumme) mit einem asymmetrischen Verschlüsselungsverfahren, so kann man (Vertrauen in seinen Public Key vorausgesetzt) sicher sein, dass keine Manipulationen vorgenommen wurden. Außerdem kann man in die Signatur Metadaten einbinden, die sich für die Suche nach dem Dokument verwenden lassen. Umgekehrt ist die Anzahl der Suchtreffer mit der gleichen Checksumme ein Indiz für die Qualität eines Dokuments, da davon auszugehen ist, dass hochwertige Inhalte eher verbreitet und gespeichert werden.

3.2 Verfügbarkeit

Die Verfügbarkeit des Systems lässt sich auf den unterschiedlichen Ebenen unterschiedlich realisieren. Für die *physikalische Sicherheit* sorgt die verteilte Speicherung vieler Kopien von wichtigen Dokumenten. Da sich zentrale Elemente in einem P2P-Netz besonders effizient angreifen lassen, sollte das Konzept dezentral aufgebaut sein.

Analog dazu sollte das Netz auch die Speicherposition von Dokumenten verschleiern, damit ein Angreifer nicht die Speicherer ausschalten kann - hier geht Verfügbarkeit mit der Anonymität Hand in Hand. Allerdings bringt das auch einen Nachteil mit sich, da es die Suche nach Dokumenten im Netz erschwert.

Auf der *Protokollebene* lässt sich die Verfügbarkeit dadurch steigern, dass man gleichzeitige Netzwerkverbindungen zu mehreren Knoten aufrechterhält, wobei diese Knoten möglichst in unterschiedlichen Regionen des Internets stehen sollten. Dadurch wird nicht nur der Ausfall einzelner Verbindungen toleriert, es wird auch möglich, eine parallele Breitensuche durchzuführen, indem man die Anfrage an alle verbundenen Knoten schickt.

Auch sollte das verwendete Netzwerkprotokoll nicht von einem Unbeteiligten erkannt werden können, was durch eine Verschlüsselung aller Verbinungen per TLS und durch die Verwendung zufälliger Portnummern erreichbar ist. Ansonsten wäre es möglich, die Verwendung des Protokolls – z.B. providerseitig – zu sperren.

Bei der Entwicklung der *Anwendung* sollte nicht nur auf einen robusten und fehlertoleranten Umgang mit von Außen empfangenen Daten geachtet werden, sondern auch auf eine faire Verteilung der Ressourcen gegenüber anderen Teilnehmern. Dabei kann ein lokaler oder globaler Reputationsmechanismus dazu dienen, die Priorität fremder Anfragen zu bestimmen und diese entsprechend abzuarbeiten.

3.3 Vertraulichkeit

Um die Vertraulichkeit von Daten zu gewährleisten, sollten diese nur verschlüsselt gespeichert und übertragen werden. Dabei ist es denkbar, dass der Schlüssel, mit dem sie wieder entschlüsselt werden können, sich außerhalb des P2P-Systems befindet, und so kein Teilnehmer in der Lage ist, die Daten einzusehen, oder dass nur eine bestimmte Benutzergruppe das Dokument entschlüsseln kann. Wird das Dokument dagegen zur Publikation im Netz verbreitet, es soll aber nicht in irgendwessen Besitz nachgewiesen werden können, kann man den Schlüssel des Dokuments unabhängig vom Dokument selbst im Netz verteilen, so dass erst aus der Kombination ersichtlich wird, um welche Daten es sich handelt.

Um nicht nur den Inhalt sondern auch die Existenz der Kommunikation zu verstecken, kann man neben Steganografie auch Pakete mit zufälligem Rauschen übertragen. Je nachdem wie viele Verbindungen der potentielle Angreifer überwachen kann, müssen entsprechend viele Rauschpakete eingestreut werden. Für weitere Verwirrung sorgen Änderungen der Route zwischen zwei aufeinander folgenden Paketen.

3.4 Authentizität

Die Authentizität der Kommunikationspartner ist wichtig, wenn es darum geht, ihre Seriösität einzuschätzen, allerdings steht sie im Widerspruch zur

Anonymität – man kann sich nicht auf die Identität eines Unbekannten verlassen. Um das Dilemma aufzulösen kann man den Benutzern erlauben, ein oder mehrere Pseudonyme anzulegen, so dass andere ihr Vertrauen an diese binden können. Wird als Grundlage dafür ein asymmetrisches Schlüsselpaar verwendet, kann der Besitzer seine Pseudonym-Identität jederzeit mit geringem Aufwand nachweisen. Andere können dafür den Public-Key um Reputationswerte erweitern und auf diese Weise ein Web of Trust aufbauen. Sollte jedoch ein Benutzer zu viel negative Reputation bekommen, kann er das System durch das Neuanlegen einer Identität umgehen.

Dieses Problem lässt sich nur durch eine zentrale Benutzerverwaltung lösen, dabei muss man allerdings vollständig auf die Anonymität der Benutzer verzichten. Diese Lösung macht insbesondere bei kleinen Nutzergruppen Sinn, vor allem wenn bereits eine lokale Public Key Infrastructure (PKI) existiert (wie das in Unternehmen häufig der Fall ist).

3.5 Anonymität

Die meisten existierenden P2P-Systeme sichern Autor- und Herausgeber-Anonymität praktisch automatisch, da man aus einem auf einem System liegenden Dokument nicht sehen kann, ob es von diesem oder von einem anderen Peer eingestellt wurde.

Die Methoden, die initiale Verbreitung anonym zu halten, sind denen für die Anonymisierung von Suchanfragen und Antworten sehr ähnlich. Zum einen können Mix-Netze eingesetzt werden, d.h. ein Datenpaket wird über mehrere Stationen verschickt und für jede davon in einen verschlüsselten Umschlag gepackt, so dass jeder Teilnehmer auf dem Weg nur seinen Vorgänger und seinen Nachfolger sieht, und nicht weiss, wie viele Stationen jeweils noch dahinter stehen.

Zum anderen sollte strikt darauf geachtet werden, welche Routing-Informationen weitergegeben werden. Eine solche Weitergabe kann zwar die Übertragung von Daten deutlich beschleunigen, birgt aber die Gefahr, dass ein Angreifer daraus auf den tatsächlichen Speicherort oder den Suchenden schließen kann.

4 Verteiltes Rechnen

Das Ziel von verteiltem Rechnen besteht darin, die eigene CPU-Zeit anderen zur Verfügung zu stellen, wenn man sie nicht braucht, und dafür bei Bedarf die kombinierte Leistung vieler Systeme einfordern zu können.

4.1 Sicherheitsbetrachtung

Die Verifikation von Rechenergebnissen ist deutlich schiweriger als die Überprüfung der Checksummen von gespeicherten Dateien. Ein Angreifer, der seine Reputation steigern oder gratis Rechenleistung beziehen will, kann besonders viele Aufgaben zum Rechnen anfordern und sofort Scheinergebnisse zurückliefern. Eine Überprüfung hat die selbe Komplexität wie die ausgeteilte Aufgabe, was das Verteilen sinnlos machen würde. Natürlich kann man eine Aufgabe auch an zwei Teilnehmer austeilen, und danach die Ergebnisse vergleichen, aber auch das funktioniert nicht, wenn sich zwei Teilnehmer absprechen, um gemeinsam das System zu missbrauchen.

Stichprobenartige Untersuchung von einzelnen Teillösungen macht Sinn, wenn die Übertragung des gesamten Lösungsweg (oder der Lösungen von Teilproblemen) die Speicher- und Netzkapazitäten nicht zu stark belasten. Werden dagegen nur die für die Stichprobe ausgewählten Lösungen angefordert, kann der Betrüger diese nach der Anforderung berechnen und ausschließlich dafür die korrekten Ergebnisse liefern.

Wird ein Mißbrauch nicht erkannt, so werden entweder falsche Ergebnisse verwendet oder eine korrekte Lösung wird nicht gefunden. Beide Situationen können die gesamte Berechnung unbrauchbar machen und sollten unbedingt vermieden werden.

Wenn ein Mechanismus existiert, mit dem Betrug erkannt werden kann, sollte die Strafe (z.B. in Form einer Absenkung der Reputation) so hoch ausfallen, dass es für die Teilnehmer statistisch günstiger ist, die Berechnungen durchzuführen und korrekte Ergebnisse zu lierfern. Damit wird Manipulation für die, die die Leistung des Systems nutzen wollen, uninteressant und es bleiben nur noch Teilnehmer über, die das Netzwerk bewußt sabotieren.

4.2 Magic Numbers

Einen interessanten Ansatz gehen Philippe Golle und Ilya Mironov mit dem Konzept der „Magic Numbers"[7]. Das ist ein Lösungskonzept für die Bruteforce-Rückwärtssuche in Einwegfunktionen, wo man den Parameter einer gegebenen Funktion für einen bestimmten Ergebniswert sucht.

Es wird davon ausgegangen, dass jeder Teilnehmer einen bestimmten Suchbereich (der Teil des Definitionsbereichs der Funktion ist) darauf untersuchen muss, ob einer der dortigen Werte als Parameter der Funktion ein bestimmtes Ergebnis liefert. Dafür soll der Teilnehmer die Funktion mit jedem der Werte des Suchbereichs durchführen und die Ausgabe mit dem Zielergebnis vergleichen.

Die einfachste Form des Betrugs würde darin bestehen, dass man für jeden Suchbereich angibt, dass der gesuchte Parameter darin nicht vorkommt. Dagegen richten sich die „Magic Numbers", der Teilnehmer bekommt statt einem Ergebniswert eine bestimmte Menge von Ergebnissen, von denen ein (dem Rechnenden nicht bekannter) Teil aus dem gegebenen Suchbereich stammt.

Um nicht als Betrüger aufzufallen, muss der Teilnehmer eine korrekte Zuordnung der Ergebniswerte zum Suchbereich zurückliefern, wofür er zwingend den gesamten Bereich durchrechnen muss.

Dagegen lässt sich die Ergebniswerttabelle mit sehr geringem Aufwand erstellen, da man hier einfach die Einwegfunktion auf mehrere Zufallswerte aus dem Suchbereich sowie außerhalb davon anwenden muss.

5 Zusammenfassung

Beim Design und der Implementierung eines Peer-to-Peer-Systems gilt es zunächst, festzustellen, wie weit die Sicherheitsziele Integrität, Verfügbarkeit, Vertraulichkeit, Authentizität und Anonymität verfolgt werden sollen, um den Netzbetrieb sicherzustellen. Dazu gehört neben einer Abwägung der einzelnen Ziele gegeneinander auch eine Analyse der möglichen Angreifer und der von ihnen verfolgten Absichten, die von der Störung der Kommunikation bis zur strafrechtlichen Verfolgung von Netzteilnehmern reichen können.

Darauf aufbauend lässt sich dann festlegen, welche Maßnahmen eingesetzt werden sollen, um die Sicherheitsziele zu erreichen. Neben kryptographischer Verfahren zur Wahrung von Integrität, Vertraulichkeit und Authentizität können auch Reputationsmechanismen eingesetzt werden, um die Verfügbarkeit von Diensten zu verbessern. Zur Wahrung der Anonymität kommen Mix-Ketten und eingestreuter Dummy-Traffic in Betracht.

Schließlich lassen sich Magic Numbers einsetzen, um beim verteilten Durchrechnen von Einwegfunktionen die Integrität der Berechnungen mit geringem Aufwand zu schützen.

Literatur

[1] http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p10.html

[2] http://www.cl.cam.ac.uk/users/rja14/eternity/eternity.html

[3] http://www.freehaven.net/anonbib/cache/strong-eternity.pdf

[4] http://www.ovmj.org/GNUnet/download/aff.ps

[5] http://www.ovmj.org/GNUnet/papers/GNUnet_pet.pdf

[6] http://freenet.sourceforge.net/papers/freenet-ieee.pdf

[7] http://crypto.stanford.edu/ pgolle/papers/distr.pdf