CLOUD SPEICHERDIENSTE ALS ANGRIFFSVEKTOREN

Martin Mulazzani, Sebastian Schrittwieser, Manuel Leithner,
Markus Huber, Edgar Weippl

Kurzfassung

Während der letzten Jahre etablierte sich eine Vielzahl von Online File Storage Diensten. Während sich manche auf Basisfunktionalitäten beschränken bieten ausgereiftere Services auch Dienste wie gemeinsame Ordner („Shared Folder") für die Parallelnutzung durch mehrere Personen, die Möglichkeit zur Echtzeitzusammenarbeit, eine Minimierung des Datentransfers oder nahezu unbegrenzte Speicherkapazität.

Im Zuge dieses Artikels soll eine generelle Übersicht zu aktuellen Speicherservices im Internet geboten werden, sowie die Ergebnisse unserer Untersuchungen zu Dropbox als Beispiel einer populären Softwarelösung präsentiert werden. Schlussfolgernd aus unseren Resultaten konnten wir zeigen das Dropbox als Speicherort für urheberrechtlich geschützte Dateien aus Filesharingnetzwerken genutzt wird. Darüber hinaus ist es möglich über Dropbox unbemerkt unbegrenzt Daten zu speichern. Das Missbrauchspotential, so Daten zu verbergen definieren wir als Online Slack Space. *Abschließend zeigen wir mögliche Gegenmaßnahmen um unsere Angriffe zu verhindern, und wie sie im Falle von Dropbox konkret implementiert werden könnten.*

1 Einleitung

Viele innovative Speicherdienste wurden in den letzten Jahren vorgestellt um Dateien auf zentralen Servern oder auf mehreren Rechnern verteilt im Internet zu speichern. Die gegenwärtigen Speicherservices folgen größtenteils einem einfachen Konzept und bieten ihren Nutzern recht elementare Funktionen. Vom technischen Standpunkt aus betrachtet basieren die meisten dieser Dienste auf etablierten Protokollen wie FTP [28] oder WebDAV [22], einer Erweiterung des HTTP-Protokolls.

Mit Aufkommen des Cloud Computing und der gemeinsamen Nutzung von Ressourcen gewannen diese zentralisierten Speicherdienste an Relevanz und die Zahl der Nutzer stieg signifikant. Im Konzept des *Cloud Storage* können Festplattenkapazitäten oder Netzwerkbandbreite diese gemeinsam genutzten Ressourcen darstellen. Für die Betreiber dieser Services ist es naheliegend diese Daten nicht doppelt zu speichern, da es eine Vielzahl von identischen Dateien (wie z.B. Installationsdateien) gibt die von mehreren Nutzern verwendet werden1. Nachdem sowohl Datentransfer als auch Speicherbelegung für den Betreiber Kostenfaktoren darstellen, wird die Tendenz zu einer Einschränkung überflüssig verbrauchter Kapazität, wie sie z.B. durch Mehrfachübertragung ein und derselben Datei entsteht, verständlich. Dropbox ist einer der größte Online Storage Service der diesbezügliche Verfahren (z.B. Deduplikation) anwendet [24].

Aus der Sicherheitsperspektive erwachsen jedoch durch die gemeinsame Verwendung von Dateien gänzlich neue Herausforderungen. Die klare Abgrenzung von Nutzerdaten ist nicht im selben Maße durchführbar wie im klassischen Dateihosting und neue Methoden werden notwendig um sicherzustellen, dass in diesem Fundus gemeinsam genutzter Daten der Zugriff nur in autorisierter Form geschehen kann. Das ist unserer Auffassung nach die grundlegende Herausforderung für effiziente und sichere Cloud-basierte Speicherdienste. Themen wie unberechtigter Datenzugriff („Unauthorized Data Access“) oder ungewollter Informationsweitergabe („Information Leakage“) wurde in diesem Bereich aber noch kaum aufgearbeitet.

Unsere Vorschläge, um unautorisierten Zugriff auf Daten oder Information Leakage zuvorzukommen, sind nicht spezifisch für Dropbox, sondern sind für alle Online Speicherdienste von Relevanz die Deduplikation verwenden. Wir sind davon überzeugt, dass die Anzahl der Cloud-basierten Speicherdienste in näherer Zukunft enorm steigen wird.

Die wichtigen Punkte dieses Artikels sind:

- Die Funktionalität eines ausgereifteren Speicherdienstes im Detail darzustellen: Dropbox;
- Zu dokumentieren unter welchen Umständen unerlaubter Zugriff auf Daten möglich ist;
- Festzustellen ob Dropbox zur Speicherung urheberrechtlich geschützter Inhalte verwendet wird;
- *Online Slack Space* zu definieren und welche Probleme sich daraus für forensische Untersuchungen ergeben können
- Client- und serverseitige Gegenmaßnahmen zu formulieren, um die Sicherheitsrisiken für Nutzerdaten zu vermindern.

2 Hintergründe

In diesem Abschnitt werden die technischen Details und die von Dropbox umgesetzten Sicherheitsmaßnahmen behandelt. Im Rahmen dieses Artikels verwenden wir den Begriff „Cloud Computing“ wie definiert in [9]. Das beinhaltet alle Anwendungen auf die über das Internet zugegriffen wird und deren Hardware sich in einem Datenzentrum befindet, also nicht zwangsläufig unter der Kontrolle des Nutzers. Im Folgenden beschreiben wir Dropbox und wichtige Sekundärliteratur zu den Speicherdiensten in der Cloud.

2.1 Dropbox

Seit seiner Markteinführung 2008 hat sich Dropbox zu einem der beliebtesten Cloud Storage Provider entwickelt. Dropbox hat 25 Millionen Nutzer und speichert mehr als 100 Milliarden Dateien, so der Stand von Mai 2011 [2], mit einem durchschnittlich geschätzten 5-minütlichen Zuwachs von 1 Million Dateien [3]. Dropbox dient vornehmlich als Speicherplatz für Online Backups, die über das Internet abrufbar sind. Eine auf dem jeweiligen Rechner installierte Software sorgt dafür, dass die Dateien in den spezifizierten Ordnern ständig mit den Servern synchronisiert werden und Änderungen auf den unterschiedlichen Geräten des Nutzers immer aktuell bleiben. Es besteht auch die Möglichkeit, mehrere Nutzer auf gemeinsame Ordner zugreifen zu lassen. Änderungen in diesen gemeinsamen Ordnern werden automatisch synchronisiert und in alle beteiligten lokalen Dropbox-Ordner überspielt. Der Großteil des Dropbox-Clients ist in Phyton programmiert.

Dropbox verwendet zur Dateiidentifikation nicht das herkömmliche Dateikonzept. Vielmehr wird jede Datei in 4 MB große Abschnitte („Chunks“) unterteilt. Fügt ein Benutzer eine Datei in seinem Dropbox-Ordner hinzu, berechnet die Software unter Verwendung des SHA-256-Hashalgorithmus die Prüfsummen (Hashwerte) für jeden dieser Abschnitte und überprüft, ob sich eventuell bereits eine Datei mit den entsprechenden Hashwerten auf dem Sever befindet. Ist dem nicht so, erhält der Client die Anweisung, die entsprechenden Chunks hochzuladen. Befindet sich die Datei bereits am Server, erfolgt kein Upload. Stattdessen wird die vorhandene Datei mit dem Dropbox-Account verlinkt. Dieser Ansatz erlaubt Dropbox Einsparungen hinsichtlich Transfer- und Speicherbelegungskosten, der Anwender wiederum profitiert vom rascheren Synchronisationsprozess. Zusätzlich finden weitere Techniken zur Steigerung der Effizient Verwendung, z.B. Delta-Encoding, bei welchem nur die seit dem letzten Synchronisationsprozess modifizierten Abschnitte der Chunks übertragen werden. Die Wahrscheinlichkeit einer zufälligen Übereinstimmung der Prüfsummen unterschiedlicher Files ist verschwindend gering. Würden jedoch zwei unterschiedliche Dateien dieselben SHA-256 Hashwerte aufweisen, so wäre es einem Anwender möglich, auf Daten eines anderen Nutzers zuzugreifen, da die Datei die sich bereits auf dem Server befindet mit dem Account verknüpft werden würde.

Die Verbindung zwischen Client und Dropbox-Server wird über SSL gesichert. Die übertragenen Daten sind mit AES-256 verschlüsselt und werden im Amazons S3 Storage Service gespeichert, einem Teil des Amazon Web Services (AWS) [1]. Der AES-Schlüssel ist benutzerunabhängig und sichert die Daten nur während der Speicherung auf Amazon S3, die Sicherheit während der Übertragung basiert auf SSL. Unsere Untersuchungen zeigten, dass das Übertragungsprotokoll die Daten direkt an Amazon übermittelt. Die Verschlüsselung selbst geschieht erst im Nachhinein, was nicht nur Fragen in Bezug auf Alternation und Speicherort der Schlüssel selbst offen lässt, sondern auch Zweifel in Bezug auf potentielle Zugriffsmöglichkeiten von Amazon auf diese Schlüssel aufwirft. Nach dem Upload erfolgt – zur Bestätigung des erfolgreichen Transfers – ein neuerlicher Abgleich des Hashwerts der eben hochgeladenen Datei und dem der Datei, die sich am Rechner des Nutzers befindet. Divergieren die Prüfsummen, wird der Transfer des jeweiligen Chunks wiederholt. Der offensichtliche Nachteil ist, dass der Server nur Werte bereits hochgeladener Chunks berechnen kann, nicht aber überprüfen kann ob der Client sich tatsächlich in Besitz der Dateien befindet, auf die er über eine Hashwert-basierte Datei-Abfrage zugreifen möchte. Der Client Software wird ungeprüft Vertrauen entgegen gebracht. Wie unsere Forschungsergebnisse zeigen, eine optimale Ausgangslage für zahlreiche Manipulationsmöglichkeiten!

Aufgrund des Aufschwungs im Bereich Cloud Computing stieg auch die Anzahl der konkurrierenden Anbieter. Die Hersteller der bekannten Betriebssysteme entwickelten Dienste, die sich mit den eigenen Produkten kompatibel zeigten, wohingegen sich kleinere Startups durch Betriebssystemunabhängigkeit oder ausgereiftere Sicherheitsmerkmale hervortun.

2.2 Sekundärliteratur

Relevante Forschungsarbeit konzentriert sich hauptsächlich darauf, ob sich der Cloud Storage Anbieter weiterhin in Besitz der Anwenderdatei befindet und ob diese möglicherweise vom Betreiber des Dienstes verändert wurde. Eine interessante Untersuchung zur allgemeinen Sicherheitsthematik von Cloud Computing findet sich in [30]. Ein Überblick von Angriffsszenarien und Sicherheitsmängeln wird in [17] geboten. Shachams Artikel [11] zeigt, dass es recht einfach ist die interne Infrastruktur von Cloud Storage Anbietern abzubilden. Darüber hinaus wurden sog. „co-location“ Angriffe beschrieben. Hierbei wird die Kontrolle über eine virtuelle Maschine

übernommen, die sich auf derselben Hardware befindet wie das anzugreifende Zielsystem. Das ermöglicht in weiterer Folge Data Leakage und die Möglichkeit eines Side Channel-Angriffs auf die Virtual Maschine.

Frühe Publikationen [25, 14] behandeln unter anderem die Forschungsfrage, ob nichtvertrauenswürdigen Dritte auf eine Date zugreifen können, ohne dass diese im Vorfeld (komplett) von ihnen übertragen werden muss. Weiters wurden weitaus komplexere Protokolle vorgeschlagen um nachzuweisen dass der Server noch im Besitz der Datei ist [11, 12, 13, 20, 23, 29, 31, 32]. In einer aktuellen Publikation werden ähnliche Angriffe vorgestellt, allerdings ohne konkrete Implementierungen [24].

3 Unautorisierte Zugriffe auf Daten

In diesem Abschnitt stellen wir drei Angriffsszenarien auf Dropbox vor um zu demonstrieren, wie ein unerlaubter Zugriff auf Daten erfolgen kann. Einzige Voraussetzung für die Angriffe ist das die Hashwerte der Dateien bekannt sind. Diese Angriffsmöglichkeiten sind jedoch nicht spezifisch für Dropbox sondern für alle Dienstleister die Deduplikation im Hintergrund einsetzen.

3.1 Hash Value Manipulation Angriff

Für die Berechnung des SHA-256 Hashwertes wird von Dropbox nicht die Phyton-eigene Hashlib-Bibliothek verwendet. Stattdessen wird die Kalkulation unter Einbeziehung von NCrypto [6], einer Wrapper Library für OpenSSL [18] durchgeführt. Dropbox-Clients für Linux und Mac OS X verwenden Bibliotheken wie NCrypto ohne sie vorher zu verifizieren. Wir veränderten den frei zugänglichen Quellcode von NCrypto dahingehend, dass die von OpenSSL berechnete Prüfsumme durch eine von uns gewählte ersetzt wurde. Der Dropbox-Client erkennt die Manipulation nicht als solche und übermittelt für jede neue Datei im lokalen Dropbox-Ordner den modifizierten Hashwert an den Server. Falls die Datei auf dem Server noch nicht vorhanden ist wird sie hochgeladen. Nach dem Upload erfolgt – zur Bestätigung des erfolgreichen Transfers – ein neuerlicher Abgleich des Hashwerts der eben hochgeladenen Datei und dem der lokalen Datei. Ausgehend von unserer Client-seitigen Manipulation können die Hashwerte nicht übereinstimmen und der Server wird eine neuerliche Übertragung anfordern, um einen offensichtlichen Übermittlungsfehler auszugleichen. Stimmen die Prüfsummen der Serverdatenbank mit der Berechnung des Clients jedoch überein erfolgt eine Verlinkung der jeweiligen Datei bzw. des Chunks mit dem Dropbox-Account. Aufgrund unserer vorangegangenen Prüfsummenmanipulation erhalten wir somit unberechtigterweise Zugriff auf Dateien. Dieser Angriff bleibt vor dem legitimen Nutzer vollkommen verborgen. Sind dem Angreifer die Hashwerte bekannt, eröffnet sich ihm so die Möglichkeit, Daten direkt vom Dropbox-Server herunterzuladen. Da keine Interaktion mit einem Benutzer nötig ist kann das Opfer diesen Angriff nicht entdecken oder verhindern, da keinerlei Zugriff auf dessen Rechner notwendig ist. Dieser Angriff wird aber auch vom Server nicht als solcher identifiziert, zumal er ja davon ausgeht, dass der Angreifer sich im Besitz der Datei befindet und diese nur seinem lokalen Dropbox-Ordner hinzugefügt hat.

3.2 Stolen Host ID Angriff

Wird eine Dropbox-Client-Applikation auf einem Computer oder Smartphone installiert, wird ein einmaliger Schlüssel vergeben, der das jeweilige Gerät mit dem Dropbox-Account des Benutzers verknüpft. Diese Host-ID wird unter Zuhilfenahme von Benutzername und Passwort vom Dropbox-Server individuell berechnet. Das bedeutet, dass die Authentifizierung von Client und User später nicht über die Kombination von Benutzername und Passwort erfolgt, sondern über diese ID. Sobald Client und Host verbunden sind, ist keine neuerliche Anmeldung erforderlich, solange die Software nicht entfernt wird – eine augenscheinliche Sicherheitslücke. Denn ist der Angreifer (z.B. durch den Einsatz von Malware oder Social Engineering) erst in Besitz der Host-ID, erlangt er durch simples Ersetzen der eigenen ID Zugriff auf alle Daten des Opfers.

3.3 Direkter Download Angriff

Das Übermittlungsprotokoll zwischen Client-Software und Server basiert auf HTTPS. Die Client-Software kann Chunks über die URL https://dlclientXX.dropbox.com/retrieve (wobei XX durch fortlaufende Nummern ersetzt wird) anfordern, indem der SHA-256 Hash des Dateiabschnitts sowie eine gültige Host-ID als HTTPS POST Daten angegeben werden. Wie bereits erwähnt, fordert der Client Dateien über deren Hashwert an. Erstaunlicherweise ist für die Abfragemethode einer auf dem Dropbox-Server liegenden Datei nur die Kenntnis ihrer Prüfsumme sowie eine gültige Host-ID Voraussetzung. Es hat sich gezeigt, dass keinerlei Überprüfung erfolgt, ob der mit der Host-ID verlinkte Account bzw. dessen Benutzer tatsächlich in Besitz der angeforderten Datei ist. Wie wir später noch zeigen werden, werden von Dropbox kaum Daten gelöscht.

3.4 Erkennen von Angriffen

Zusammenfassend lässt sich sagen, dass ein Angreifer, der Zugang zum Inhalt der Client-Datenbank erlangt, alle Dateien die mit diesem Account verknüpft sind direkt vom Dropbox-Server laden kann. Dazu ist kein weiterer Zugriff auf das System des Geschädigten erforderlich, im einfachsten Fall ist überhaupt nur die Host-ID notwendig. Sollte sich das Interesse des Angreifers auf bestimmte Dateien beschränken so reicht zum Zugriff darauf die Kenntnis der entsprechenden Hashwerte aus. Findet eine der oben genannten Methoden Anwendung, ist der unerlaubte Zugriff für den tatsächlichen Besitzer der Dateien nicht ersichtlich.

Aus der Sicht des Cloud Storage Service Operators sind sowohl Stolen Host-ID Angriffe als auch der direkte Download Angriff bis zu einem bestimmten Grad nachweisbar. Dahingegen bleibt ein Hash-Manipulation-Angriff unerkannt, da diese Methode des unautorisierten Zugriffs den Anschein erweckt, der Angreifer würde sich bereits im Besitz der Datei befinden. Mögliche Gegenmaßnahmen beschreiben wir unter Abschnitt 6.

4 Angriffsvektoren und Online Slack Space

Folgender Abschnitt widmet sich Angriffstechniken die darauf abzielen, das Wirkungsprinzip von Cloud Storages generell auszunutzen. Beschrieben werden bereits bekannte Angriffsvektoren und diesbezügliche Umsetzungsmöglichkeiten mithilfe von Dropbox oder anderen Cloud Storage Services mit mangelhaften Sicherheitsvorkehrungen. Der Großteil zieht schwerwiegende Folgen nach sich und sollte im Gefahrenmodell dieser Services bedacht werden.

4.1 Online Slack Space

Der Upload von Dateien auf den Dropbox-Server ähnelt dem bereits beschriebenen Download über HTTPS. Der Upload eines Chunks über die Client-Software erfolgt nach Aufrufen von https://dl-clientXX.dropbox.com/store mittels Hashwert und Host-ID als HTTPS POST Daten, zusätzlich zu den eigentlichen Daten. Nach erfolgreichem Transfer erfolgt über eine weitere HTTPS-Anfrage eine Verlinkung von übertragener Datei und Host-ID und die neu hinzugefügten Dateien werden mit allen Geräten der Benutzers oder allen Accounts, die Zugriff auf diese Ordner haben, synchronisiert. Mit einem modifizierten Client ist es allerdings möglich, diesen Verlinkungsprozess zu unterbinden und unbemerkt unbeschränkt große Dateien hochzuladen, ohne dass dieses Datenvolumen vom Server als solches registriert wird und somit im Verborgenen weiterer Speicherplatz genutzt werden kann. Diese Sicherheitslücke haben wir als *Online Slack Space* bezeichnet, in Anlehnung an die forensische Methode Informationen im letzen Block von Dateien zu verbergen, die diesen Block nicht vollständig ausfüllen [21]. Doch anstatt Informationen im letzen Datenblock zu verbergen, geschieht dies hier unter Verwendung von Chunks, die nicht mit dem Benutzer-/Angreifer-Account verlinkt sind. In Verbindung mit einem Live Betriebssystem von CD gestartet verbleiben so keine nachweisbaren Spuren auf dem Computer, aus denen sich forensisch auswertbare Informationen ableiten ließen sobald der Rechner heruntergefahren wird. Unserer Meinung nach gibt es keine Beschränkung dafür wie viel Information so verborgen werden kann, zumal zum Missbrauch Mechanismen verwendet werden die jenen gleichen die von der Dropbox-Software selbst eingesetzt werden.

4.2 Angriffsvektoren

Ist dem Angreifer die Host-ID bekannt kann er beliebige Dateien hochladen und mit dem Account eines Opfers verlinken. Anstatt durch eine zweite HTTPS-Anfrage eine Verknüpfung mit dem eigenen Account anzufordern erfolgt diese mit einem willkürlich gewählten Account. In Kombination mit einer Schwachstelle in der Vorschaufunktion diverser Betriebssysteme ergibt sich so eine sehr effektive Angriffsmethode. Ein Angreifer könnte eine 0-day Schwachstelle dahingehend für seine Zwecke missbrauchen, als er eine manipulierte Datei in den Dropbox-Ordner des Opfers einfügt und abwartet bis der Benutzer das Verzeichnis öffnet und über die Voransichtsfunktion Code ausgeführt wird. Über gezieltes Social Engineering könnte man ein Opfer ebenfalls zum Ausführen einer Datei mit vielversprechendem Namen zu bewegen. In Besitz der Host-ID zu gelangen gestaltet sich herausfordernder, zumal Zugang zum Dateisystem erforderlich ist. Ein Großangriff über diese Methode ist trotz allem recht unwahrscheinlich, denn jemand der in der Lage ist in Besitz der Host-ID zu gelangen ist meistens bereits in der Lage das System zu übernehmen. Auch als „Covert Channel“ [16] oder um Daten aus einem Unternehmensnetzwerk zu schmuggeln ließen sich die hier beschriebenen Angriffe verwenden.

5 Auswertungen

Der folgende Abschnitt demonstriert einige der vorgestellten Angriffsszenarien. Wir überprüfen zum einen ob Dropbox dazu verwendet wird, gängige Dateien des Filesharing-Netzwerks *thepiratbay.org* zu hinterlegen, und wie lange Dateien im oben definierten *Online Slack Space* gespeichert werden können.

5.1 Auf Dropbox gespeicherte Daten

Mithilfe des Hash Manipulation-Angriffs sowie des Direkt Download Angriffs lässt sich leicht überprüfen, ob sich eine Datei bei bekanntem Hashwert bereits auf den Dropbox-Servern befindet. Anhand dieser Methoden haben wir getestet ob Dropbox zur Speicherung von Filesharing-Dateien verwendet wird, zumal einige Filesharing-Protokolle wie BitTorrent grundlegend auf Hashfunktionen zur Dateiidentifikation basieren. Dazu haben wir mit Stand Mitte September 2010 die Top-100-Torrents von thepiratebay.org heruntergeladen [7]. Aus urheberrechtlichen Gründen haben wir nur jene Dateien innerhalb der *.torrents* heruntergeladen, die nicht urheberrechtlich geschützt sind aber trotzdem als Beweis fungieren, dass Dropbox als Speicherort für solche Inhalte verwendet wird. Unsere BitTorrent Software wurde auch dahingehend modifiziert damit keine Daten zu anderen Nutzern geschickt werden [27]. Weiters fordert der von uns modifizierte Client jeweils nur das erste 4MB-Chunk an, da diese Information bereits ausreichend ist um festzumachen, ob die jeweilige Datei bei Dropbox vorhanden ist.

Die in den *.torrent* enthaltenen Dateien können wie folgt klassifiziert werden:

- urheberrechtlich geschützte Inhalte wie Filme, Musikdateien oder Folgen beliebter Serien;
- „Identifizierende Dateien“, die zwar direkt an urheberrechtlich geschützte Dateien geknüpft sind wie z.B. Sample-Dateien, Screenshots oder Prüfsummendateien, ihrerseits aber nicht dem Urheberrecht unterliegen;
- Statische Dateien, zum einen die *.torrents* Datei selbst oder auch Dateien die Links zu Webseiten der Gruppe die sich für die *.torrent* Datei verantwortlich zeigt enthalten

Die identifizierenden Dateien wiesen folgende Erweiterungen und Inhalte auf:

- *.nfo*: Informationen zum Ersteller des *.torrents*, Dateilisten, Installationshinweise oder detaillierte Informationen oder Bewertungen zu Filmen.
- *.srt*: beinhalten Untertitel zu Videodateien
- *.sfv*: beinhalten CRC32-Prüfsummen für jede Datei innerhalb der .torrent
- *.jpg*: beinhalten Screenshots von Filmen oder Album Covers
- *.torrent*: die BitTorrent-Datei an sich umfasst die Hashwerte aller Dateien, die Chunks sowie die für den Client nötigen Tracker-Informationen.

Von den Top-100 BitTorrent-Archiven enthielten 98 identifizierende Dateien, die zwei abweichenden *.torrents* entfernten wir aus dem Testset. Um herauszufinden, welche Zeitspanne in etwa zwischen Veröffentlichung eines BitTorrents und Hinterlegung auf Dropbox liegt, verglichen wir 24 Stunden später die aktuelle Top-100-Liste mit der bereits vorliegenden. 9 neue BitTorrents, hauptsächlich Folgen beliebter Serien, waren in der Zwischenzeit hinzugefügt worden. Wir erzeugten für jede *.torrent*-Datei und jede ihrer identifizierenden Datei SHA-256-Prüfsummen und überprüften, ob diese auf Dropbox zu finden sind. In Summe kamen wir so auf *368 Hashwerte*. In der Folge verwendeten wir den direkten Download Angriff mittels HTTPS um zu überprüfen ob die Dateien bei Dropbox gespeichert sind. 356 Dateien von insgesamt 368 waren abrufbar, die 12 verbleibenden waren demzufolge noch nie von einem Anwender bei Dropbox abgespeichert. Bei genauerer Betrachtung ergab sich aber, dass diese Fehler vereinzelt waren und die relevanten Inhalte auf Dropbox gespeichert waren. Das bedeutet dass für fast *100% der getesteten Dateien* auf Dropbox gespeichert werden, und es möglich gewesen wäre die Dateiinhalte über Dropbox herunterzuladen.

In weiterer Folge analysierten wir die Lebensdauer dieser *.torrents* und auch, wie schnell Dropbox-Nutzer diese und die dementsprechenden Inhalte downloaden und ihrerseits uploaden. Die meisten der *.torrent*-Dateien waren dabei sehr aktuell. Mehr als 20% der Top-100 *.torrent*-Dateien waren weniger als 24 Stunden auf piratebay.org, ehe wir über Dropbox darauf zugreifen konnten.

5.2 Online Slack Space Evaluation

Um zu überprüfen, ob man Dropbox dazu verwenden könnte Dateien zu verbergen indem man nach dem Upload eine Verlinkung mit einem Useraccount unterbindet, haben wir ein Set von 30 Dateien mit zufälligem Inhalt generiert und diese über die HTTPS-Request-Methode hochgeladen. Um zu prüfen ob alte Daten vom Server entfernt werden luden wir zusätzlich 55 mit Zufall als Inhalt über einen regulären Account auf Dropbox und löschten sie umgehend. Von den 55 Dateien befanden sich 30 in einer „Shared Folder“ und 25 in einem nicht gemeinsam verwendeten Ordner. Ferner untersuchten wir, ob eine Art „Garbage Collection“ existiert, die die Daten nach einer gewissen Zeitspanne entfernt. Wir luden die Dateien im 24-Stundentakt mittels dem HTTPS Direktdownload herunter und überprüften die Beständigkeit durch die Berechnung multipler Hashwerte.

Bis Ende April 2011, als Dropbox begonnen hatte, der HTTPS-Download-Attacke aktiv entgegenzuwirken, waren die 55 Dateien zu *100% durchgehend verfügbar*. Die Dateien waren auch mehr als sechs Monate nach dem Upload ausnahmslos vorhanden. Die über das HTTPS Interface hochgeladenen Dateien waren mehr als 4 Wochen verfügbar. Kurz vor einer Änderung von Dropbox um die HTTPS-Download-Attacke zu verhindern, sank die Verfügbarkeit auf 50%.

5.3 Diskussion der Ergebnisse

Wir waren überrascht, dass im Zusammenhang mit den jeweiligen *.torrent*-Dateien entweder der *.torrent*, der Inhalt oder beide über Dropbox abrufbar waren. Überraschend besonders deshalb, wenn man bedenkt das die *.torrent*-Dateien nur einige Stunden vor unserem Test erstellt wurden. 97% (356 von 368) Erfolgsrate weisen darauf hin, dass Dropbox in großem Maße zur Speicherung von Dateien aus Filesharing-Networks verwendet wird. Interessant ist auch anzumerken, dass der Inhalt mancher *.torrent*-Dateien bei weitem mehr Speicherplatz belegt als ein kostenloser Dropbox-Account umfasst (2 GB zum Zeitpunkt der Abhandlung). 11 der 107 getesteten *.torrents* verbrauchten mehr als 2 GB (mit 7,2 GB als Spitzenwert), da es sich um DVD-Images handelte. Daraus lässt sich schlussfolgern das die Person bzw. die Personen die diese Dateien auf Dropbox speichern einen Dropbox Pro Account (gegen eine monatliche Gebühr) oder durch das Anwerben von Freunden über das Dropbox-Bonusprogramm zusätzlichen Speicherplatz erworben hat. Aus unseren Untersuchungen können wir schlussendlich nur das Vorhandensein dieser Dateien ableiten, nicht aber quantifizierend feststellen, in welchem Ausmaß Dropbox unter den Benutzern zu Filesharing-Zwecken verwendet wird. Unsere Ergebnisse spiegeln nur wider, dass zumindest ein BitTorrent-User seine Downloads auf Dropbox abgelegt hat.

Unsere Untersuchungen zum Online Slack Space zeigen, dass es relativ einfach ist Daten versteckt auf Dropbox zu lagern, ohne strafrechtliche Folgen fürchten zu müssen. Außerdem zeigt es sich als recht simpel ohne zu zahlen an eigentlich kostenpflichtige Zusatzleistungen wie unlimitierte Dateiwiederherstellung oder erweiterte Dateiversionierung zu gelangen. Die Tatsache, dass man durch die Unterbindung des Verlinkens von Datei und Account unlimitierten Speicherplatz in Anspruch nehmen kann, resultiert in weiterer Folge aber auch in Erschwernissen hinsichtlich diesbezüglicher forensischer Untersuchungen. In einem erweiterten Szenario könnten Ermittler sich

sogar mit einem Computer ohne Festplatte konfrontiert sehen, der von einem Read-Only-Medium wie einer Linux Live-CD bootet und alle Daten im Online Slack Space ablegt. Ein solcher Computer würde keine brauchbaren Spuren oder lokale Beweise erzeugen [15]. Eine Problemstellung, die für die Computerforensik zukünftig sicher an Relevanz gewinnen wird, da es dem „Privatmodus“ moderner Browser gleicht, der keine Informationen lokal speichern [8].

6 Gegenmaßnahmen

Unsere Angriffe sind nicht nur in Bezug auf Dropbox anwendbar, sondern gefährden jedes Cloud Storage Service bei dem eine serverseitige Methode verwendet wird um Übertragung und Speicherung von Dateiduplikaten vorzubeugen. Die derzeitigen Implementierungen basieren auf einfacher Prüfsummenbildung, doch kann man nicht von einer vollkommenen Vertrauenswürdigkeit der Client-Software in Bezug auf die korrekte Berechnung des Hashwertes ausgehen. Eine verlässlichere Überprüfung ob der Anwender wirklich im Besitz der Datei ist sollte für jeden dieser Services implementiert werden. Es hat sich gezeigt, dass ein vollkommen neuer Aspekt aufzugreifen ist, denn bislang wurde in der Diskussion um die Vertrauenswürdigkeit dieser Dienste kaum auf die Gewichtigkeit des Clients als Sicherheitsschwachstelle eingegangen. Um sicherzustellen, dass sich der Client im Besitz einer Datei befindet ist ein verstärktes Protokoll nötig, egal ob nun auf Kryptographie basierend, auf statistischen Methoden, oder auf beidem.

Dies lässt sich über einen der jüngsten „Provable Data Possession“-Algorithmen wie in [11] verwendet umsetzen, bei denen der Cloud Storage Operator bestimmt, welche Anfragen der Client beantworten muss, ehe ihm Zugang zur Datei am Server gewährt wird. So lässt sich eine erneute Übertragung umgehen, die sowohl für den Nutzer als auch den Anbieter mit Kosten verbunden ist. Aktuelle Publikationen empfehlen unterschiedliche Ansätze mit unterschiedlichem Kosten [10, 12, 20]. Darüber hinaus sollte jeder Service seine Kommunikations- und Datentransferdienste über SSL abwickeln, was, wie wir feststellen konnten nicht immer der Fall ist.

Mehrere Schritte sind nötig um Dropbox abzusichern. Zu allererst muss ein Protokoll umgesetzt werden welches verhindert, dass der Hashwert einer Datei ausreicht um diese abzurufen. Längerfristig sollte jeder Cloud Storage Operator ein solches Protokoll einsetzten, wenn der Client nicht als Teil der vertrauenswürdigen Umgebung angesehen werden kann. Wir raten zur Implementierung eines einfachen Challenge-Response-Mechanismus. Im Prinzip wird einen Zwischenschritt zu den bestehenden Protokollen einfügt, während dem überprüft wird ob bei Übereinstimmung der Hashwerte der Client- und Serverdateien auch eine Übereinstimmung zufällig gewählter Dateibruchstücke vorliegt.

Das Hochladen von Chunks ohne Verknüpfung mit der Dropbox eines Anwenders sollte unterbunden werden, um einerseits unlimitierte Speicherplatzkapazität von bösartigen Anwendern entgegenzuwirken, und andererseits Online Slack Space zu verhindern. In manchen Fällen ist es immer noch kostengünstiger, zusätzliche Speicherkapazität anzufügen, als eine verlässliche Metrik zu entwickeln, welche Daten nun eigentlich zu löschen sind – will man jedoch den Missbrauch historischer Daten und von Online Slack Space verhindern, wird es dennoch unumgänglich, alle Chunks zu löschen, die nicht mit einer speziellen Datei in Verwendung verknüpft sind. Um die Sicherheitsvorkehrungen zusätzlich zu optimieren, können weitere behavioristische Aspekte einfließen. Zum Beispiel die Kontrolle der Host-ID-Aktivität: sobald ein Anwender seinen Computer hochfährt und über die Dropbox-Anbindung überprüft wird können Transaktionen mit dieser Host-ID zugelassen werden. Im Anschluss sollte es nur mehr über diese IP-Adresse gestattet sein, Daten von der Dropbox dieser Host-ID herunterzuladen. Sollte sich die IP des Anwenders

aufgrund der Verwendung eines VPN oder eines Standortwechsels ändern, muss die Verbindung zu Dropbox ohnehin neu aufgebaut werden und in diesem Zuge könnte die Zuordnung der Host-ID an die neuen IP erfolgen. Die Host-ID sollte, wenn zu Authentifizierungszwecken verwendet, wie ein Cookie eingesetzt werden: von dynamischer Natur und veränderlich. Doch vor allem muss von Dropbox verfolgt werden, welche Dateien sich in welcher Dropbox befinden. Fordert ein Anwender einen Chunk an, der sich nicht in seiner Dropbox befindet ist das für Dropbox leicht feststellbar.

Wir können kaum Angaben dazu machen, inwieweit unsere Verbesserungen Auswirkungen auf die Geschwindigkeit oder Kosten zeigen, doch sind wir der Meinung, dass die meisten unserer Vorschläge über einfache Datenbankabfragen effizient zu implementieren wären. Verschiedene Datenbesitz-Algorithmen wurden bereits hinsichtlich ihres Overheads untersucht, so z.B. S-DPD und E-DPD von [11] welche durch O(1) in Ihrem Aufwand beschränkt sind.

7 Zusammenfassung

In diesem Artikel wurden einige Angriffe auf Cloud Storage Anbieter beschrieben, die dem Angreifer ermöglichen unter bestimmten Umständen Dateien unberechtigterweise herunterzuladen. Wir demonstrierten deren Anwendbarkeit anhand des Online Storage Anbieters Dropbox und konnten zeigen, dass Dropbox massiv dazu verwendet wird, Daten von *thepiratebay.org*, einer beliebten BitTorrent-Seite, abzuspeichern. Darüber hinaus definierten wir den Begriff *Online Slack Space* und konnten mittels unserer Untersuchungen zeigen, dass auf diese Art effektiv Daten im Internet versteckt werden können.

Wir sind davon überzeugt, dass sich diese Sicherheitsmängel keinesfalls auf Dropbox beschränken, da das zugrunde Kommunikationsprotokoll sehr wahrscheinlich auch von anderen Cloud Storage Anbietern eingesetzt wird um Bandbreite und Speicherkapazität einzusparen. Die besprochenen Gegenmaßnahmen, allen voran die serverseitige Datenbesitzüberprüfung, sollten von allen Cloud Storage Anbietern umgesetzt werden.

Disclaimer

Die Details dieser Arbeit werden im August 2011 auch auf der USENIX Security Konferenz[1] in San Francisco vorgestellt.

Referenzen

[1] Amazon.com, Amazon Web Services (AWS). http://aws.amazon.com.
[2] At Dropbox, Over 100 Billion Files Served–And Counting, retrieved May 23rd, 2011. Online at http://gigaom.com/2011/05/23/at-dropbox-over-100-billion-files-served-and-counting/.
[3] Dropbox Users Save 1 Million Files Every 5 Minutes, retrieved May 24rd, 2011. Online at http://mashable.com/2011/05/23/dropbox-stats/.
[4] Grab the pitchforks!... again, retrieved April 19th, 2011. Online at http://benlog.com/articles/2011/04/19/grab-the-pitchforks-again/.
[5] How Dropbox sacrifices user privacy for cost savings, retrieved April 12th, 2011. Online at http://paranoia.dubfire.net/2011/04/how-dropbox-sacrifices-user-privacy-for.html.

[1] http://www.usenix.org/events/sec11

[6] NCrypto Homepage, retrieved June 1st, 2011. Online at http://ncrypto.sourceforge.net/.
[7] Piratebay top 100. Online at http://thepiratebay.org/top/all.
[8] AGGARWAL, G., BURSZTEIN, E., JACKSON, C., BONEH, D. An analysis of private browsing modes in modern browsers. In Proceedings of the 19th USENIX conference on Security (2010), USENIX Security'10.
[9] ARMBRUST, M., FOX, A., GRIFFITH, R., JOSEPH, A. D., KATZ, R., KONWINSKI, A., LEE, G., PATTERSON, D., RABKIN, A., STOICA, I., AND ZAHARIA, M. A view of cloud computing. Communications of the ACM 53, 4 (2010), 50–58.
[10] ATENIESE, G., BURNS, R., CURTMOLA, R., HERRING, J., KHAN, O., KISSNER, L., PETERSON, Z., AND SONG, D. Remote data checking using provable data possession. ACM Transactions on Information and System Security (TISSEC) 14, 1 (2011), 12.
[11] ATENIESE, G., BURNS, R., CURTMOLA, R., HERRING, J., KISSNER, L., PETERSON, Z., AND SONG, D. Provable data possession at untrusted stores. In Proceedings of the 14th ACM Conference on Computer and Communications Security (2007), CCS '07, ACM, pp. 598–609.
[12] ATENIESE, G., DI PIETRO, R., MANCINI, L., AND TSUDIK, G. Scalable and Efficient Provable Data Possession. In Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (2008), ACM, pp. 1–10.
[13] BOWERS, K., JUELS, A., AND OPREA, A. HAIL: A high-availability and integrity layer for cloud storage. In Proceedings of the 16th ACM Conference on Computer and Communications Security (2009), ACM, pp. 187–198.
[14] BOWERS, K., JUELS, A., AND OPREA, A. Proofs of retrievability: Theory and implementation. In Proceedings of the 2009 ACM Workshop on Cloud Computing Security (2009), ACM, pp. 43–54.
[15] BREZINSKI, D., AND KILLALEA, T. Guidelines for Evidence Collection and Archiving (RFC 3227). Network Working Group, The Internet Engineering Task Force (2002).
[16] CABUK, S., BRODLEY, C. E., AND SHIELDS, C. Ip covert timing channels: design and detection. In Proceedings of the 11th ACM Conference on Computer and Communications Security (2004), CCS '04, pp. 178–187.
[17] CHOW, R., GOLLE, P., JAKOBSSON, M., SHI, E., STADDON, J., MASUOKA, R., AND M OLINA, J. Controlling data in the Cloud: Outsourcing Computation without Outsourcing Control. In Proceedings of the 2009 ACM Workshop on Cloud Computing Security (2009), ACM, pp. 85–90.
[18] COX, M., ENGELSCHALL, R., HENSON, S., LAURIE, B., YOUNG, E., AND HUDSON, T. Openssl, 2001.
[19] EASTLAKE, D., AND HANSEN, T. US Secure Hash Algorithms (SHA and HMAC-SHA). Tech. rep., RFC 4634, July 2006.
[20] ERWAY, C., KUPC U, A., PAPAMANTHOU, C., AND TAMASSIA, R. Dynamic Provable Data Possession. In Proceedings of the 16th ACM Conference on Computer and Communications Security (2009), ACM, pp. 213–222.
[21] GARFINKEL, S., AND SHELAT, A. Remembrance of data passed: A study of disk sanitization practices. Security & Privacy, IEEE 1, 1 (2003), 17–27.
[22] GOLAND, Y., WHITEHEAD, E., FAIZI, A., CARTER, S., AND JENSEN, D. HTTP Extensions for Distributed Authoring-WEBDAV. Microsoft, UC Irvine, Netscape, Novell. Internet Proposed Standard Request for Comments (RFC) 2518 (1999).
[23] GROLIMUND, D., MEISSER, L., SCHMID, S., AND WATTEN-HOFER, R. Cryptree: A folder tree structure for cryptographic file systems. In Reliable Distributed Systems, 2006. SRDS'06. 25th IEEE Symposium on (2006), IEEE, pp. 189–198.
[24] HARNIK, D., PINKAS, B., AND SHULMAN-PELEG, A. Side channels in cloud services: Deduplication in cloud storage. Security & Privacy, IEEE 8, 6 (2010), 40–47.

[25] JUELS, A., AND KALISKI JR, B. PORs: Proofs of retrievability for large files. In Proceedings of the 14th ACM Conference on Computer and Communications Security (2007), ACM, pp. 584–597.
[26] KRISTOL, D. HTTP Cookies: Standards, privacy, and politics. ACM Transactions on Internet Technology (TOIT) 1, 2 (2001), 151–198.
[27] PIATEK, M., KOHNO, T., AND KRISHNAMURTHY, A. Challenges and directions for monitoring P2P file sharing networks - or: why my printer received a DMCA takedown notice. In Proceedings of the 3rd Conference on Hot Topics in Security (2008), USENIX Association, p. 12.
[28] POSTEL, J., AND REYNOLDS, J. RFC 959: File transfer protocol. Network Working Group (1985).
[29] SCHWARZ, T., AND MILLER, E. Store, forget, and check: Using algebraic signatures to check remotely administered storage. In Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on (2006), IEEE, p. 12.
[30] SUBASHINI, S., AND KAVITHA, V. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications (2010).
[31] WANG, C., WANG, Q., REN, K., AND LOU, W. Ensuring data storage security in cloud computing. In Quality of Service, 2009. IWQoS. 17th International Workshop on (2009), Ieee, pp. 1–9.
[32] WANG, Q., WANG, C., LI, J., REN, K., AND LOU, W. Enabling public verifiability and data dynamics for storage security in Cloud Computing. Computer Security–ESORICS 2009 (2010), 355–370.