Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space

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

SBA Research

secure

sba-research.org

Outline

Cloud Storage in General

Dropbox in particular

Results & Countermeasures

Cloud Storage Overview

ubuntu one
SPIDEROAK
iCloud
Dropbox
BACKBLAZE
wuala
mozy
amazon cloud drive
SugarSync

Systems Overview

Simple systems:

- FTP, WebDAV, NFS ...

More complex systems:

- Delta sync
- Folder sharing, incl. push
- P2P
- Encryption (?)

©porsche89y, flickr.com

More complex systems

Examples:

Name	Protocol	Encrypted transmission	Encrypted storage	Shared storage
Wuala	Cryptree	yes	yes	yes
SpiderOak	proprietary	yes	yes	yes
Ubuntu One	u1storage	yes	no	yes
Dropbox	proprietary	yes	no	yes

User has to choose threat model:

- Danger of honest, but curious operator?
- Unauthorized file access by third parties?
- Location of data?

Data Deduplication

At the server:

- Same file only stored once
- Benefit: Save storage space at the server

At the client:

- Calculate hash sum or other digest
- Benefit: Reduce communication with clients

Beneficial for everyone, right?

Data Deduplication

At the server:

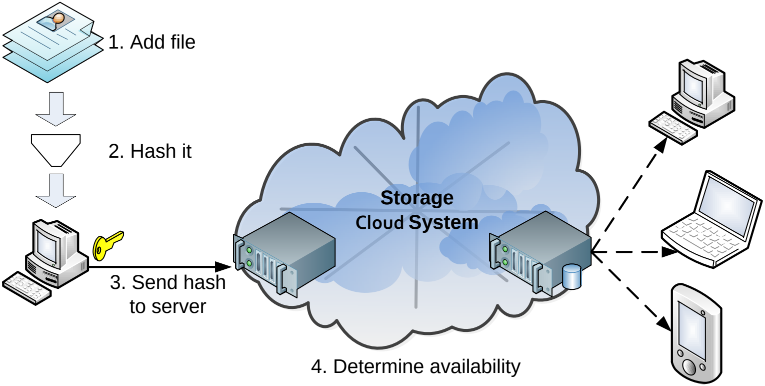
- Same file only stored once
- Benefit: Save storage space at the server

At the client:

- Calculate hash sum or other digest
- Benefit: Reduce communication with clients

Beneficial for everyone, right?

An efficient cloud architecture



Our contributions

Outline three attacks:

- Hash Manipulation Attack
- Stolen Host ID Attack
- Direct Up-/Download Attack

Show their feasibility on Dropbox, a popular cloud storage service

Details Dropbox

- uses Amazon Simple Storage System (S3)
- data deduplication, using SHA-256
- files split in 4 MB chunks
- (server-side) AES-256

- 25 million users
- Store more than 100 billion files
- 1 million files added every 5 minutes

Details Dropbox

- uses Amazon Simple Storage System (S3)
- data deduplication, using SHA-256
- files split in 4 MB chunks
- (server-side) AES-256

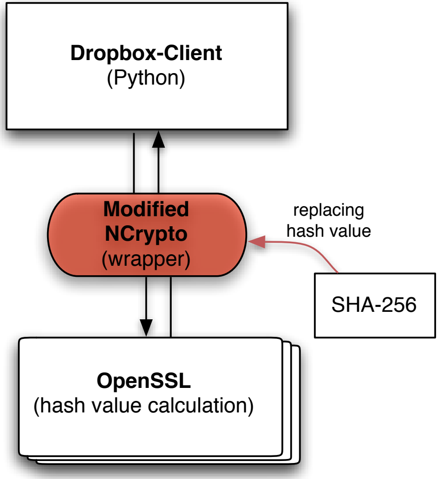
- 25 million users
- Store more than 100 billion files
- 1 million files added every 5 minutes

Attack #1 - Hash Manipulation Attack

Manipulating local hash computation

- Every time a new file is added
- Can be set arbitrarily
- Hash value needs to be known
- Results in **unauthorized file access**
- **Undetectable** for victim or Dropbox

Disclaimer: attack valid against all systems with client-side data deduplication

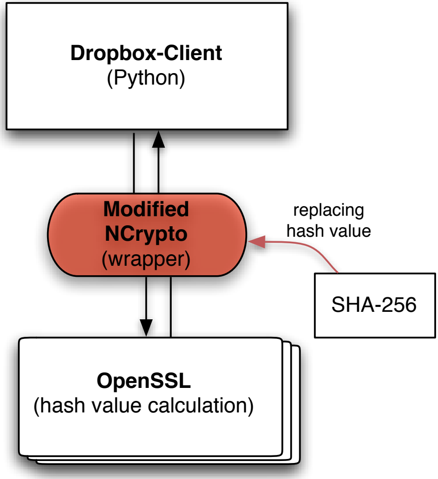


Attack #1 - Hash Manipulation Attack

Manipulating local hash computation

- Every time a new file is added
- Can be set arbitrarily
- Hash value needs to be known
- Results in **unauthorized file access**
- **Undetectable** for victim or Dropbox

Disclaimer: attack valid against all systems with client-side data deduplication



Attack #2 - Stolen Host ID Attack

Dropbox uses host ID to link particular host with account

- Credentials needed only once
- 128bit in length
- Arguable a security issue?
- Can be detected / prevented by Dropbox

Independently discovered by Derek Newton, April 2011

Attack #3 - Direct Up-/Download Attack

Transmission protocol is built upon HTTPS

- Simple HTTPS request: `https://dl-clientXX.dropbox.com/retrieve`
- As POST data: SHA-256 value & a valid host ID

- No check if chunk is linked with account!
- Easily exploitable
- Same effect as hash manipulation attack, but less stealth
- Can be detected / prevented by Dropbox

Attack #3 - Direct Up-/Download Attack

Transmission protocol is built upon HTTPS

- Simple HTTPS request: `https://dl-clientXX.dropbox.com/retrieve`
- As POST data: SHA-256 value & a valid host ID

- No check if chunk is linked with account!
- Easily exploitable
- Same effect as hash manipulation attack, but less stealth
- Can be detected / prevented by Dropbox

Attack #3 - Hiding data in the cloud

Same as retrieval, but for storing chunks

- Uploading without linking
- Simple HTTPS request: `https://dl-clientXX.dropbox.com/store`
- No storage quota / unlimited space
- If host ID is known: push data to other peoples Dropbox
- Can be detected / prevented by Dropbox

Attack #3 - Hiding data in the cloud

Same as retrieval, but for storing chunks

- Uploading without linking
- Simple HTTPS request:
  `https://dl-clientXX.dropbox.com/store`

- No storage quota / unlimited space
- If host ID is known: push data to other peoples Dropbox
- Can be detected / prevented by Dropbox

Evaluation - Part 1

We measured time until (hidden) chunks get deleted:

- Random data in multiple files
- Hidden upload: at least 4 weeks
- Regular upload: unlimited undelete possible ($>$ 6 months)

We used the HTTPS attack:

- Stealthiness was not an issue
- Hash manipulation equally suitable

Evaluation - Part 2

Popular files on Dropbox:

- `thepiratebay.org` Top 100 Torrent files
- Downloaded copyright-free content (.sfv, .nfo, ...)
- 97 % ($n = 368$) were retrievable
- Approx. 475k seeders
- 20 % of torrents were less than 24 hours old

Interpretation:

- At least one of the seeders uses Dropbox

Evaluation - Part 2

Popular files on Dropbox:

- `thepiratebay.org` Top 100 Torrent files
- Downloaded copyright-free content (.sfv, .nfo, ...)
- 97 % ($n = 368$) were retrievable
- Approx. 475k seeders
- 20 % of torrents were less than 24 hours old

Interpretation:

- At least one of the seeders uses Dropbox

Countermeasures

Countermeasures:

- Upload every file, no client-side data deduplication
- (Data possession proofs e.g., *[Ateniese et al., CCS 2007]*)
- "Proof of Ownage“, by Harnik et al. [under submission]

Our solution: Interactive challenge-response protocol

Challenge-Response

Challenge the client:

- Client and Server are in possession of the same file
- Client has to answer challenges
- Precomputable by the server
- Possible challenges:
  - Hash a subset of data
  - Append & XOR random bits and bytes
  - Possibly multiple rounds

Drawbacks:

- Challenges can be forwarded
- Not a real proof!
- But makes hash manipulation attacks detectable

Challenge-Response

Challenge the client:

- Client and Server are in possession of the same file
- Client has to answer challenges
- Precomputable by the server
- Possible challenges:
  - Hash a subset of data
  - Append & XOR random bits and bytes
  - Possibly multiple rounds

Drawbacks:

- Challenges can be forwarded
- Not a real proof!
- But makes hash manipulation attacks detectable

Timeline & Kudos

Timeline:

- First results in Summer 2010
- First paper draft November 2010
- Same time notified Dropbox via a national CERT

Independent results:

- Danny Harnik, Benny Pinkas & Alexandra Shulman-Peleg (Dec. 2010)
- Derek Newton, Stolen host ID Attack (Apr. 2011)
- Chris Soghoian & Ashkan Soltani, Information leakage and FTC complaint
- Various others tools: Dropship, DropboxReader, ...

Timeline & Kudos

Timeline:

- First results in Summer 2010
- First paper draft November 2010
- Same time notified Dropbox via a national CERT

Independent results:

- Danny Harnik, Benny Pinkas & Alexandra Shulman-Peleg (Dec. 2010)
- Derek Newton, Stolen host ID Attack (Apr. 2011)
- Chris Soghoian & Ashkan Soltani, Information leakage and FTC complaint
- Various others tools: Dropship, DropboxReader, ...

Conclusion

Aftermath - Dropbox reacted in April 2011:

- They fixed the HTTPS Up-/Download Attack
- Host ID is now encrypted on disk
- No more client-side data deduplication (recently)

Conclusion:

- Hash manipulation attack is undetectable
- Applicable to all services using client-side data deduplication
- Client-side data possession proofs needed

Conclusion

Aftermath - Dropbox reacted in April 2011:

- They fixed the HTTPS Up-/Download Attack
- Host ID is now encrypted on disk
- No more client-side data deduplication (recently)

Conclusion:

- Hash manipulation attack is undetectable
- Applicable to all services using client-side data deduplication
- Client-side data possession proofs needed

Thank you for your time!

Questions?

mmulazzani@sba-research.org
Try Dropbox (and get me extra space) :)

`http://db.tt/dFKyXce`