

Exposing the Lack of Privacy in File Hosting Services

Nick Nikiforakis

Marco Balduzzi

Steven Van Acker

Wouter Joosen

Davide Balzarotti



Sharing is caring

- Internet expanding
 - More users
 - More Web services
 - More Web technologies
- Users need to share files
 - P2P is not always the answer
 - Emails?

File Hosting Services

- Cloud-storage for the masses
- Share files with other users
- Security through obscurity access-control
- Sharing personal documents as well as pirated files [1]

Lifecycle of a file

- Alice decides to shares some digital content (file) through a FHS
- FHS received the file, stores it on its Cloud and generates an identifier which it:
 - i. binds with the uploaded file
 - ii. returns to the user in a URI form
- URI is shared depending on the nature of the uploaded file

File Identifier & Privacy

- The file ID is used to enforce access-control in a security-through-obscurity way
 - ID == access to file
- FHS are typically not-searchable
 - ID acts as a shared secret between a FHS and each user's files
 - Non-owners should not be able to “guess” this secret

Top 100 FHS

- We studied the top 100 FHS to discover, among others, the way they generate unique “secret” identifiers
 - Uploading files, recording the given ID and comparing
- Removed 12 that had search/browse capabilities

Sequential IDs

- 34/88 FHS were generating sequential identifiers
 - numeric, or alphanumerical
- 20/34 did not append any other non-guessable information
 - e.g. filename or secondary ID
- E.g.
 - <http://vulnerable.com/9996>
 - <http://vulnerable.com/9997>
 - <http://vulnerable.com/9998>

Scraping file information

- Given a link a user must follow a set of steps to actually download a file
 - Download “foo.txt” -> “Free user” -> Wait n seconds -> “Download “foo.txt”
- Advantageous for an attacker
 - Visit first page, scrape filename and file-size
 - Download only the files of interest

Crawling 20 FHS

- Designed a crawler for the 20 sequential FHS
- Run for 30 days
 - Random delays to avoid DoS and blacklisting
 - Scraping only the filenames and sizes (privacy)
- Results:
 - > 310,000 file records

Finding private files...

- Depending on the nature of a file, it will be shared in different ways
- Exploit the ubiquity of search-engine crawlers to characterize a file as private or public.
- Given a filename
 - 0 search results -> Private



Private Files Results

- Using Bing:
 - 54.16% of files returned 0 search results
 - Rough approximation of private files due to close pirate communities

Filetype	#Private documents
Images (JPG,GIF,BMP)	27,711
Archives (ZIP)	13,354
Portable Document Format	7,137
MS Office Word	3,686
MS Office Excel Sheets	1,182
MS Office PowerPoint	967

Random but short

- Brute-force short random identifiers

Length	Charset	#Tries	#Files Found
6	Numeric	617,169	728
6	Alphanumeric	526,650	586
8	Numeric	920,631	332

Design & Implementation errors

- Security audit of a popular FHS software product
 - Used in 13% of FHSs
 - Directory traversal vulnerability
 - De-randomization attack for deletion code
 - Report-link contained the first 10 characters of the 14-character delete code
 - 16^{14} -> 16^4 combinations

Status...

- File hosting services are vulnerable
 - Sequential identifiers
 - Weak non-sequential identifiers
 - Bugs in their source code
- Do attackers know about this?
 - How do we found out?

HoneyFiles

- HoneyPot for FHS attackers
 - Decoy files promising valuable content
 - Each file “called-home” when opened
 - `` in HTML files
 - embedded HTML in doc files
 - TCP socket in executables
 - Attempt to open page in pdf files



Carding forum

- card3rz.co.cc
 - fake underground carding community
 - One of the decoy files contained valid credentials for the forum
- Reasons:
 - i. Hide our monitors
 - ii. Do attackers use data that they find in illegally obtained files?





Card3rz Login

Username

Password

Login

This website is for similarly minded people. Unless you have a valid username/password combination, you are advised to leave...

LEET 2011
NOW

HoneyFiles results

- Monitoring sequential FHSs for 30 days:
 - 275 honeyfile accesses
 - more than 80 unique IP addresses
 - 7 different sequential FHSs
 - 1 had a catalogue functionality
 - 2 had a search functionality
 - 4 had neither
 - Accesses from all around the world

Geo-location



HoneyFiles results

- Download ratio of each file:

Claimed content	Download ratio
Credentials to PayPal accounts	40.36%
Credentials for card3rz.co.cc	21.81%
PayPal account Generator	17.45%
Leaked customer list	9.09%
Sniffed email	6.81%
List of emails for spamming purposes	5.09%

card3rz.co.cc results

- 93 successful logins
 - 43 different IP addresses
 - 32% came back at a later time
- Attacks against the monitor and the login-form
 - SQL-injection & file-inclusion attacks
- Attackers do in-fact use data from illegally obtained files

Honeyfiles cntd.

- Monitor 20 non-seq. FHSs for 10 days:
 - 24 honeyfile accesses
 - 13 unique IP addresses
 - 3 different FHSs
 - Two of them offered a search functionality
 - The third didn't
 - but actually did...

Status...

- File hosting services are vulnerable
 - Sequential identifiers
 - Weak non-sequential identifiers
 - Bugs in their source code
- Attackers are abusing them
 - They are using the data found in other user's files

SecureFS

- A client must protect himself
- Encryption is a good way
 - Do people know how to?
 - If they do know, does their OS assist them?
- SecureFS
 - Encryption to protect a user's data
 - Steganography to mislead potential attackers



SecureFS

- Browser-plugin monitoring uploads and downloads
- Protects uploads on-the-fly:

important.doc



SecureFS

- Browser-plugin monitoring uploads and downloads
- Rewrites download links to include the random key
 - <http://unsafefhs.com/12345>
 - [http://unsafefhs.com/12345/sfs_key/\[RND_KEY\]](http://unsafefhs.com/12345/sfs_key/[RND_KEY])

Future Work

- Security/Privacy monitor for well-known FHS
- Every illegal download/open would be registered to a Web service
 - Insecure FHS
 - Help users to choose a safe one
 - Put pressure on FHS developers to redesign their systems

Ethics

- We didn't download user files
- HoneyFiles were not harmful to a user's computer in any way
- HoneyFiles were uploaded as private files in various FHSs
- All vulnerable FHSs were notified

Conclusion

- Large percentage of FHSs fail to provide the user with adequate privacy
 - Hundreds of thousands of files ready to be misused
- Attackers know & exploit this fact
- A user must protect himself:
 - SecureFS

Thank you

- Questions/Comments?

nick.nikiforakis@cs.kuleuven.be