Inside Druva inSync

Looking beyond marketing BS

Dhiru Kholia (dhiru@openwall.com)

19.06.2013

About Me

- JtR, Ettercap and hashkill developer
- Metasploit and Nmap contributor
- #openwall channel on Freenode
- @DhiruKholia on Twitter

Agenda

- About Druva inSync
- Authentication Issues
- Licensing Hack
- Remote code execution
- Misuse of SSL
- Bytecode Protection Issues
- Vendor Response
- Demo

About inSync

- Druva inSync is an on-premise and cloud-based backup software (Wikipedia)
- Druva provides enterprise laptop backup solutions that protect corporate users data with 10x faster backups and 90% reduction in storage requirements (Twitter)
- The data is encrypted both during transit (256-bit SSL) and in the storage (256-bit AES).

Popularity

- 1,823 enterprises
- 1,324,587 endpoints
- 48 countries
- 98% Customer Satisfaction Rate
- Customers include NASA, PwC, Deloitte, Amway, Xerox and McAfee among others
- Rated ”excellent” by Gartner
- (June 2013 data)

Security Claims

- “inSync Cloud offers the industry-best security.”
- SAS 70 Type II, PCI DSS Level 1, ISO 27001, ISAE 3000 Type I
- Industry-First Two-Factor Encryption. Even Druva can’t access your data.
- How do they do de-duplication? Does ”dropship” like attack works?

Authentication Database

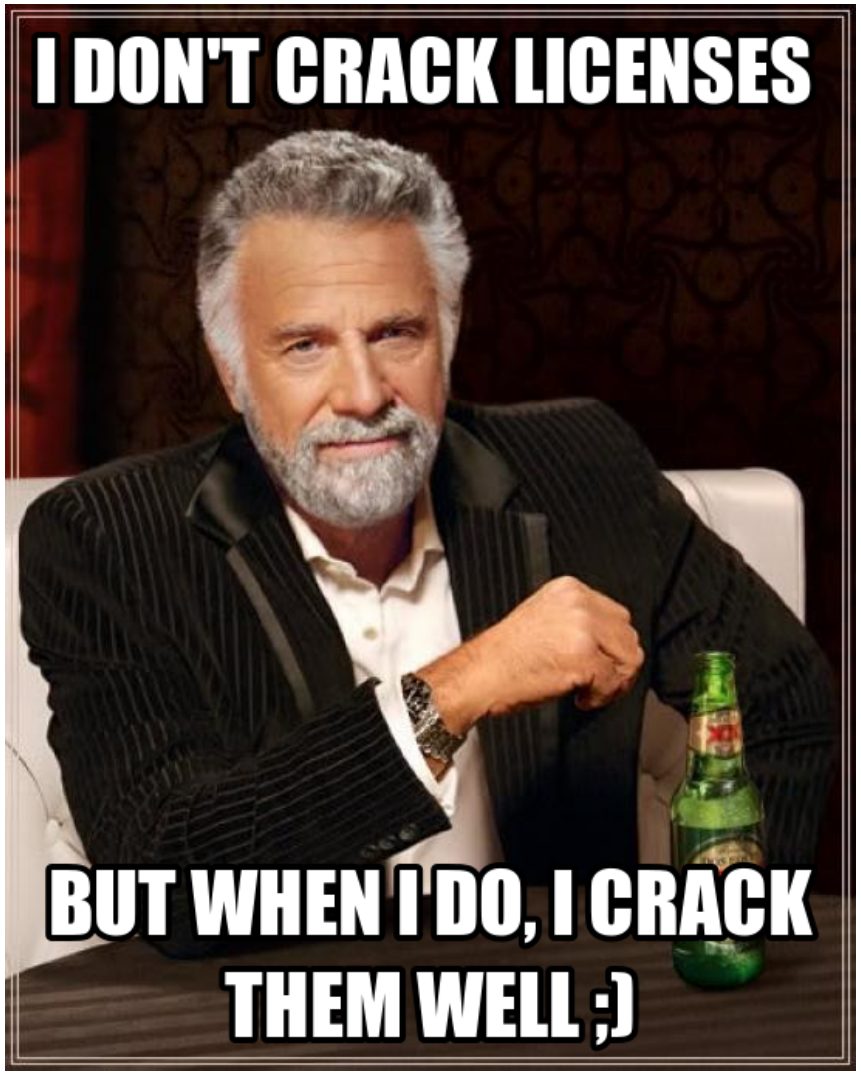
- ”inSync Cloud offers the industry-best security”
- Password hashes are stored in a SQLite database file
- Uses single iteration of md5 to protect admin and user passwords. hash = md5(id + password)
- select id, name, emailid, password from administrator
- Such hashes are crackable at high speeds using JtR or hashcat family of softwares. (4.2B c/s possible with oclHashcat-lite on AMD 7970)
- Ever heard about PBKDF2?

MD5

inSync Licensing

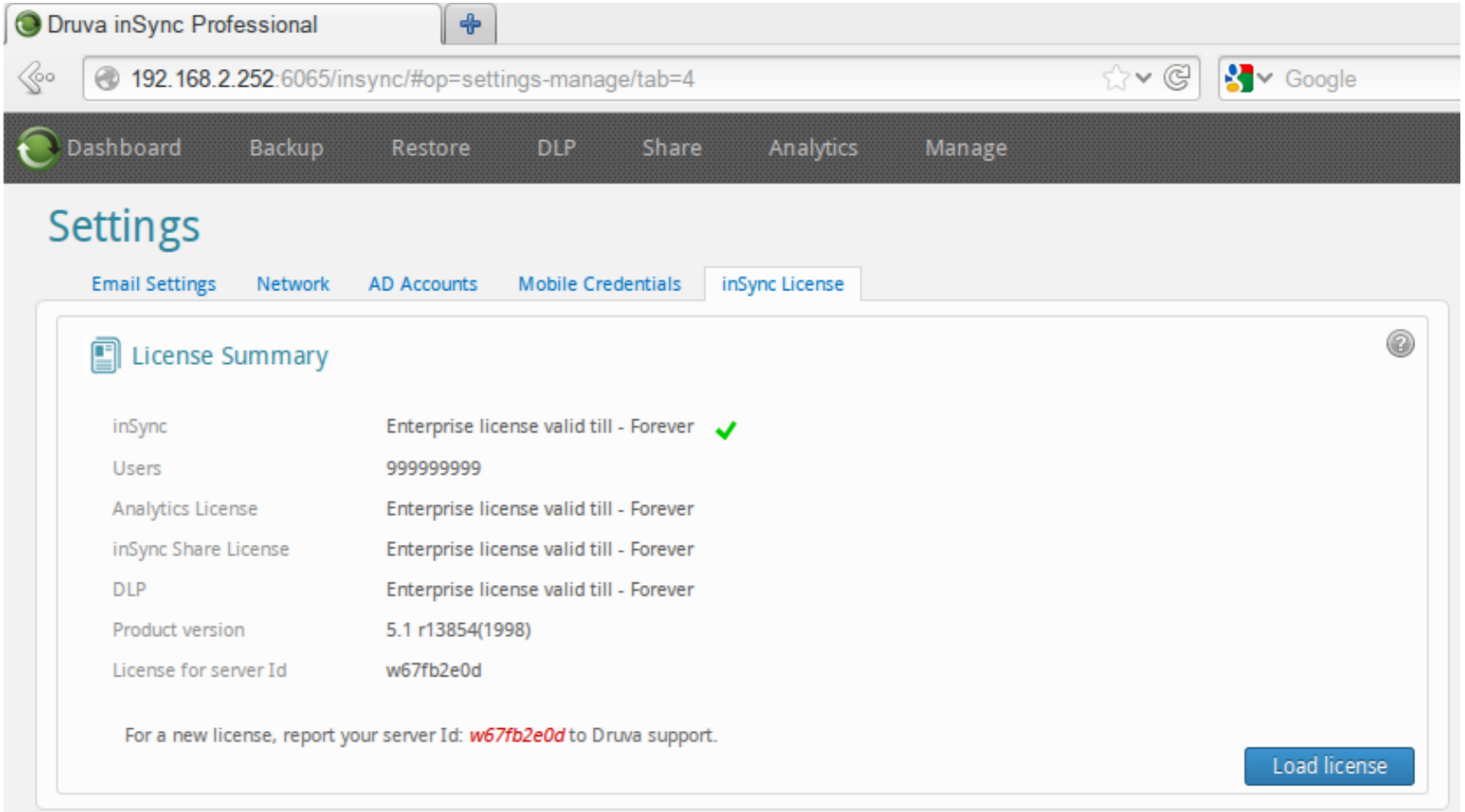
- Trial license expires after 30 days
- Enterprise license is limited to 500 users per server.
- Need to pay extra $$$ for features like file sharing, DLP and analytics and these are ”time-bombed”.

Hacking Licensing



License Hack

- It is easy to reverse-engineer and generate unlimited Enterprise licenses.



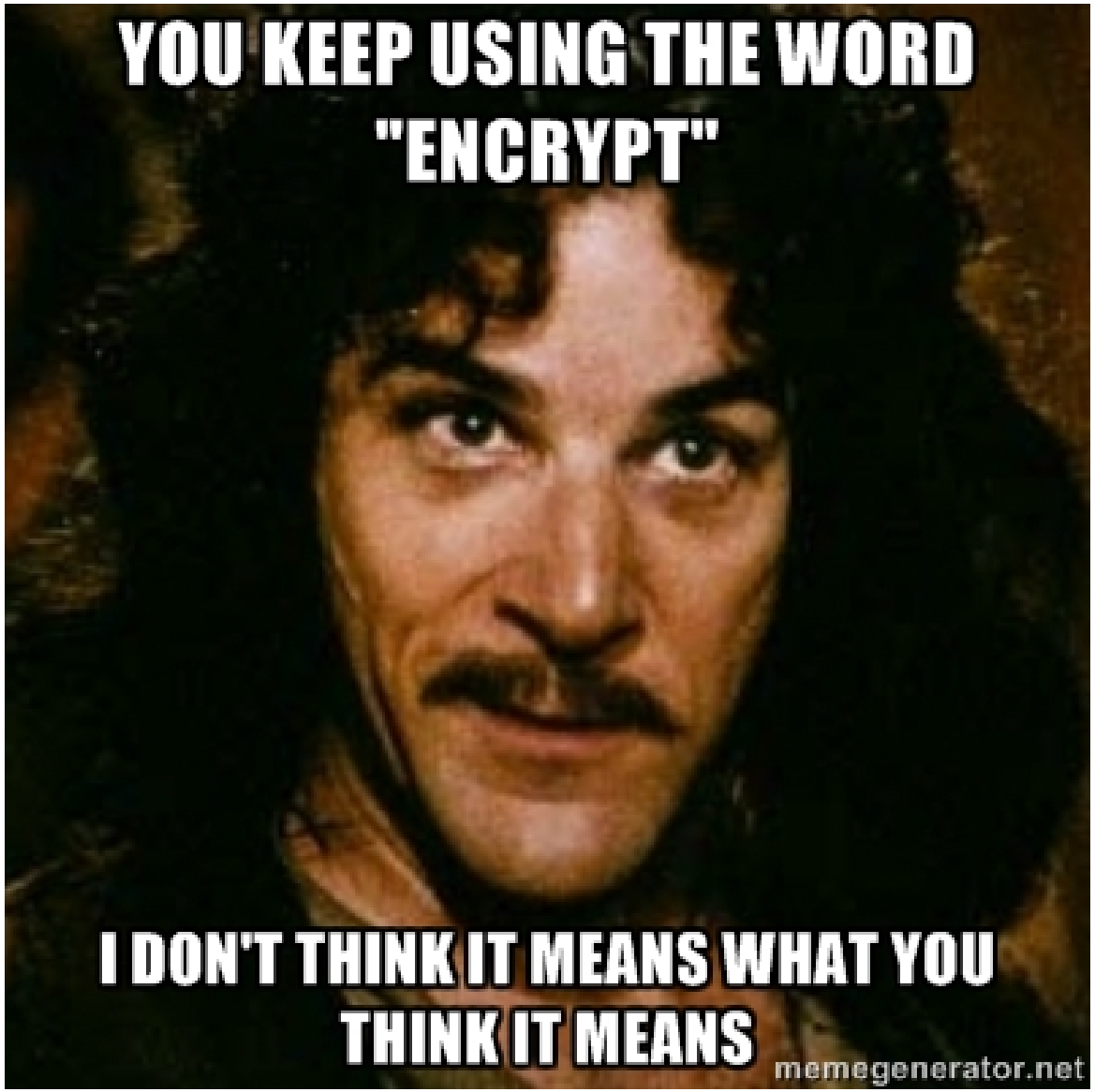
License Hack

- inSync places md5 hash (of all fields) at the end of plain-text license string. Easy to manipulate and change numbers of users, expiration dates etc. Something like ”a=b:md5(a=b)”.
- inSync then ”encrypts” this string with the following ”encryption” function

```
def encrypt(in):
return base64.b64encode(bz2.compress(in, 9))
```

- Such ”encrypted” license strings are easy to ”decrypt”

Preventing License Hacks



Preventing License Hacks

- Hard problem to solve.
- Even the industry ”best” protection systems have been cracked (eventually)!
- Asymmetric cryptography can help? (WinRAR)

SMTP configuration

- It is mandatory to configure SMTP
- Same ”encryption” function is used to ”encrypt” SMTP password.
- Possible to do insidious social-engineering attacks if access to this SMTP account is gained.

Better solution

- Maybe try using CryptoAPI for slightly better protection.

Arbitrary remote code execution

- License files are in fact pickled strings.
- We can generate malicious license files!
- A successful social engineering attack on inSync administrator can lead to complete data loss!

pickle code execution

- pickle is the standard mechanism for object serialization in Python
- By design, pickle allows code execution. It sure is convenient but isn't secure.

```
import pickle

pickle.loads("cos\nsystem\n(S'ls ~'\ntR.")
```

- This code runs "ls" command. Source: Nadia Alramli's Blog
- Google for "Sour Pickles Black Hat" for more information

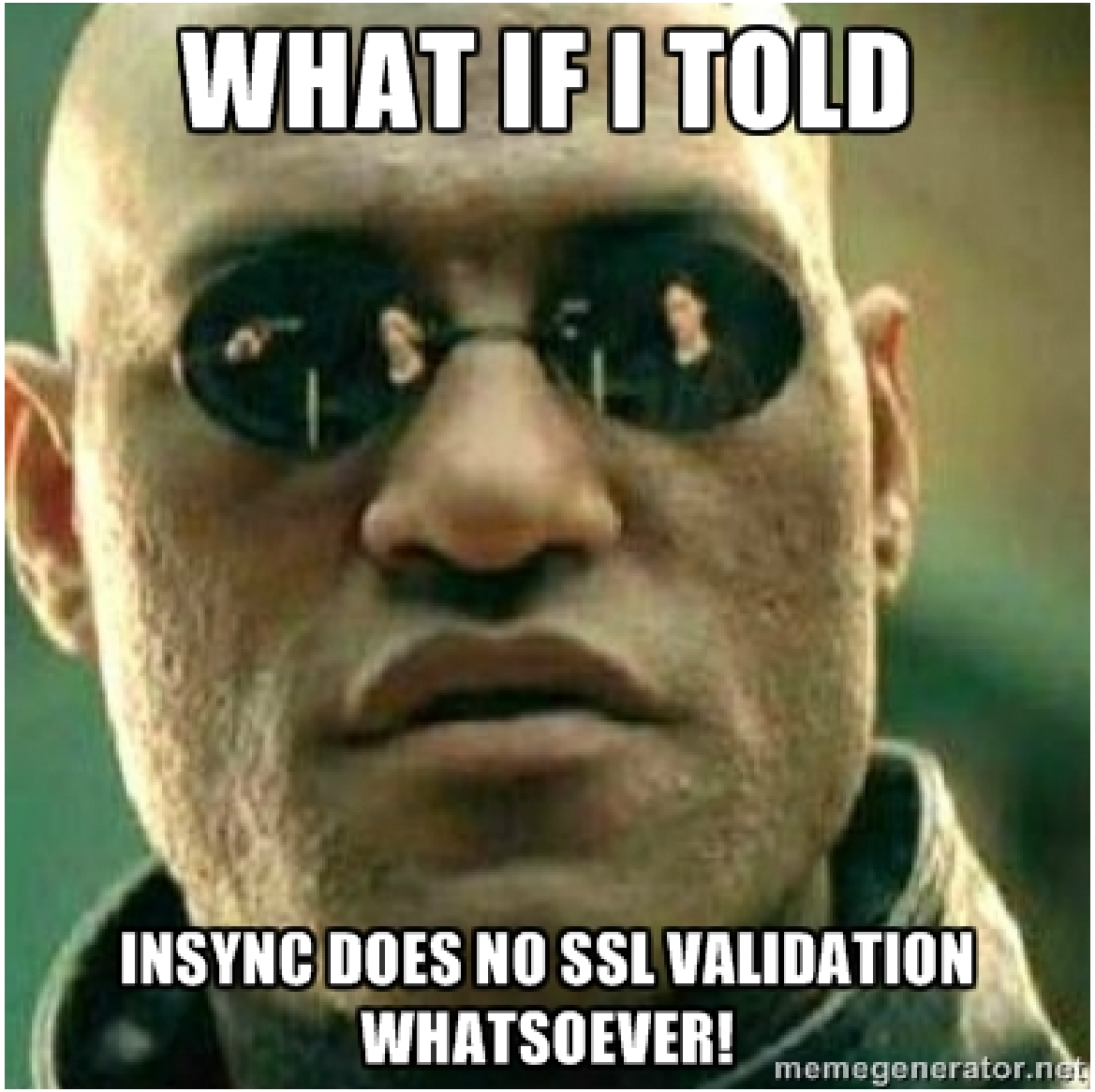
Arbitrary remote code execution

- Never do ”pickle.load(file_handle)” when data source is not trusted and controlled.
- By design, pickle allows code execution. It sure is convenient but isn’t secure.
- Writing your own custom plain-text format is trivial (or just use JSON).

on-the-wire data protection claims

- ”inSync Cloud offers the industry-best security”
- ”256-bit SSL encryption for data in transit”
- ”Secure HTTPS and LDAPS protocols for access”

Reality


WHAT IF I TOLD
INSYNC DOES NO SSL VALIDATION WHATSOEVER!
memegenerator.net

Reality

- 256-bit SSL? Sure
- However, No SSL certificate verification is done.
- inSync client (installed on end devices) does NO verification of SSL certificates whatsoever. #epicfail
- Hello MiTM attacks!

MiTM code

- https://github.com/kholia/ettercap/tree/inSync
- Allows to steal passwords or ”hashes”
- Anyone of them can be used to steal (or wipe) all data!

MiTM prevention

- ”256-bit SSL encryption” text is used for pure marketing purposes by Druva.
- 256-bit doesn’t do any good if you are not doing SSL certificate validation
- Deploy ”real” certificates on inSync server for best results
- Publish (and verify) certificate fingerprint

(lack of) Bytecode Protection

- Druva uses py2exe (on Windows) to bundle and distribute inSync
- It is easy to reverse-engineer Druva inSync.
- unzip command + a Python decompiler (uncompyle2) are enough to obtain complete source-code of inSync.

Generic ”unpacker”

```
import zipfile

fileName = "inSync"

ztype = zipfile.ZIP_DEFLATED

f = zipfile.PyZipFile(fileName, "r", ztype)

f.extractall("pyc_orig")
```

Decompiling bytecode

- https://github.com/Mysterie/uncompyle2
- As easy as doing ”uncompyle2 -o hello.py hello.pyc”
- There is no protection whatsoever. Have fun ;)

Bytecode Protection Techniques

- Opcode obfuscation
- Bytecode encryption
- Booby-trapped customized python27.dll (like Dropbox) ;)
- Static linking of Python interpreter

Vendor Response

My Thoughts

Companies don't see it as a security problem; they see it as a PR problem" - Bruce Schneier on security issues.

- Contacted vendor on 18th December 2012. They asked for "details" which I sent promptly. No further contact.
- Contacted CEO and CTO on 2nd January 2013.
- Got vendor response on 27th March 2013. Druva is working on fixing some of the problems.

Twitter Encounter

Dhiru Kholia,
Your Tweet got a reply!

Dhiru Kholia @DhiruKholia 02 Jan
@druvainc Found arbitrary remote code execution (remote data wipe is possible!) in inSync. Maybe this will get your attention ;)

Druva Inc
@druvainc

@DhiruKholia Thanks for the note. We will have a product specialist contact you shortly. Stay Tuned!
08:59 AM - 15 Jan 13

They deleted their own tweet for unknown reasons ;)

Don’t repeat mistakes @druvainc

- LinkedIn leak (6.5 million SHA1 hashes, over 90% of them got cracked!)
- Best not to invent your own crazy schemes
- Read up on PBKDF2
- Stop misusing / using pickle
- Compression is not ”encryption” (and hashing is not encryption)

Future Work

- Obtain trial to their cloud version of inSync and break it. Any help is welcome!
- Understand inSync's key derivation process. Most likely it will be single iteration of MD5 #lol
- Reverse-engineer custom storage "blob" format used by inSync
- Can we run from decompiled "sources"?
- What about releasing an open-source Druva inSync client? ;)

Questions?

DEMO!

- http://www.druva.com/insync-enterprise-releases
- Extracting bytecode
- De-compiling bytecode

Thanks!