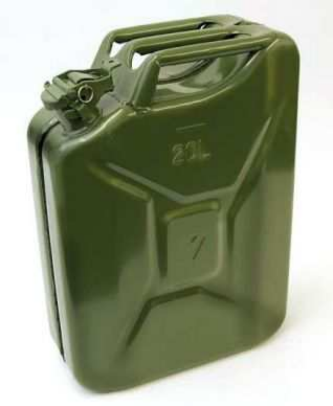
Reverse engineering

sost@fabletech.com

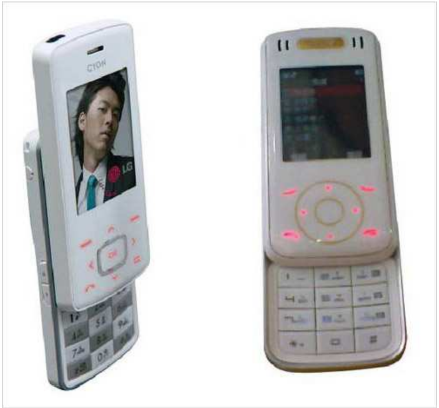
Fabletech.com

2011-04-26

Jerry can

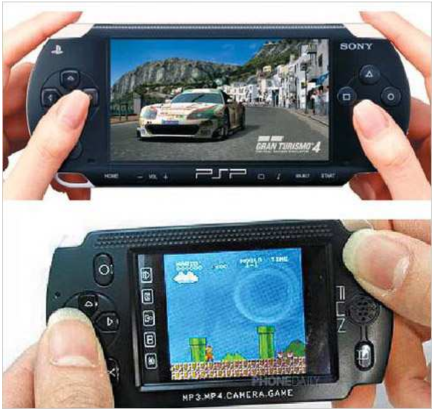


LG



Nokia

PSP



PS3

Languages

- Python
- Java
- C#

FableTech

Python bytecode

- .pyc
- .pyo
- .pyd

FableTech

Python .pyc/.pyo format

Magic number

The magic number is the first four bytes of the file.

.pyc dump

```
d1 f2 0d 0a 52 7b b6 4d 63 00
00 00 00 00 00 00
```

Python .pyc/.pyo format

Timestamp

The timestamp is byte 4 to 8 of the file.

.pyc dump

```
d1 f2 0d 0a 52 7b b6 4d 63 00
00 00 00 00 00 00 ...
```

Python .pyc/.pyo format

Python bytecode

The bytecode is from byte 8 to the end of the file. This is a marshaled file.

.pyc dump

```
d1 f2 0d 0a 52 7b b6 4d 63 00
00 00 00 00 00 00 ...
```

Decompiling Python

- unpyc
- decompyle (created in defcon ctf)
- python dis(disassembling)

FableTech

Decompiling java

- jad
- jd-gui
- jode (dissembling java bytecode)

FableTech

Decompiling C#

- Reflector

FableTech

anti reversing techniques

Obfuscation vs Packers vs Protectors

FableTech

Obfuscation

- Less readable code
- Remove debuginfo
- Remove comments
- Alter flow control
- Restructure code
- inject unusable code

FableTech

Java/C#

- String encryption
- String replacement
- Unusable code
- shitty code etc.

FableTech

Python

- Obfuscation (same as C# and Java)
- Wrappers Py2exe, Freeze, Py2app
- Change magic numbers
- Custom marshaling
- change opcodes

FableTech

You have never seen this slide period