

Captcha Intruder

<http://cintruder.sf.net>

THSF – 2012

EN

- OCR (Optical Character Recognition)
- CIntruder: **The Captcha Intruder**
- Installation/Usage:
 - *tracking*
 - *training*
 - *cracking*
 - *sharing (CInet)*
- Ninja scenarios
- Present → Future

OCR (Optical Character Recognition)

“mechanical or electronic conversion of scanned images of handwritten, typewritten or printed text into machine-encoded text”

OCR is a field of research in:

- *pattern recognition*
- *artificial intelligence*
- *computer vision*

Ray Kurzweil (1974): "best application of this technology would be to create a reading machine for the blind"

OCR (Optical Character Recognition)

List of OCR software (not security pentesting oriented):

http://en.wikipedia.org/wiki/List_of_optical_character_recognition_software

By licenses:

- *Proprietary: 18*
- *BSD: 2*
- *Apache: 3*
- *GPL: 3*

Some code/ideas for captcha 'crackers':

- *PWNtcha*
- *Captcha Sniper*
- *Captcha Killer 2.0*
- *Tesseract OCR engine*

CIntruder: The Captcha Intruder

Semi-automatic pentesting tool to bypass captchas.

- Python (2.x)
- Semi-automatic tasks
- Free Software → GPLv3.0
- Start-up: 1st April, 2012
- Current version: v0.2 (27/04/2012)
- Changelog: [review](#)
- Source: [download it!](#)
- Code repository:

[hg clone http://hg.code.sf.net/p/cintruder/code](http://hg.code.sf.net/p/cintruder/code) cintruder-code

Installation

Code runs on many platforms. It requires:

- *python-pycurl* *Python bindings to libcurl*
- *python-libxml2* *Python bindings for the GNOME XML library*
- *python-imaging* *Python Imaging Library*

On Debian-based systems (ex: Ubuntu), run:

sudo apt-get install python-pycurl python-libxml2 python-imaging

Usage

--version	show program's version number and exit
-h, --help	show this help message and exit
-v, --verbose	active verbose mode output results
--proxy=PROXY	use proxy server (tor: http://localhost:8118)
--track=TRACK	download a number of captchas from url (to: 'inputs/')
--train=TRAIN	apply common OCR techniques to captcha
--crack=CRACK	brute force using local dictionary (from: 'iconset/')
--xml=XML	export result to xml format

Advanced OCR (training):

--set-id=SETIDS	set colour's id manually (use -v for details)
--editor	launch an editor to apply image filters

Modules (training):

--list	list available modules (from: 'core/mods/')
--mod=NAME	train using a specific OCR exploiting module

Usage

Handling (cracking):

--tool=COMMAND replace suggested word on commands of another tool. use 'CINT' marker like flag (ex: 'txtCaptcha=CINT')

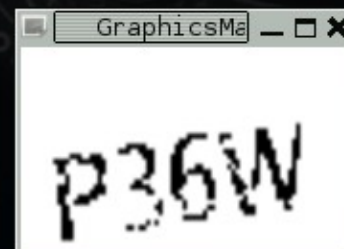
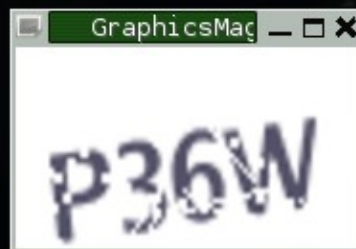
CIntruderNet (<http://cintruder.sf.net/cinet>):

--send-net send resolved captcha to CIntruderNet
--view-net visit distributed online dictionary website

```
Image position : 3
Broken Percent : 100 % [+]
-----
Word suggested : 6
-----

Image position : 4
Broken Percent : 100 % [+]
-----
Word suggested : w

=====
Possible Solution: [ p36w ]
Elapsed OCR time : 1337888061.23
=====
```



Example of "cracking", with unix editor results

Usage

Tracking retrieve captchas directly from url, is usefull for:

```
$ python cintruder -track "http://host/path/captcha.gif" 100
```

- session captchas
- limited dictionary (download all!)

Training apply common OCR techniques to captcha ('train' dictionary):

generate a dictionary to launch brute force attacks:

```
$ python cintruder -train "http://host/path/captcha.gif|file.gif"
```

- train specific modules (ex: easycaptcha, reCaptcha...):

```
$ python cintruder -train "http://host/path/captcha.gif|file.gif" --mod "easy"
```

Usage

Cracking compare captchas with an iconset dictionary:

```
$ python cintruder --crack "http://host/path/captcha.gif"
```

- local dictionary
- distributed dictionary (on the way!)

Sharing send valid "broken" captchas to a common results place:

```
$ python cintruder --crack "http://host/path/captcha.gif|file.gif" --send-net
```

Main idea is to create a good database of broken captchas. We will 'crack' more faster, sharing them.

<https://identi.ca/cintrudernet> - <https://twitter.com/cintrudernet>

Captcha Intruder Distributed Online Dictionary

- #cintruder v0.2 - Distributed Online Dictionary - info: <http://t.co/PKHZLBF5> about 19 hours ago
- #cintruder 69d1172c39a15867cbd5b5b762bb0987 - target: <http://t.co/l8QyvDvm> | word: ygre 27 days ago
- #cintruder dae6a7fb3e1fdf4a09609298280e1b08 - target: <http://t.co/187e39GB> | word: yeye 28 days ago
- #CIntruder v0.2 - Distributed Online Dictionary - info: <http://t.co/PKl4jbFZ> 28 days ago

Ninja scenarios

Security professionals:

- possibility to perform pentesting tests on forms with captchas
- with `--tool` is possible to use cintruder like "proxy"

Example:

Replace suggested word by CIntruder after cracking, on input commands of another tool (ex: XSSer)

```
python cintruder --crack "http://host.com/path/captcha.gif" --tool  
"xsser -u http://host.com/path/param1=foo&param2=bar&txtCaptcha=CINT"
```

Hacktivists:

- perform massive campaigns of web mailing (just spam!)

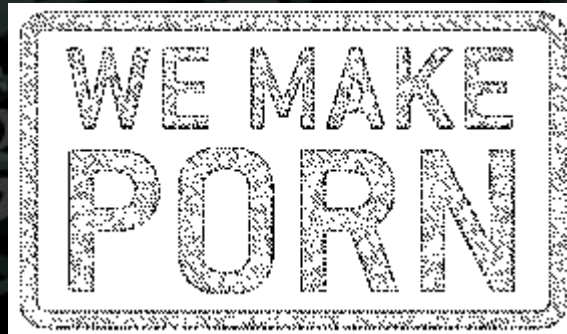
Present → Future

Open tasks:

- research more specific OCR algorithm modules (reCaptcha!?)
- recognize unities after training processes as words
- move that unities to correct folder on dictionary (/iconset/)
- “crack” captchas using distributed online dictionary (Cinet)
- build a GUI and add “drag and drop” options
- still dreaming more ideas ;-)

Mailing list:

cintruder-users@lists.sourceforge.net



OCR processing...

=====

Training Results:

=====

Number of 'words' extracted: 10

Output folder : outputs/words/

2088997d1b4bb6d88a282732ff273eb4.gif 3ca10691389f923acee969dd115c6b96.gif

2a60ce512973d456261ccb353d67d24c.gif 3ca64d561c6c3a2bb9af663ebf00e2ca.gif

34abd1afddca075591750d2880885c96.gif 5645d0807fa06811b94c5cef69a34cb8.gif

Now, move each image to the correct folder on your dictionary: '/iconset/'

=====

Possible Solution: [wemakeporn]

=====

lordepsylon.net

epsylon@riseup.net

0X3CAA25B3

<https://identi.ca/psy>

https://twitter.com/lord_epsylon