Steganography

The art of hiding stuff in stuff so others don't find your stuff

&

A little about my Botnet Stego C&C project

Some information drawn from following articles:

Exploring Steganography:Seeing the Unseen
http://www.jjtc.com/pub/r2026.pdf

With a little from
Lossy Compression Tolerant Steganography
http://nas.takming.edu.tw/chkao/LNCS2001.pdf
Hide and Seek: An Introduction to Steganography
http://www.citi.umich.edu/u/provos/papers/practical.pdf

Definition

- Steganography is the practice of hiding data in other data in an effort to keep 3rd parties from knowing that the intended message is even there
- Encryption's ugly step brother
- It has art aspects since human judgment is involved

Isn’t this security though obscurity?

- Sort of…
- With Encryption alone, 3rd parties may not be able to read the message, but they know one was sent
- In some cases, just being caught sending a message can bring suspicion, or give information to the 3rd party
  - Why is this person hiding something?
  - Crypto laws http://rechten.uvt.nl/koops/cryptolaw/
  - Why all the communication right now?
- Resistant to “Rubber-hose Cryptanalysis”
  Thanks to Marcus J. Ranum for that lovely term

About the 1st article

- “Exploring Steganography: Seeing the Unseen” was published in 1998
- Over the last 12 years, bandwidth and storage have skyrocketed
- 24bit images are common now, as are PNGs that use lossless compression
- Still, the article gives a good intro to the subject which is why I chose it over some newer articles
- The article mostly talks about images, but Steganography can be used in many other places

Historical examples

- Greeks and wax covered tablets
- Histiaeus and the shaved head
- Invisible inks in WWII
- Open coded messages (Pershing example)
- Microdots

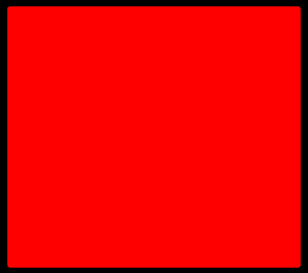
Images

- Information about pixels

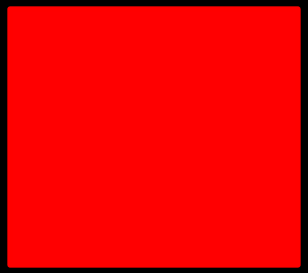
R	G	B
0-255	0-255	0-255
00-FF	00-FF	00-FF
00000000-11111111	00000000-11111111	00000000-11111111

LSB (Least Significant Bit) Encoding

- Can you tell the difference?



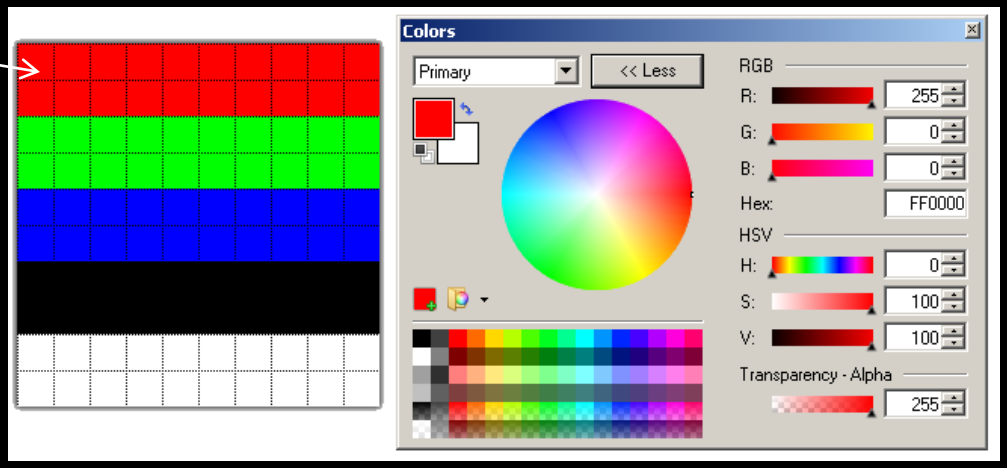
Before Encoding:
255,0,0
FF,00,00
11111111,00000000,00000000



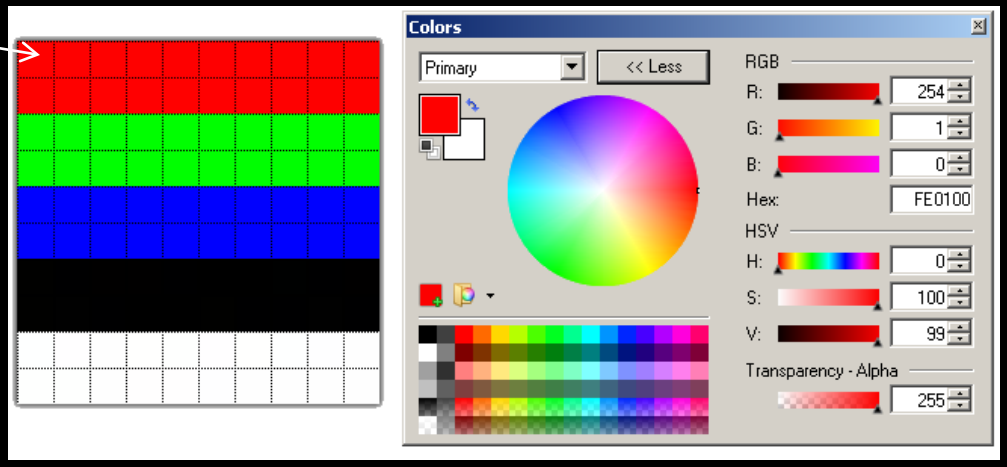
After Encoding “101”:
255,0,1
FF,00,01
11111111,00000001,00000001

Can you tell the difference?

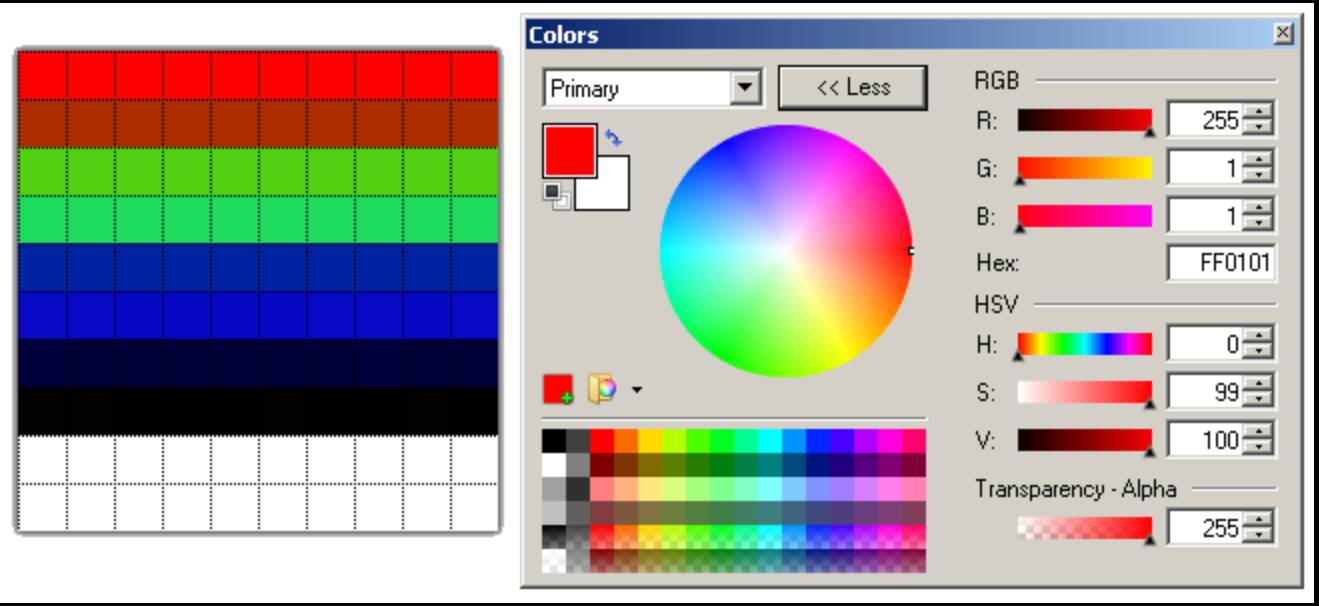
Original



Same file with “I should be able to hold 37 bytes!!!” encoded



Why lossy formats/re-encoding are problems



Wow, that got mangled!!!

JPEGs use a different color space (YCbCr),
stego can be done, but in a different way because of color space
and the use of Discrete Cosine Transform lossy compression

Image resizing/recompressing

- Causes changes in palette and bit order
- May be solvable with redundancy
- Hamming Code
  http://en.wikipedia.org/wiki/Hamming_code
  http://candle.ctit.utwente.nl/Docs/wp5/tel-sys/exercises/datalinkp2p/hamming74demo.html
- James Shewmaker
  https://media.defcon.org/dc-16/video/Defcon16-James_Shewmaker-StegoFS.m4v

Digital Watermarks

- Copyright enforcement
- Redundant pattern encoding to resist data loss during resize/re-encoding
- Change the media enough to kill the watermark, the media degrades beyond the point of usefulness (Think leaked movies)

Detection

- Access to the original image
- Statistical analysis
  (source material category makes a big difference)
- Odd artifacts

Stego Tools

- Since the article is 12 years old, lets look for newer tools:
- Search Sourceforge http://sourceforge.net/search/?type_of_search=soft&words=Steganography
- Steghide (JPEG, BMP, WAV and AU) http://steghide.sourceforge.net/
- Outguess http://www.outguess.org/
- My example code http://www.irongeek.com/i.php?page=security/unicode-and-lsb-stego-code

Other steganography examples

- Truecrypt hidden volumes
  http://www.irongeek.com/i.php?page=videos/truecrypt1

- Office 2007 documents as ZIP archives
  Putting a file inside of a DOCX, it s just a ZIP file with some XML, just add you inserted file name into [Content_Types].xml so the DOCX does not report as corrupted.
  Tacked on to image (copy /B image.jpg+putty.zip test.jpg)
  Slack space
  Alternative Data Streams
  More on these:
  http://www.irongeek.com/i.php?page=videos/anti-forensics-occult-computing

- EXIF or other Metadata

- IP over ICMP or DNS

Text Based Stego

Pros:

- Most “Web 2.0” apps accept text, not necessarily images
- Text takes up little space

Cons:

- Harder to encode and be stealthy
- Less bits to hide in
- In some ways harder to code from a logic standpoint

Pershing Example: Which Character?

The key is knowing what character to pay attention to:

- Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by products, ejecting suets and vegetable oils.
- pershingsailsfromnyjunei
- Pershing sails from NY June 1

Unicode Stego

- 65536 positions in UTF16
- Characters that look similar (homoglyphs) are encoded at multiple positions
- Using these, values can be encoded
  !"$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
- Example:
  Code Point 65 = A
  Code Point 65315 = A

Antonio Alcorn's Work

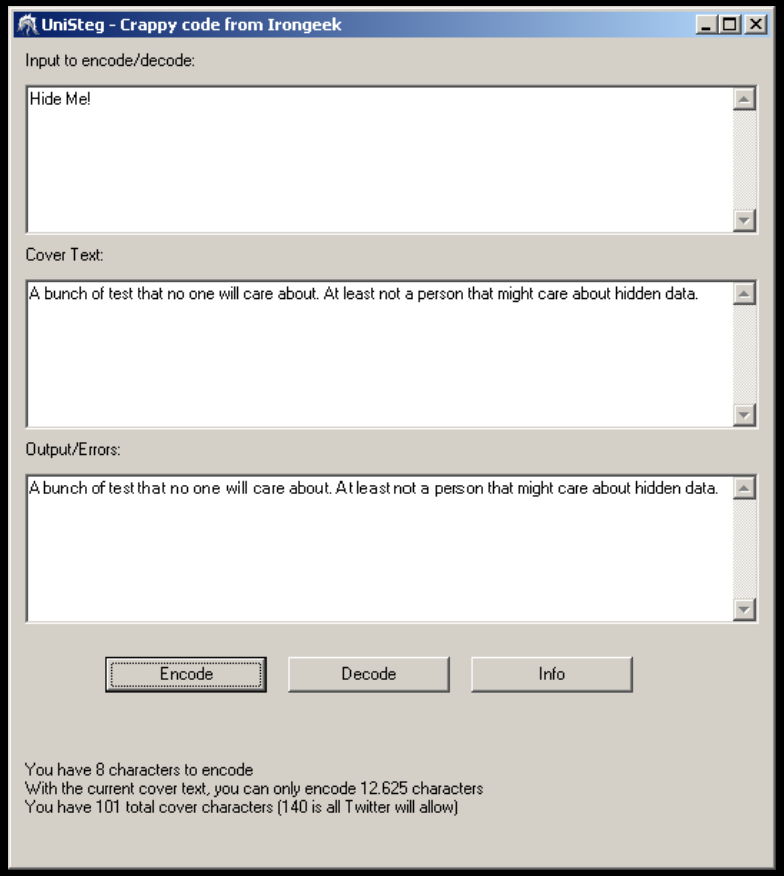
CGI can be found at:
http://www.cs.trincoll.edu/~aalcorn/steganography/encrypt

- "Hello, I need some cover text to use."
- The word "test" is encoded above

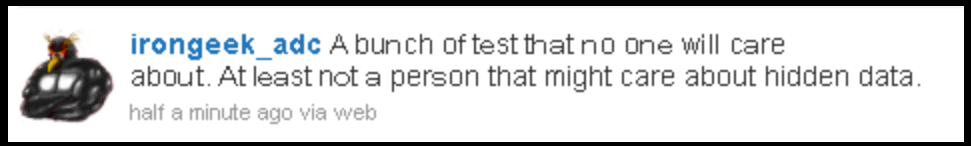
My work on Unicode Stego

http://www.irongeek.com/i.php?page=security/unicode-and-lsb-stego-code

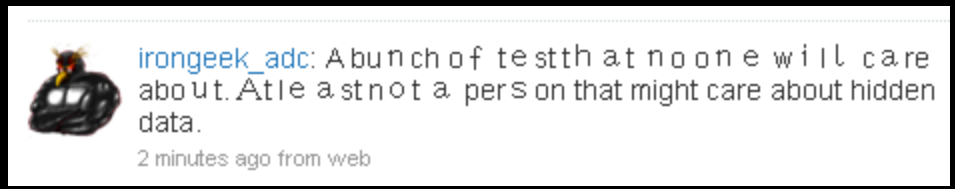
- The Latin alphabet is encoded more than once in Unicode, high values used to represent 1s, lows represent 0s (most characters I could just recode as full width Latin by adding 65248)



Firefox and Twitter:



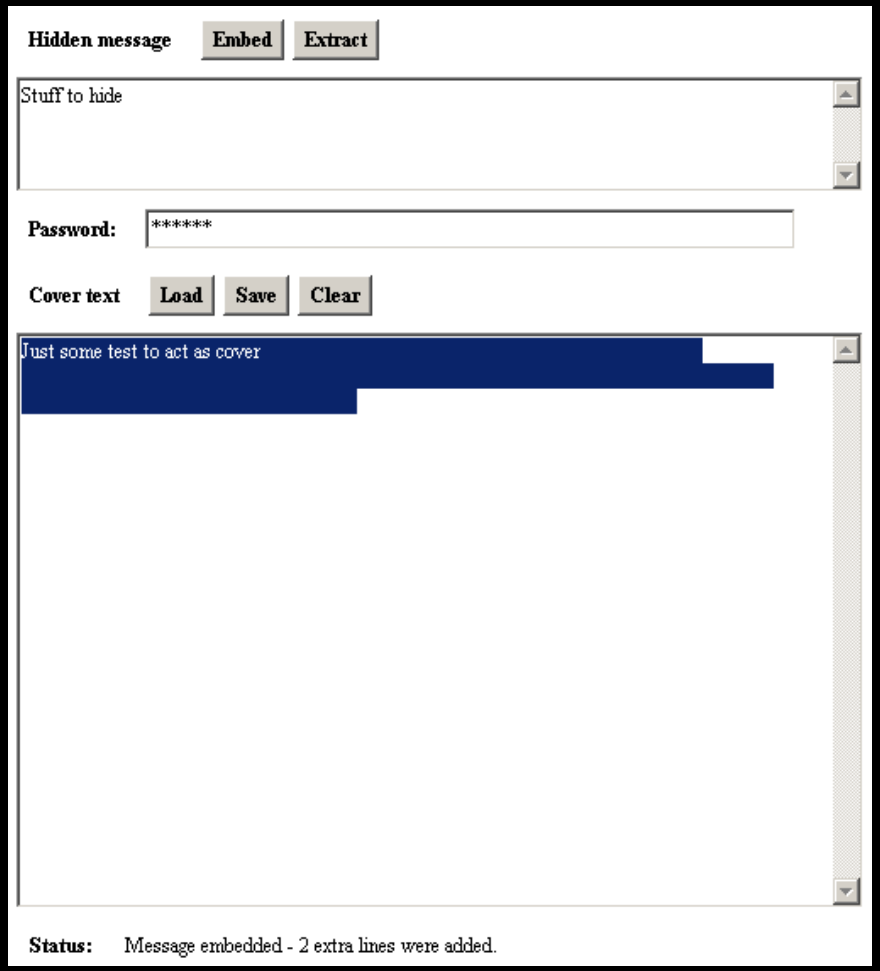
IE and Twitter:



http://Irongeek.com

Snow:White Space Stego

- http://www.darkside.com.au/snow/
- http://fog.misty.com/perry/ccs/snow/jsnowapp/jsnowapp.html



Other Ideas I'm working on

How about a code book for leet/texting/misspellings speak?

Bits	Replace	Transforms
00	s_ | s	z_ | $
01	I | you | e	_i_ | _u_ | 3
10	o | are	0_ | _r_
11	a| why	4 | _y_

I can has cheese burger? How are you?

i can haz chee$e burg3r? How are you? = 01000001 = A

i can has ch3ese burger? H0w r you? = 01011010 = Z

Red are encoded

Blue characters are ones that could have been encoded, but were not needed

Issues:

- Encoder and decoder will be tougher program, but I could do it all in low ASCII.
- I would likely have less room to add data.

More ideas/concepts I've been playing with

- Simplify the language to conserve space
- Give the user a set of control characters they have to integrate into their writing (Punctuation)
  - "test" becomes ",&:!,',&"
  - User adds word to the Punctuation to make it make sense:

    Hi, Robin & I have been working on botnets:stegofun! Progress is slow, it's taking a long time, it is time consuming & frustrating

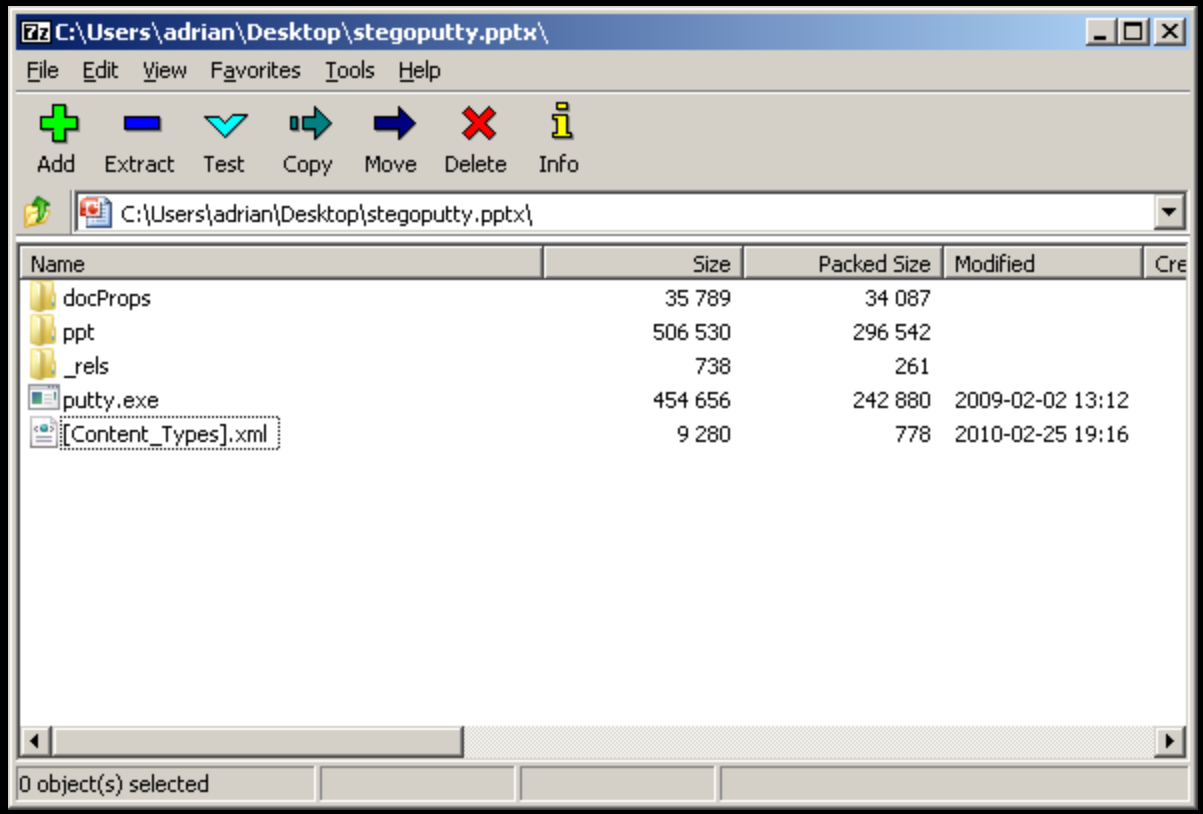
  - Could encode most common letters as one symbol, but that would break if crypto were used
- Trade off between frequency of character (more data can be hidden) and ease of writing cover text (Vanna White Problem)

Punctuation Encoding Lookup Table

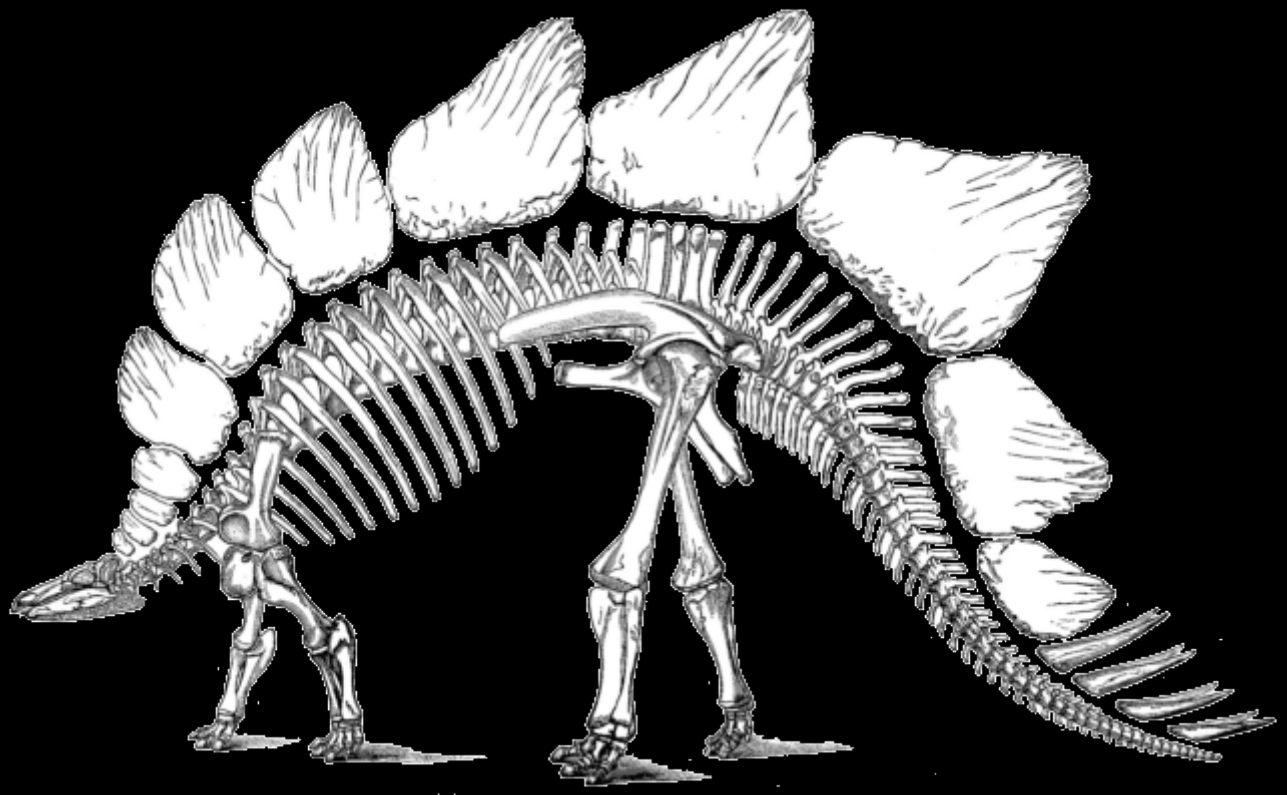
0000 = ;
0001 = ?
0010 = .
0011 = '
0100 = &
0101 = !
0110 = :
0111 = ,
1000 = $
1001 = -
1010 = #
1011 = =
1100 = %
1101 = *
1110 = +
1111 = @

Send a Zip file as an Office doc?

- Upload to Google Docs
- Email to an account that the other end checks



SnarlBot Project

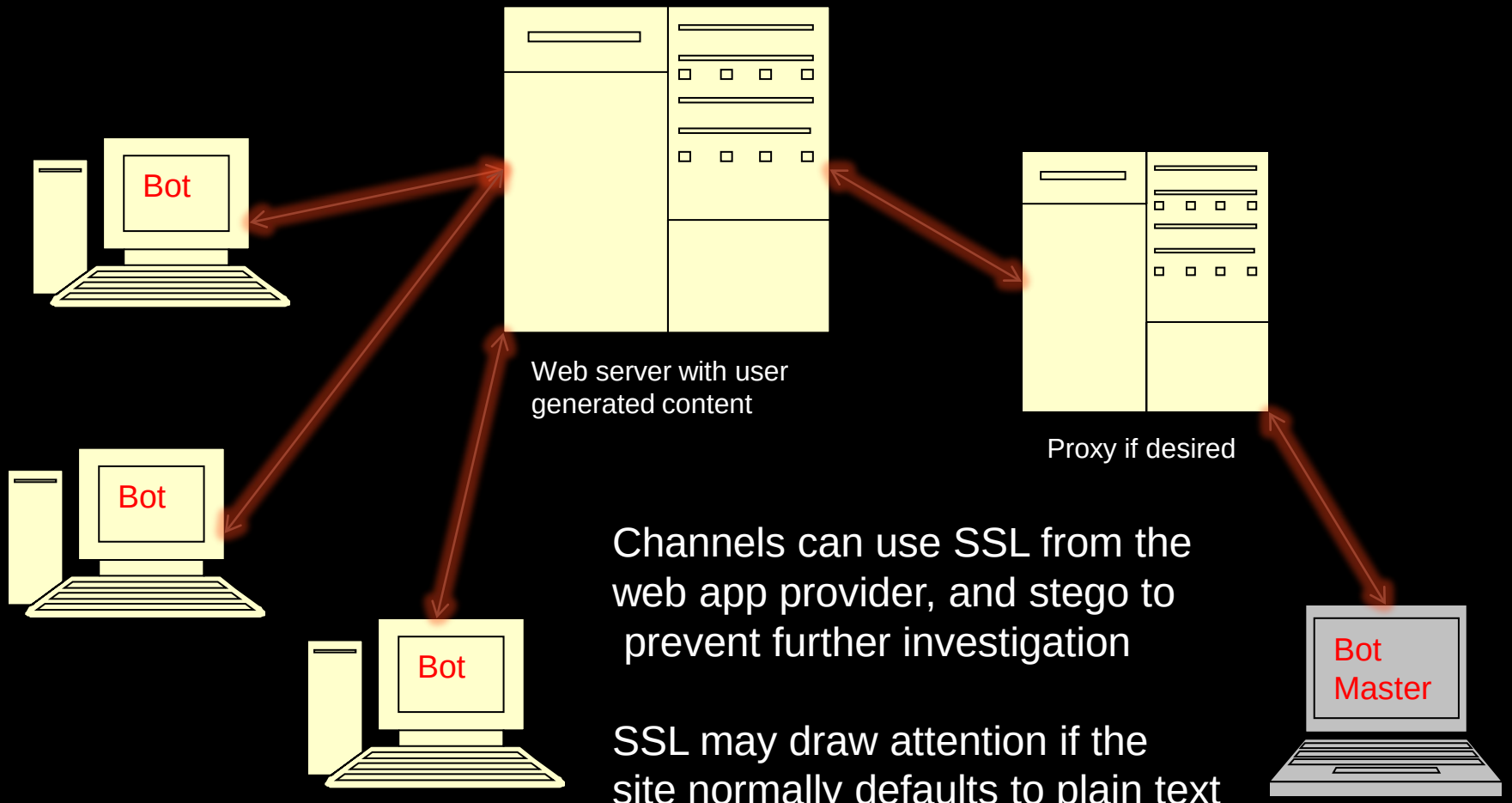


I chose the name because I’m a dork who was a kid in 1984. Figure it out. ☺

SnarlBot

- A simple botnet that uses Social Media/Web 2.0 web apps for “blind drops” as part of the command and control channel
- Content at the blind drops use Steganography so it’s not obviously a botnet doing the communicating

Topology



Channels can use SSL from the web app provider, and stego to prevent further investigation

SSL may draw attention if the site normally defaults to plain text

This schemes advantages

- The blind drop obfuscates who is controlling the botnet
- Proxies can be used for web traffic to further obfuscate the identity of the bot herder
- Steganography plus encryption makes the channel hard to detect
- Social web sites like Twitter or Facebook are not as likely to be blocked as IRC or P2P
- SSL support for the C&C provided by the web host of the blind drop

Disadvantages

- More data has to be sent to get a message though
- The more complicated something becomes, the more bugs it will have
- May have to simplify the C&C commands
  - Use single byte command: “a” for attack
  - IPv4 addresses can be expressed in 4 bytes
  - This make the Steganography less adaptable, but more meaning can be encoded in less bytes

Isn’t this a little black hat? Other uses?

- Yep, it’s a little black hat, but who’s to say someone is not already doing it?
- This could start research on how it can be detected.
- May have applications for privacy providing darknets like I2P or Tor http://www.irongeek.com/i.php?page=videos/darknets-i2p-tor-phreaknic

Similar Project

- Robin Wood's KreiosC2
  http://www.digininja.org/kreiosc2/index.php

  Does not use stego yet, but should be easy to add for someone that knows Ruby

Conclusions/Questions

- Other Steganography techniques?
- Usefulness?
- Detection?
- Other uses for research?