# Mimicry: An Introduction

By Mystic

# What is Mimicry?

- The ability to survive by mimicking surroundings

# What is Mimicry?

- The ability to survive by mimicking surroundings
- Visibility of encryption

# What is Mimicry?

- The ability to survive by mimicking surroundings
- Visibility of encryption
- Grammar filtering to find encrypted data

# Peter Wayner's Mimic Functions

- A way to encrypt/hide data in which the output is statistically and grammatically sound

# Peter Wayner's Mimic Functions

- A way to encrypt/hide data in which the output is statistically and grammatically sound
- Generates text using the syntax described in a Context Free Grammar (CFG) and hides data by the choices it makes

# Context Free Grammars

- A way of describing language

# Context Free Grammars

- A way of describing language
- Uses
  - Terminals: Words and phrases that are static

# Context Free Grammars

- A way of describing language
- Uses
  - Terminals: Words and phrases that are static
  - Variables: Places where decisions can be made

# Context Free Grammars

- Productions: describes how a variable can be converted into different sets of variables or terminals

# Context Free Grammars

- Productions: describes how a variable can be converted into different sets of variables or terminals
- Example of a production:

  **variable** ---> phrase || words

# Context Free Grammar

- Example of a CFG:

**Start** ---> **noun** **verb**

**noun** ---> Fred | | Barney | | Fred and Barney

**verb** ---> went fishing. | | went bowling.

# How Data is Hidden

- Mimic generates a binary tree based on the possibilities in a context free grammar

# How Data is Hidden

- Mimic generates a binary tree based on the possibilities in a context free grammar
- It then chooses the leaves that would encode the right bits

# How Data is Hidden

- Given the fallowing CFG:

**Start ---> noun verb**

**noun --->** Fred | | Barney

**verb --->** went fishing **where** | | went bowling **where**

**where --->** in **direction** Iowa. | | in **direction** Minnesota.

**direction --->** northern | | southern

# How Data is Hidden

- Bits to be hidden: 1010

**Start ---> noun verb**
**noun --->** Fred | | Barney
**verb --->** went fishing **where** | | went bowling **where**
**where --->** in **direction** Iowa. | | in **direction** Minnesota.
**direction --->** northern | | southern

| Step | Answer in Progress | Bit Hidden | Production Choice |
|---|---|---|---|
| 1 | **Start** | *none* | **Start ---> noun verb** |
| 2 | **noun verb** | 1 | **noun --->** Barney |
| 3 | Barney **verb** | 0 | **verb --->** went fishing **where** |
| 4 | Barney went fishing **where** | 1 | **where --->** in **direction** Minnesota. |
| 5 | Barney went fishing in **direction** | 0 | **direction** ---> northern Minnesota |

**Final sentence: Barney went fishing in northern Minnesota**

# Mimic Grammar File

- Variables: Start with an asterisk and must be one word

# Mimic Grammar File

- Variables: Start with an asterisk and must be one word
- Productions are separated by numbers between forward slashes

# Mimic Grammar File

- Variables: Start with an asterisk and must be one word
- Productions are separated by numbers between forward slashes
- The numbers indicate the weight given to the production (the higher the number the more probable it will occur in the production)

# Mimic Grammar File

- Variables: Start with an asterisk and must be one word
- Productions are separated by numbers between forward slashes
- The numbers indicate the weight given to the production (the higher the number the more probable it will occur in the production)
- The end of a variable is indicated by double slashes

# Mimic Grammar File

- The starting variable is the variable that is alphabetically first

# Mimic Grammar File

- The starting variable is the variable that is alphabetically first
- Example:

  *AAStart = Fred went to *con /.1/ Barney went to *con /.1//

  *con = Defcon /.1/ Black Hat /.01//

# Mimic Grammar File

- Grammars can not be ambiguous, meaning there must be only one way of producing any given phrase (sentence)

# Mimic Grammar File

- Grammars can not be ambiguous, meaning there must be only one way of producing any given phrase (sentence)
- Example of an ambiguous grammar:

  *AAStart = *first *second//

  *first = love /.5/ love and /.5//

  *second = death /.5/ and death /.5//

# Mimic Grammar File

- To avoid ambiguous grammars, grammars must be in Greibach Normal Form (GNF) meaning variables must be at the end of productions

# Limitations of Mimic Functions

- Can create huge files from small amounts of text, depends on the number of possibilities in the grammar

# Limitations of Mimic Functions

- Can create huge files from small amounts of text, depends on the number of possibilities in the grammar
- Output starts repeating

# Limitations of Mimic Functions

- Can create huge files from small amounts of text, depends on the number of possibilities in the grammar
- Output starts repeating
- Randomness is built in, but not very sophisticated

# ircMimic

- Mimicry in a live chat environment

# ircMimic

- Mimicry in a live chat environment
- Sends a Mimic encoded message in the form of one side of an IRC conversation

# ircMimic

- Mimicry in a live chat environment
- Sends a Mimic encoded message in the form of one side of an IRC conversation
- ircMimic Grammar File
  - Added “new line” character
  - Maximized possibilities

# ircMimic

- Going further with ircMimic
  - Have two bots talk with each other in order to send a message

# ircMimic

- Going further with ircMimic
  - Have two bots talk with each other in order to send a message
  - Add more to the grammar

# Resources + Links

- Mimic
  - http://www.wayner.org/texts/mimic/
  - http://www.spammimic.com
  - http://www.owlriver.com/test/disk2/
- Disappearing Cryptography
  - http://www.wayner.org/books/discrypt2/
- ircMimic
  - http://code.lostways.com/ircmimic/

# Resources + Links

- TVSG
  - http://www.tvsg.org/