Enhanced Covert Channel Techniques for Twitter

...

Kotaiba Alachkar and Peter Prjevara

...

Offensive Technologies Project

Agenda

- Our motivation
- Existing techniques
- Our solution
- Demo
- Summary
- Future work

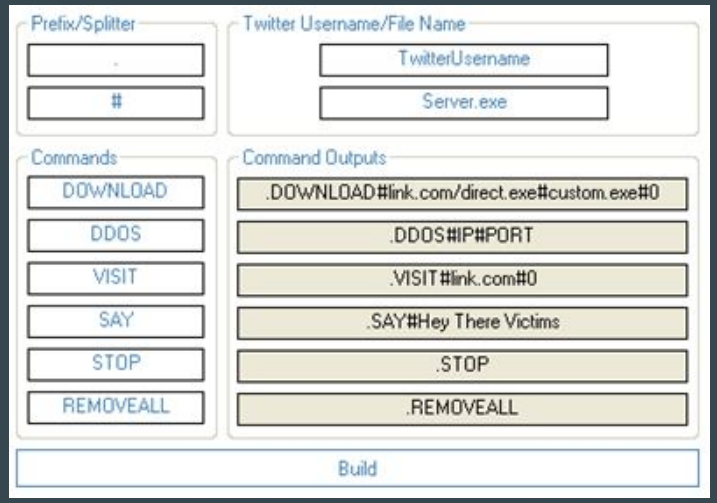
Why to use Social Media for C&C?

- Convenient, reliable infrastructure
- Handy
- Secure: HTTPS is standard

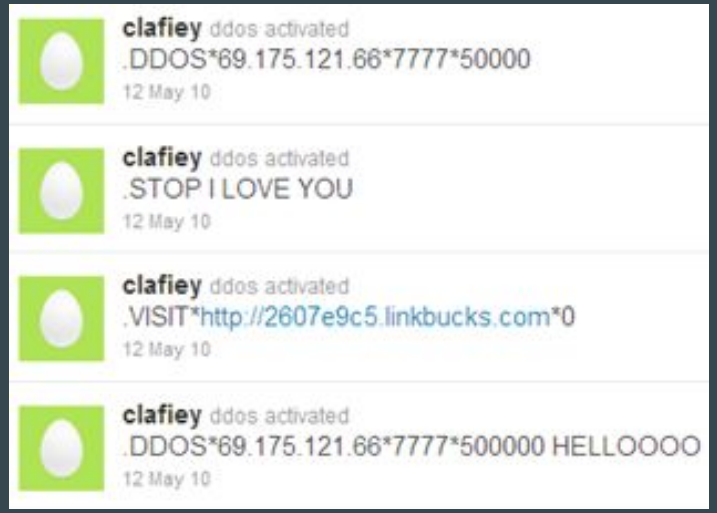
...

"Defenders of corporate networks are unlikely to notice the offending traffic in the large volumes of other Internet-bound sessions" - Zeltser

Existing Techniques 1 - Totally Obvious



TwitterNET Builder



Example

Existing Techniques 2 - Better Obfuscation

- API dependent
- No payload
- (re)tweet or like
    - must include one of the commands to be executed

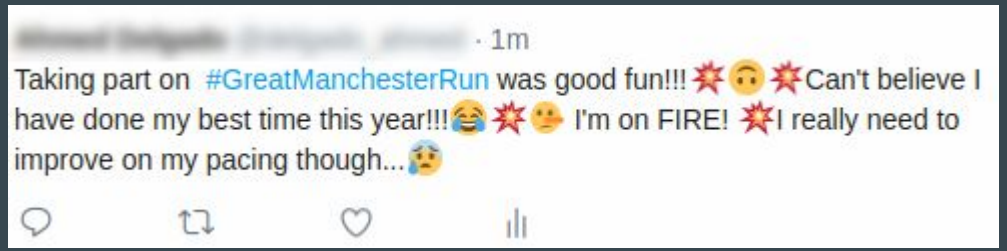
Existing Techniques - Debatable?

- Command hidden in PNG
- Better solution, but still lacks full anonymity
- Twitter is compressing images immediately after upload

Figure 6. Twitter command and control

Improved Solution - Our Goals

- Dynamic command - #hashtag allocation
- Tweets with normal looking content
- Possibility to include payload (IP address etc.)
- Eliminate false + / - even when fully anonymous



Configurable Parameters

Parameters	Description
timeInterval	Trend's time interval (e.g. 1 for the past hour)
isDynamicLocation	Trend's country of origin based on tweet location
countryName	Trend's country of origin (statically)
userName	Twitter account username
searchQueries	Advanced search query (including AND, OR, and NOT)
commandsList	List of commands with its own trend mapping
FetchTime	How often daemon fetch for tweets and trends

Solution 1 - API Dependent

Pro's

Easy to code

Less detectable in a corporate env.

Con's

Developer credentials - phone number, access tokens...

Limited amount of API calls

Special Trend polling algorithm
False Negatives++

Slow

Fixed Trend location / time

Solution 2 - API Independent

Pro's

Semi / Fully Anonymous

Dynamic / Static Location and Time

Known Trend source code
False Negatives--

No rate limitation

Fast

Con's

Harder to code

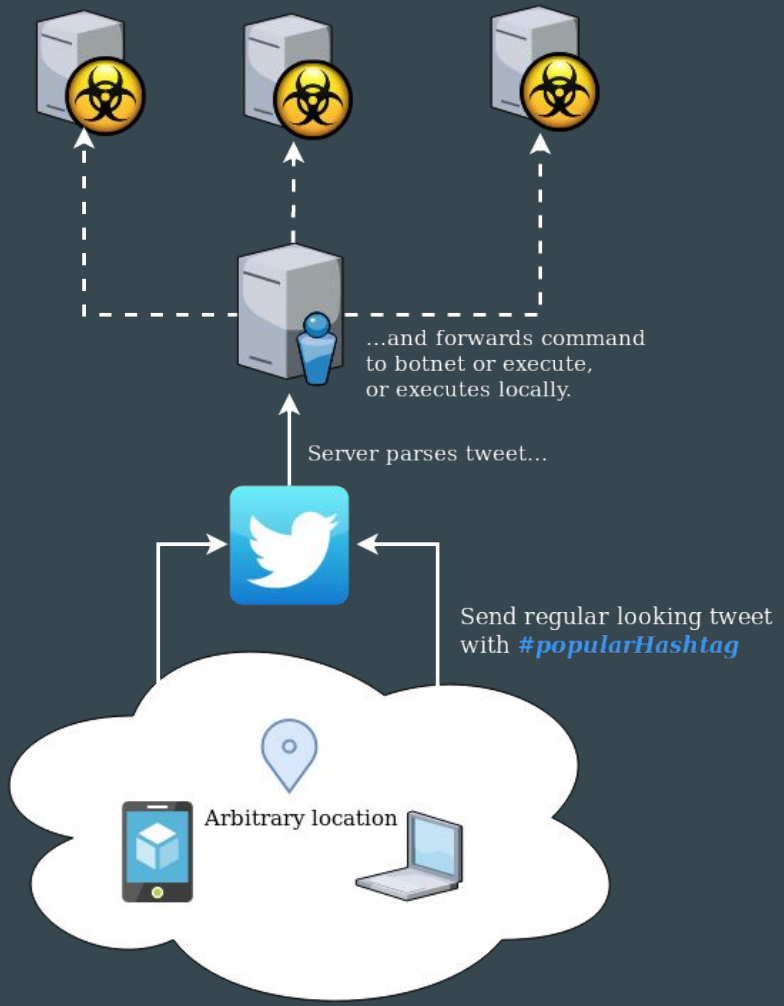
Corporate env.

Our Encoding Scheme

- Dynamic #hashtag to command allocation
- Every hour with configurable T offset
- IP to Emoji converter
  - Created our own numeral system
  - Allocation of emojis / numbers is randomized based on selected command

Demo

1. Choose a command:
    a. echo / ping / nslookup
2. Provide us your IP - :)
3. Add otProject on telegram

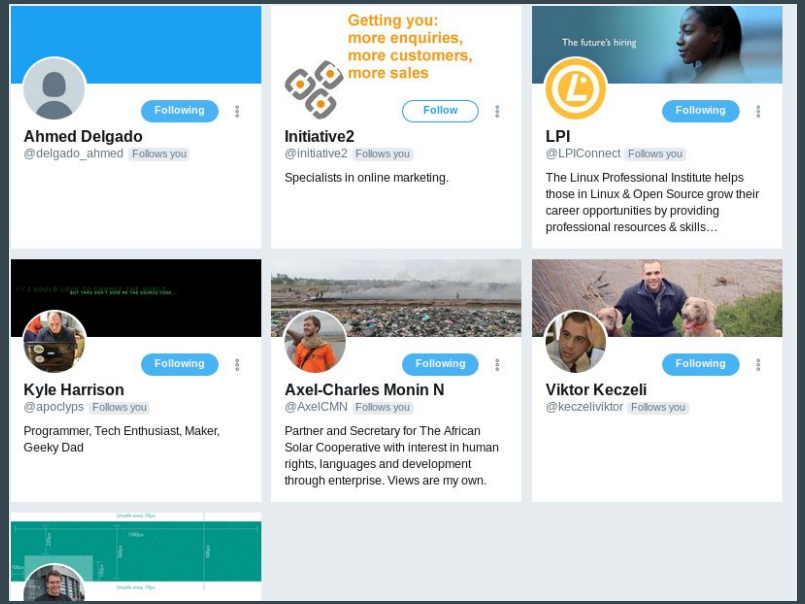

...and forwards command
to botnet or execute,
or executes locally.
Server parses tweet...
Send regular looking tweet
with #popularHashtag
Arbitrary location

Verification of Success

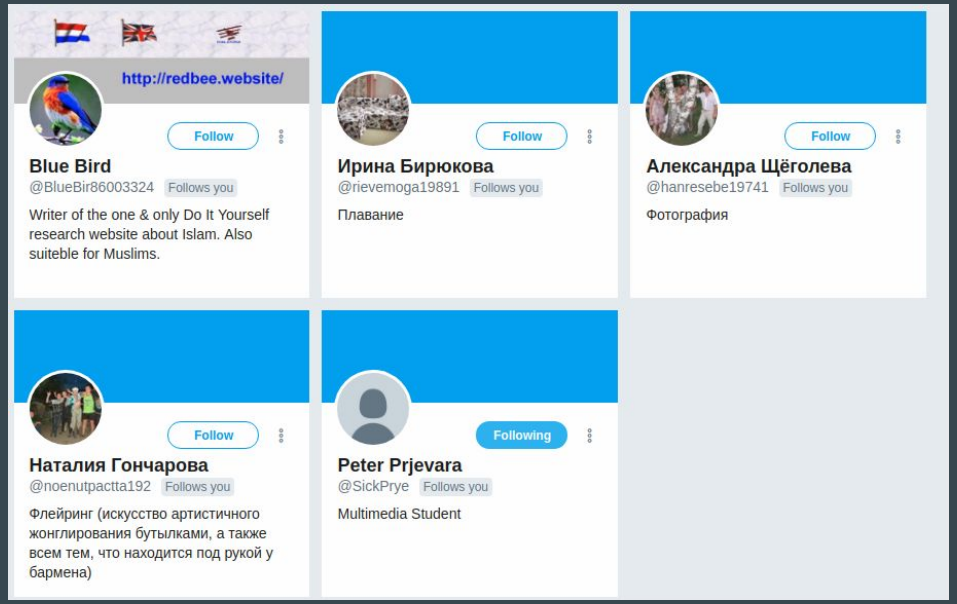
- Design experiments to measure detectability of
  a. Positive tweets that contain commands
     - By chance only = UNDETECTED
  b. Malicious Twitter accounts

Experiment B

My followers



Ahmed's followers



Summary of Enhancements

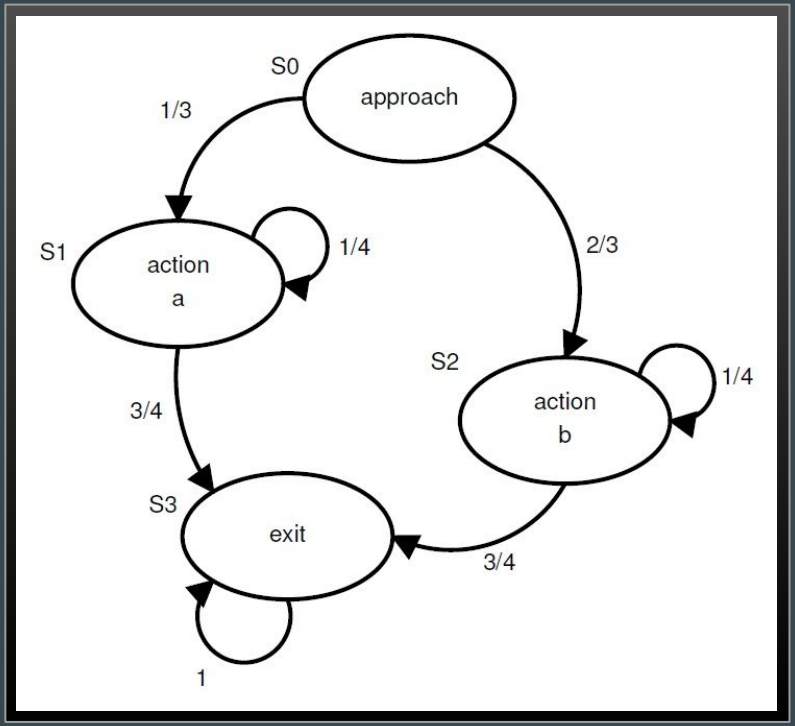
- No API access allows for anonymity + no rate limits
- Dynamic / Static Location and Time
- Known Trend source = **False Negatives--**
- Benign looking tweets
- Faster deployment
- Complex command structures possible

Possible Mitigation Techniques

- Corporate Environment
  - Monitor for suspicious HTTP calls
  - Don’t allow Twitter API if not necessary
- Botnets
  - Same as above
  - Difficult, as not many people monitor home network

Future Work

- IPv6 encoding scheme
- Fix issue with spaces in Trends
- Complex command structures
  - Finite State Machines allow for implementing complex command structures



Questions

Source code available at our github repositories:

- *https://github.com/kalachkar/twitter-CandC*