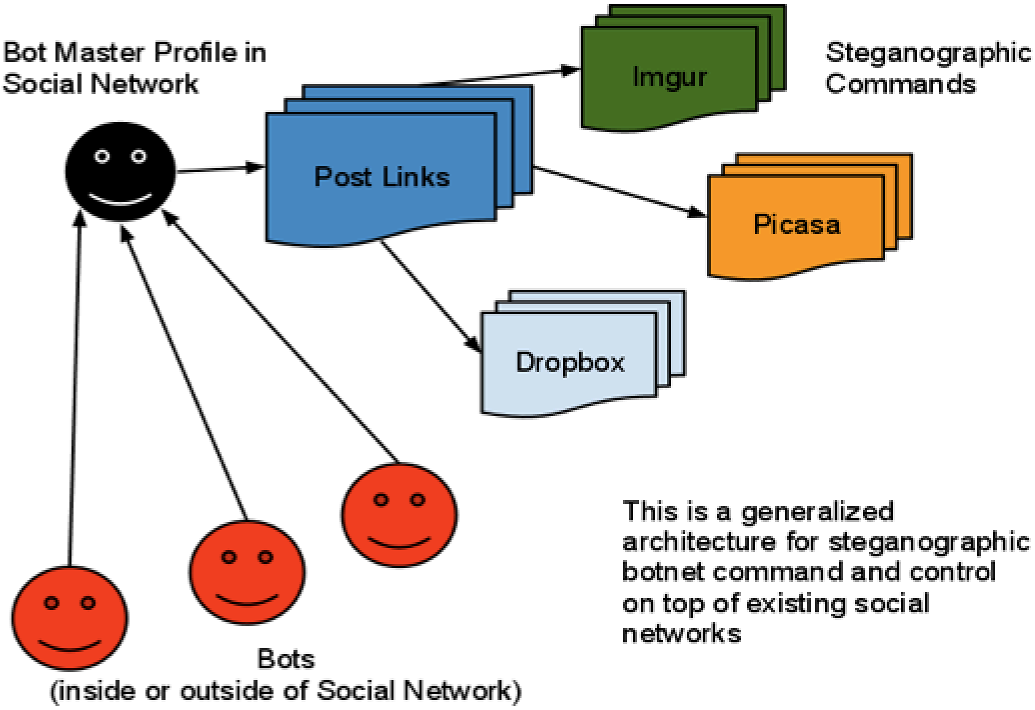
CS 577
Resilient Facebot
Prof. Dietrich
Proposal Due October 13, 2011

Orie Steele, Tom Parisi, Simon Sidhom, Ken Bodzak

1. **Abstract**

In this paper we propose an architecture for steganographic botnet command and control leveraging social networks and media sharing services. Our general approach will be social network agnostic, but we plan on developing a few social network specific examples as proofs of concept. Our areas of focus will include: methods and mediums for steganographic communication in social networks, social network specific botnet network topologies, and detection and prevention methods.


Bot Master Profile in Social Network
Post Links
Imgur
Steganographic Commands
Picasa
Dropbox
Bots
(inside or outside of Social Network)

This is a generalized architecture for steganographic botnet command and control on top of existing social networks

2. **Motivation**

Social Networks are growing in size and complexity. As traffic and activity increase, so does the potential for covert communication. Duplicated or reshared material in social networks (memes for example) provide an excellent medium for covert communication as the activity is likely to appear uninteresting to the casual observer. Social Networks also provide an excellent distribution platform for malicious code, legitimate code, and ideas. With the increasing role of social networks in protests and acts of civil disobedience, there is a lot of potential for voluntary botnets coordinated through social networks.

3. Approach

In order to operate the assumption that a certain number of computers will be compromised with some type of malware will be made. Therefore, creating the attack vector as well as the the infecting malware will be out of this scope. This will allow the bulk of attention to be spent on the creation of the botnet as well as the prevention. All of the infected nodes will connect to the Facebook webserver through the internet. Of course, this will not be the actual Facebook, but a simulation set up in DETER. The bots will not know of each other. This means that the structure of the botnet is a Command and Control server setup as opposed to a peer to peer setup. The normal concern is that if the central server goes down, the botnet will be lost. This is mitigated by the fact that Facebook plans for this type of situation in regards to a down server. If a Facebook server goes down, the website will lose money and users. Therefore, Facebook has solved this problem and it would be best to use their implementation and their servers without them knowing. Facebook has also has load balancing techniques that will be used in our favor as well. The bots will be checking a list of command profiles at regular intervals. This is analogous to the peer to peer setup in reference to a down node, or in this case a removed profile. There will be a list of profiles to check in case of a removed profile. Strategically, more than one profile should be kept so that the botnet is never lost.

Usually when a botnet is created, there is a level of encryption of the communication. If the traffic of the bots is monitored, it would be fairly obvious that something is wrong. This is because Command and Control servers are usually obscure and thus easily detectable. However, if traffic to Facebook is discovered, it will look like normal traffic. If the packets are examined, it becomes even more likely that the obscure server is using encryption to talk to all its visitors. Steganography is better because the data is not encrypted. Instead it is only hidden. This means that the packet inspections would look like just normal Facebook traffic making it almost invisible to unwanted eyes.

To code the bot net, python will be used. This is because python has a lot of libraries that will help with the coding of the bot software. These libraries will make it easier to make the bots do what is desired. Along with the language, there are some steganography tools that will be used to hide the communication to the bots. These tools will have to be tested to see which one operates the best with the least amount of consumption of CPU power. This is important because the communications need to stay secret from the users of the bots. The basic information flow is as follows:

1. The bots are infected somehow.
2. The bots will visit the command profile of Facebook.
3. Steganography will be used to relay commands to the bots
4. The bots will carry out the commands.

The FaceBot can be stopped in different ways. First, the IP addresses of all the viewings over a period of time should be correlated. The correlation should show how many times an IP address visited a profile and at one time. These should be checked against normal human behavior. If the amount far exceeds this or if the visits are at regular intervals, then the profile should be shut down because there is something not right going on. This type of correlation would show that something other than humans are visiting the profiles. If the bots are checking into the command profile at regular intervals, there would be traffic to the profile even at night and at other odd times. Another way to stop this kind of bot is to scrub all of the pictures and other information that is put onto social networking sites. This way there is no way that steganography can be used to communicate to other users.

4. **Existing Work**

Botnets are not new to social network. There have already been Facebook legitimate application with malicious components, for example [1] presents an application which acts as Trojan Horse. Every time a user clicks in the trojan application, the browser would download images in the background from another website. With enough use, this technique could be used to launch distributed denial of service attacks. [2] is essentially exactly what we are proposing to do.

Antisocial Networks: Turning a Social Network into a Botnet [1] proposes the creation of distributed systems using social networking websites. These systems can be exploited and manipulated to carry out actions and malicious attacks. This paper lays out a proof of concept for taking a social network and converting it into a capable attack platform.

A second paper, *Stegobot: a covert social network botnet* [2] , discusses a covert botnet that communicates data and control information over a social networking service. This eliminates a need for creating new channels of communication between nodes and allows the network to limit its visibility. The actual messages and communication sent over a social network would on the other hand, have a huge amount of visibility. To combat this the paper uses steganography to hide its communication within images. This allows the botnet to spread information at reasonable rates while retaining a level of undetectability.

5. **Deliverables**

The deliverables will be a Facebook bot net complete with the code that makes the bot net run. We will also deliver strategies to stop a Facebook botnet from occurring.

6. **Division of Labor**

Coding will be done in a team programming style. We will at least design the program together in the same room and then split up the modules.

There will be many opportunities for relevant research which may or may not be implementable in the time frame provided. We plan to track relevant topics, and provide channel for crowd sourced contribution to the project.

7. **Timeline**

Planning

a. Proposal: 10/3
b. Presentation: 10/13

Develop Botnets and Network topologies

a. Abstract setup : 10/27
b. Integrate with Popular Social Networks 11/3
c. Combine above approaches, build some toys 11/10

Analyze, Detect, and Defend

a. Implement our solutions of defending: 11/24
b. Analyze the results: 11/28
c. Further implementation:12/1

Paper and Presentation

a. Paper: 12/8
b. Presentation: 12/19

8. **References**

[1] http://i.zdnet.com/blogs/facebotisc08.pdf
[2] https://netfiles.uiuc.edu/ahouman2/www/papers/IH11-Stegobot.pdf