Proposal for Interdisciplinary Research

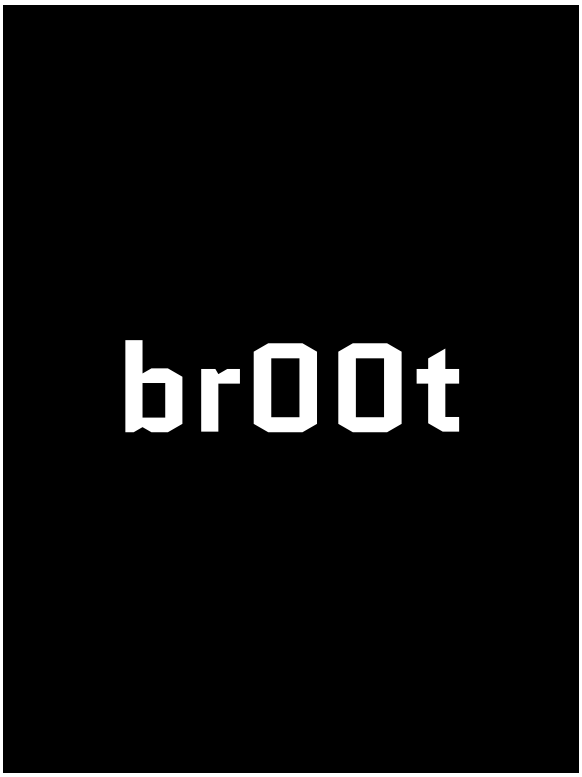
Fordham University, Rose Hill

Students in the Departments of:

Computer and Information Science

Department of Mathematics

Department of Economics



Contact: Aaron DeVera, adevera@fordham.edu

Last Edit: Thursday, January 22, 2015¶

Project Summary

The purpose of this project is to develop a web server capable of harnessing the computing power of the clients connected to its webpage in order to crack passwords. To accomplish this, the project will utilize concepts in the following areas of computing sciences, mathematics, and economics: brute-force password cracking, combination algorithms, distributed computing and parallel processing, and optimization. The project will develop upon the concepts found in botnets, a type of infiltration/command & control where an adversary launches attacks through a network of computers that have already been hacked as a platform for new attacks. This project will be fundamentally interdisciplinary, and will be one of the few known instances of *cloud-distributed password cracking*. It will be demonstrative of exceptional programming and mathematical capabilities. It will also be demonstrative of the potential threats the average internet use may face on websites that seem "harmless" or "friendly".

Project Background

The first implementation of the preceding specification was deployed at MHacks 2014, the largest University hackathon in the United States and hosted by University of Michigan. Upon invite to compete in the hackathon, the above specification was programmed and deployed in the last 4 hours of the competition, and entered under the title *br00t*, a nod to the brute-force password cracking concepts it utilized. The cloud-distributed approach to password cracking was praised by several security-minded engineers judging the hackathon.

The original *br00t* project contained a Node.js web server written in Javascript, along with a client-facing website written in standard HTML5/CSS/JS. It quickly cracked passwords with a simple 26-character variability (lowercase alphabet) of up to 8 characters. The project was deployed on publicly addressable web servers.

Technical Walkthrough

The *br00t* server broadcasts a webpage intentionally tailored to look "friendly", and to encourage the people visiting the page—clients—to stay on the page as long as possible. Once a client visits the *br00t* webpage, the *br00t* server indexes the client. The moment at least one client is visiting the page, the *br00t* server begins sending the client's computer parts of an algorithm that can crack a password.

The password-cracking algorithm is brute-force, meaning that it assembles thousands of combinations of alphanumeric characters, and tries each combination against the machine it is breaking into. In conventional computing this may take a long while, depending on how complicated the password is. This process can be sped up with distributed computing.

If multiple clients are visiting the *br00t* webpage, then the *br00t* server divides the work of the algorithm amongst the number of clients. This "chunking" of the cracking algorithm can be divided up against how many current clients there are, speeding up the amount of time it takes to crack the target password.

If a client leaves the webpage, the *br00t* server must be able to redistribute the remaining combinations to try amongst the remaining clients. Upon a succesfful cracking of a password, the *br00t* server will have read/write access of the target computer.

Current Proposal

There are several improvements that can be made to the *br00t* server that can strengthen its adversarial capability and capability to distribute the cracking algorithm amongst different client-owned computers. This project will set out to build a *br00t* server with the following design considerations and project operations.

- Develop the web server in Python, using packages like Flask. Python is a much more intrinsic to program, and much more powerful on the OS level than Javascript. It will be much easier to write the cracking algorithm in Python considering the mathematical applications of the language. Python packages like Flask can provide the web server capability for the project.
- Engineer a password-cracking algorithm that can crack passwords in short or long length, with alphanumeric characters, capitalized or not.
- Optimize the *br00t* server to dynamically "chunk" the algorithm amongst its clients, and be able to redistribute workloads in real-time
- Configure the *br00t* server with SSH capabilities, so that the server can attack and crack our own real dummy servers over the web.

Project Requirements

Python programming, TCP/IP/HTTP networking, parallel processing, algorithm engineering, combination mathematics, optimization, resource management, A/B testing, webserver space, and web design.

Project Deadlines

This project is open to any student on the Fordham Rose Hill campus who wishes to conduct research in the disciplines of computer science, mathematics, and economics, upon the acceptance of they application to the project.¶

Team Dynamic

Project team members should be available to meet as a group or in person at least twice a week for a minimum of one hour per session. They should also be open to the use of collaborative project management tools such as GitHub, Slack, and be responsive to team updates.

Project Deadlines

The timeline for this project is from now to the deadline for the Undergraduate Research Symposium. Abstracts are due March 16, 2015.

If we desire funding, a grant proposal would be due February 6, 2015.