The Long and Short of Passwords

Rich Shay

November 5, 2009

Outline

Motivating the Motivation

- Numerous difficulties with text-based passwords
- Passwords can be cracked
- Complex passwords difficult to remember
- Users often inundated with many passwords
- So why do we use (and study) passwords?

Passwords Enable Authentication

- Systems need to authenticate users
  - Servers, ATMs, email, and many other computer services
  - More and more websites enable users to create accounts
- Passwords enable authentication
  - If I know my password, and no one else does,
  - and system sees someone attempting to log in with my name and password,
  - then that system can be *reasonably* certain the user is I

Advantages of Passwords

- Passwords already established and accepted
- People already familiar with using passwords
- Passwords require no added input hardware (unlike biometrics)
- Passwords can be used over a terminal like SSH (unlike graphical schemes)
- No extra devices to carry around (unlike ID cards)

Mandatory Doom and Gloom Slide

- Passwords often the only protection against intruder [Kuo *et al.*, 2006, Summers & Bosworth, 2004]
- A single user becoming compromised can lead to an entire system becoming compromised [Bishop & Klein, 1995]
- Unless policy says otherwise, users tend to create very simple passwords [Bishop & Klein, 1995, Proctor *et al.*, 2002, Leyden, 2003]
- Password policy can impact financial health of an organization [Robert M. Polstra, 2005]

Outline

How do Passwords Fail?

- What is a password?
  - A short string of characters
- What is a password used for?
  - To authenticate a user
- On what assumptions does password depend?
  - User knows his or her password
  - *Only* user knows his or her password
- How do these assumptions fail?
  - User forgets (or fails to memorize) password
  - Someone else learns password

Outline

Brute-Force Attacks

- Brute-force attack consists of attacker trying different potential passwords until one works
- Shorter password more vulnerable than longer password
- Password with only letters more vulnerable than password with other characters

Brute-Force Example

- Consider password with 6 lower-case letters
  - Assuming all possible combinations equally likely, random guess has probability of 26^{-6}, or one in 300 million
- Consider password with 8 characters, using numbers and lower-case letters
  - Assuming all possible combinations equally likely, random guess has a probability of 36^{-8}, or one in 2.8 trillion

Dictionary Attacks

- Attacker cracks password by trying every word in dictionary
- English dictionaries readily available
  - On your Mac, check out */usr/share/dict/web2*
- Cracking dictionaries exist
  - Check out *http://www.openwall.com/wordlists/*
- Can combine words, and try modified words

Social Engineering

- Password length and complexity offer no protection
- In a 2003 study, 90% of users willing to divulge password for pen [Leyden, 2003]
- Recent phishing attack obtained 10,000 Hotmail passwords

Outline

Introducing Entropy

- What does it mean for text to be “complicated”?
- Claude Shannon answered this in the 1940s and 1950s
- Wikipedia page has lots of useful information
  - `wikipedia.org/wiki/Entropy_(information_theory)`
- If you want a more in-depths understanding, I recommend
  - Shannon, C.E.: Prediction and entropy of printed English. Bell Systems Technical Journal (1951)

Entropy, explained by Wikipedia[Wikipedia, a]

- Quantifies in bits the amount of information per character
  - Or, amount of information lost if character removed
- A fair coin has an entropy of one bit
- $H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i)$
  - log base is often 2, to explain result in bits
  - $p(x_i)$ is the probability that X equals x_i
  - When $p(x_i)$ is zero, $p(x_i) log_2 p(x_i)$ is considered zero
- Can be used to measure variance in text
- Applications to data compression, encryption

Entropy Example: Coin flip

- Consider simple example: Flipping a Fair Coin
- Variable: X
  - X=H or X=T
- $H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i)$
  - $= -(p(X = H) log_2 p(X = H) + p(X = T) log_2 p(X = T))$
  - $= -(\frac{1}{2} log_2 \frac{1}{2} + \frac{1}{2} log_2 \frac{1}{2})$
  - $= -1 * log_2 \frac{1}{2} = -1 * -1 = 1$
- Therefore a fair coin flip represents one bit of information
- We need one bit of information to represent the result of a coin flip

Entropy and a Letter

- Consider Entropy for a Random Letter
  - If α is randomly-chosen English letter, the probability that α is any particular letter is $\frac{1}{26}$
  - Entropy for a randomly-chosen English letter is:
    - $-\sum_{i=1}^{26} \frac{1}{26} log_2 \frac{1}{26}$
    - $= -26 * \frac{1}{26} log_2 \frac{1}{26}$
    - $= -log_2 \frac{1}{26}$
    - $= log_2 26$
    - $= 4.7$, rounded
- Therefore, a randomly-selected English letter represents 4.7 bits of information
- We need approximately 4.7 bits to represent the value of one random letter
- Therefore, if a password consists of ten randomly-selected letters, it has an entropy of 47 bits

Entropy and Unequal Frequency

- But wait! English doesn't use letters with equal frequency
- What happens when some letters are used more frequently than others?
  - Instead of each letter having a probability of $\frac{1}{26}$, let's suppose that for a randomly occurring letter:
    - ten letters have a probability of $\frac{1}{30}$
    - ten letters have a probability of $\frac{2}{75}$
    - six letters have a probability of $\frac{1}{15}$
- Now the entropy of a random character is
- $H(letter) = -(10 * \frac{1}{30} log_2 \frac{1}{30} + 10 * \frac{2}{75} log_2 \frac{2}{75} + 6 * \frac{1}{15} log_2 \frac{1}{15})$
- $= 10 * \frac{1}{30} log_2 30 + 10 * \frac{2}{75} log_2 \frac{75}{2} + 6 * \frac{1}{15} log_2 15$
- $= \frac{1}{3} * 4.9 + \frac{20}{75} * 5.2 + \frac{6}{15} * 3.9 = 4.58$

Entropy and English

- When we make some letters slightly more likely than others, the entropy of a given letter changes from 4.7 to 4.58
- In general, less variance leads to less entropy
- In fact, Shannon calculated that a given letter in English has an entropy of *1*
- This means that a letter of English text can be represented on average by a single bit

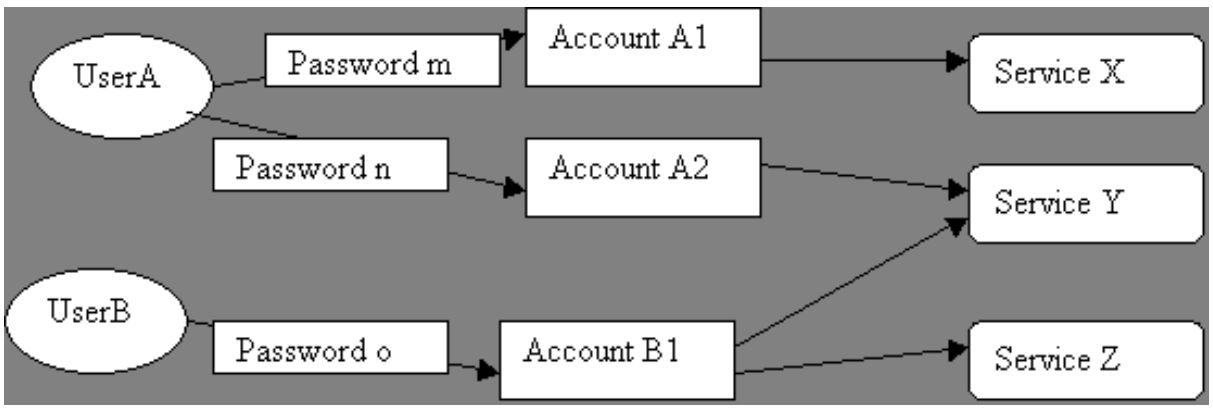
Outline

A Brief Overview of My Prior Work

- *A comprehensive simulation tool for the analysis of password policies*
  - Richard Shay and Elisa Bertino [Shay & Bertino, 2009]
  - International Journal of Information Security
  - Springer, 2009
- Simulating users and their password policies in an organization
- Studies impact of password policy on financial health of organization
- Most citations in this presentation taken from the paper
- Download at `http://richshay.com/files`

The Model Components

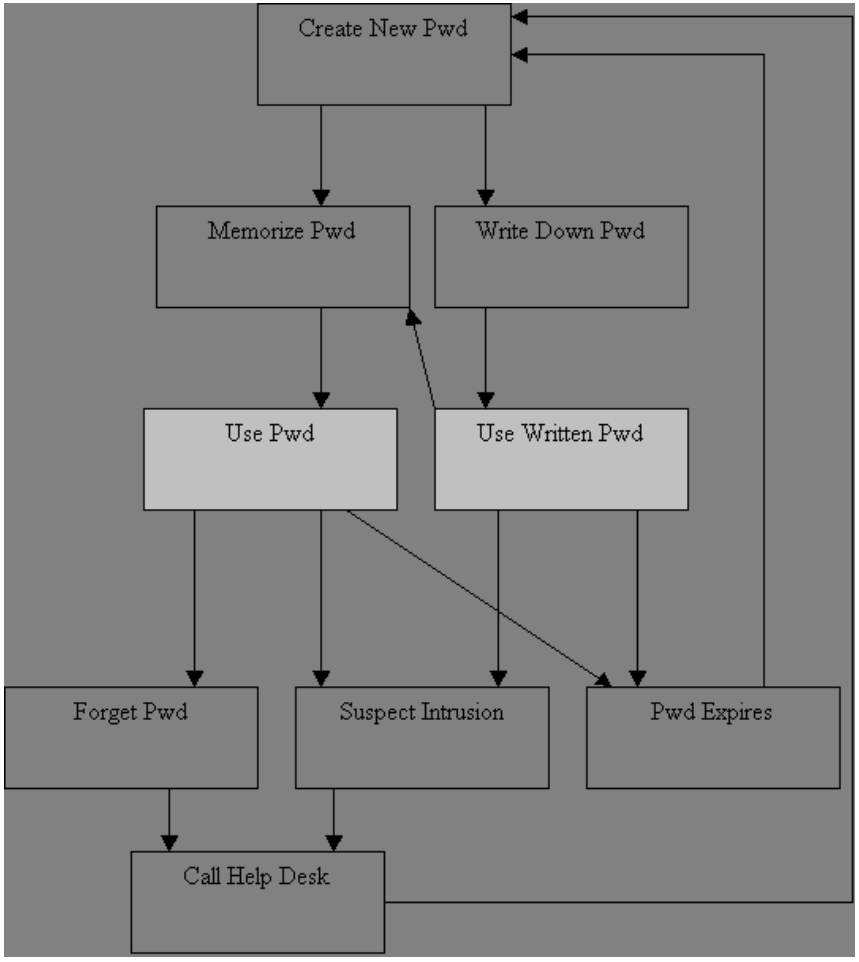
- Parameters can be defined for *users*, *accounts*, *services*



- Users, services have daily fixed cost
- Users generate income by using accounts to access services

Password Lifecycle Model

- Each account has exactly one password
- Policy dictates password length, complexity, change frequency
- Password changed when it expires, users suspects account compromised, user forgets password
- User with password not memorized writes it down



User Memorization

- Users subjected to memory checks with new password
- Checks continue until users memorizes password
- Until user has memorized password, user writes it down
- Probability of success of check depends on:
  - User probability of memorizing seven-digit phone number (entered)
  - Variable indicating how quickly the user learns (entered, 0 to 1)
  - Complexity of password (per-character entropy*length, entered)
  - How long user has been using password
  - How many new passwords the user creates daily, average

Threat Model

- If an account is *compromised*, all services it uses are *compromised*
- Compromised services produce admin-specified fraction of usual income, and may have added daily cost
- Compromised service remains compromised until all accounts tethered to it are no longer compromised
- Compromised account remains compromised until its password changes

Becoming Compromised: Internal

- Each day that a user's password is written down, there is a daily probability that the user becomes compromised because of this
- This probability is specified by administrator
- Represents threat created by user writing down password by computer

Becoming Compromised: Modeling Cracking

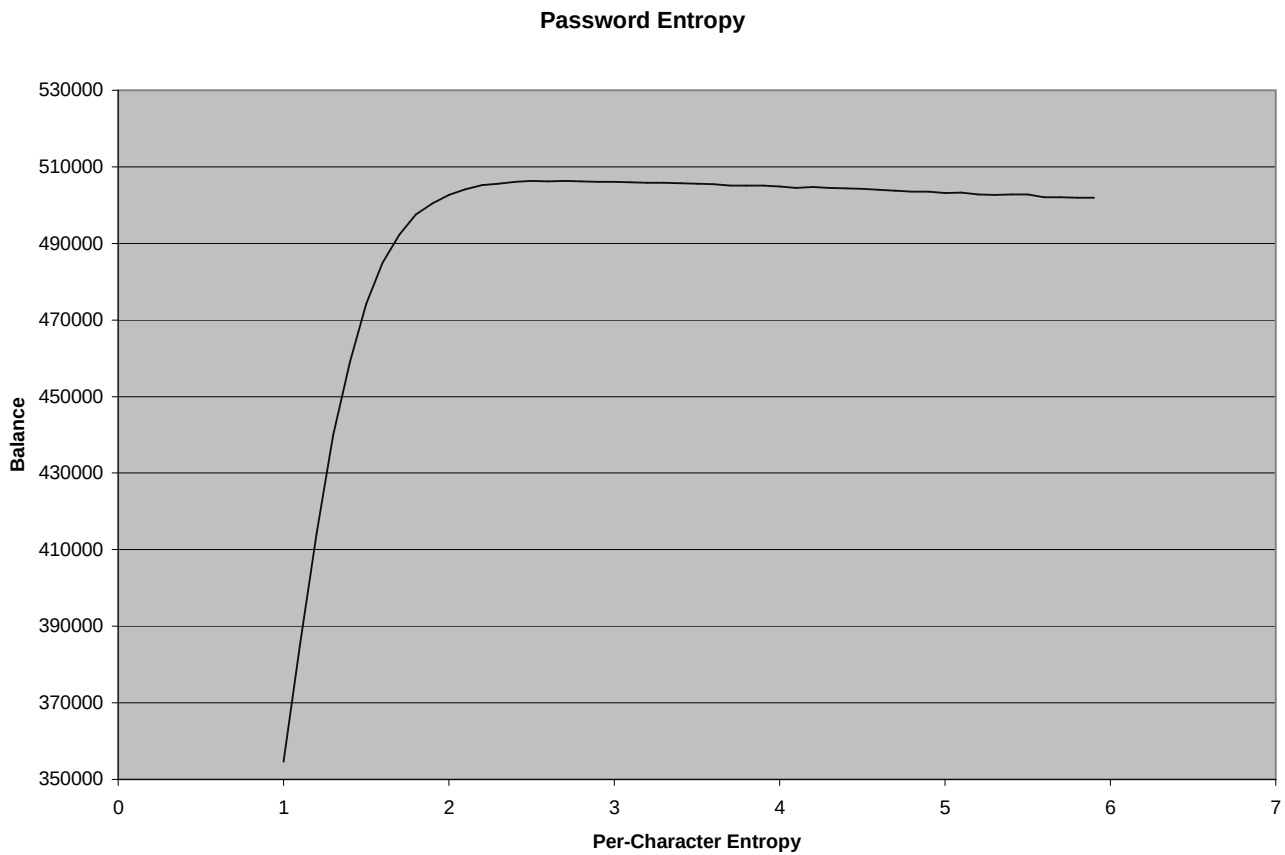
- My favorite part of the model!
- Each user subjected to a specified number of daily brute-force attacks
- Each account has a password with a specified length and per-character entropy
- Therefore, each password has a specified total entropy $\mathcal{H}$
- This means that the password is represented by one configuration of $\mathcal{H}$ bits
- Therefore, a random brute-force attack has a probability of success of
  - $\frac{1}{2^{\mathcal{H}}}$

Result Methodology

- Results examine how changing input parameters changes end financial balance
- *Y-axis* is end financial balance of simulated organization
  - Higher final balance indicates less money lost due to security breaches
  - Security breaches lead to added costs and reduced income
  - Therefore balance is measure of policy success
- *X-axis* shows parameters in question
- Each point shows mean balance of several runs
- Program has many input parameters; the rest held constant
- Experiments created through graphical interface
- Model capable of many other experiments

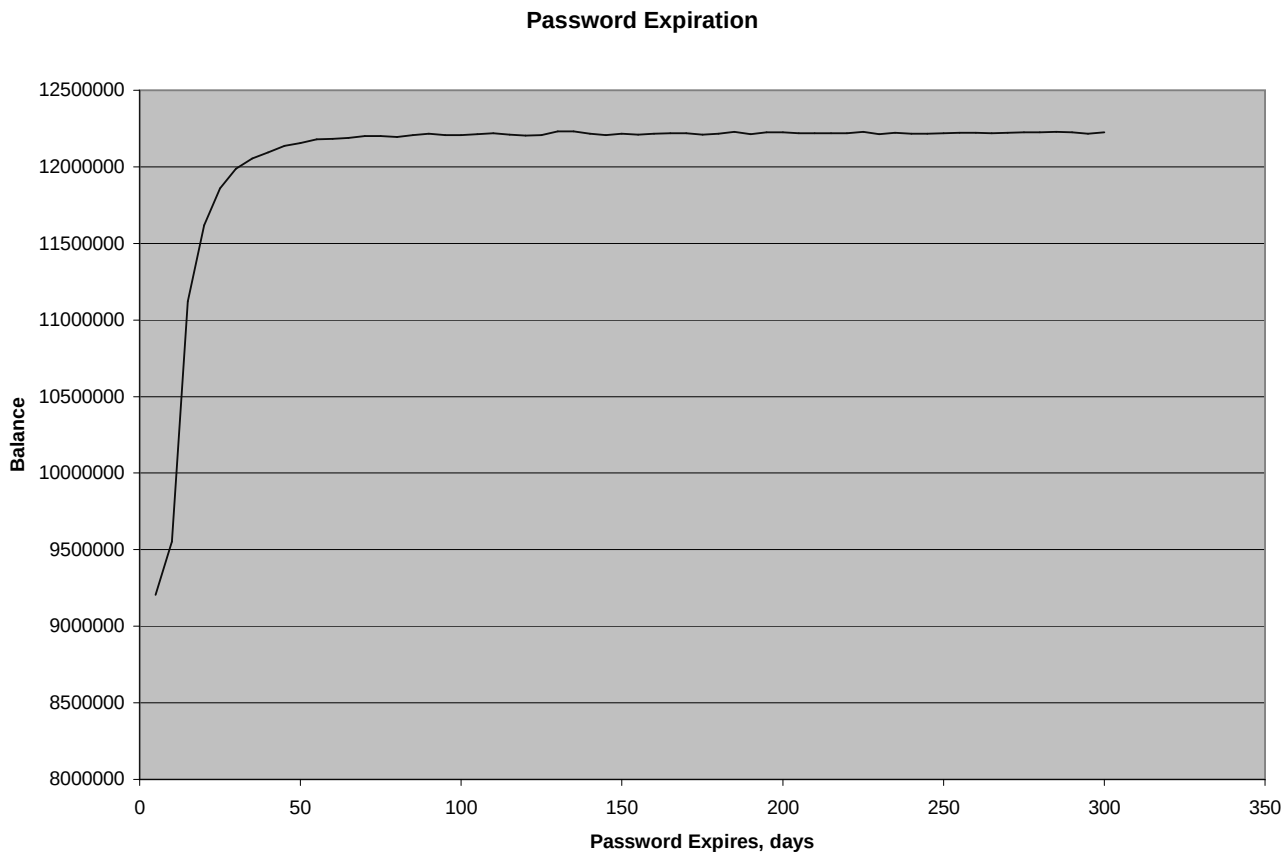
Experiment One

- Result of increasing password complexity



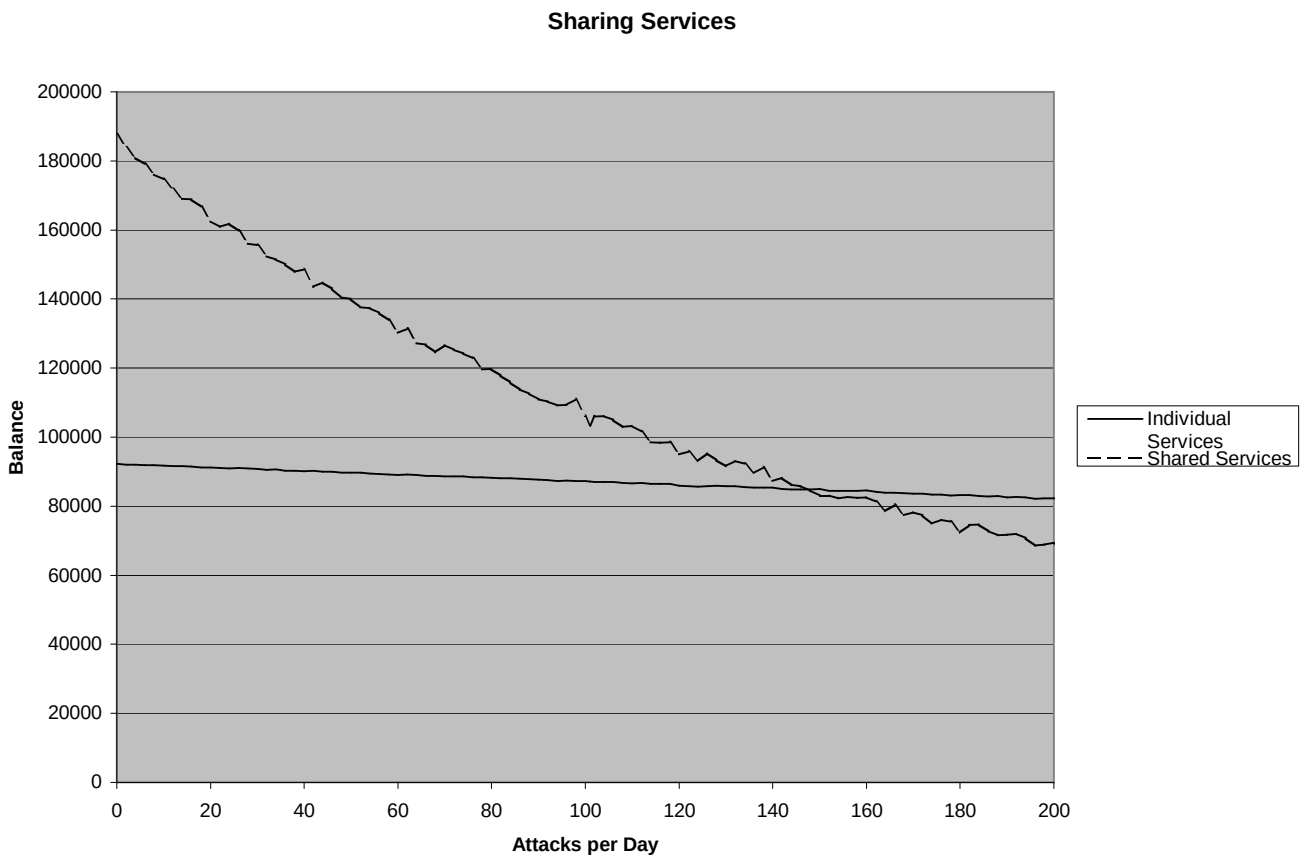
Experiment Two

- Result of increasing duration of password life



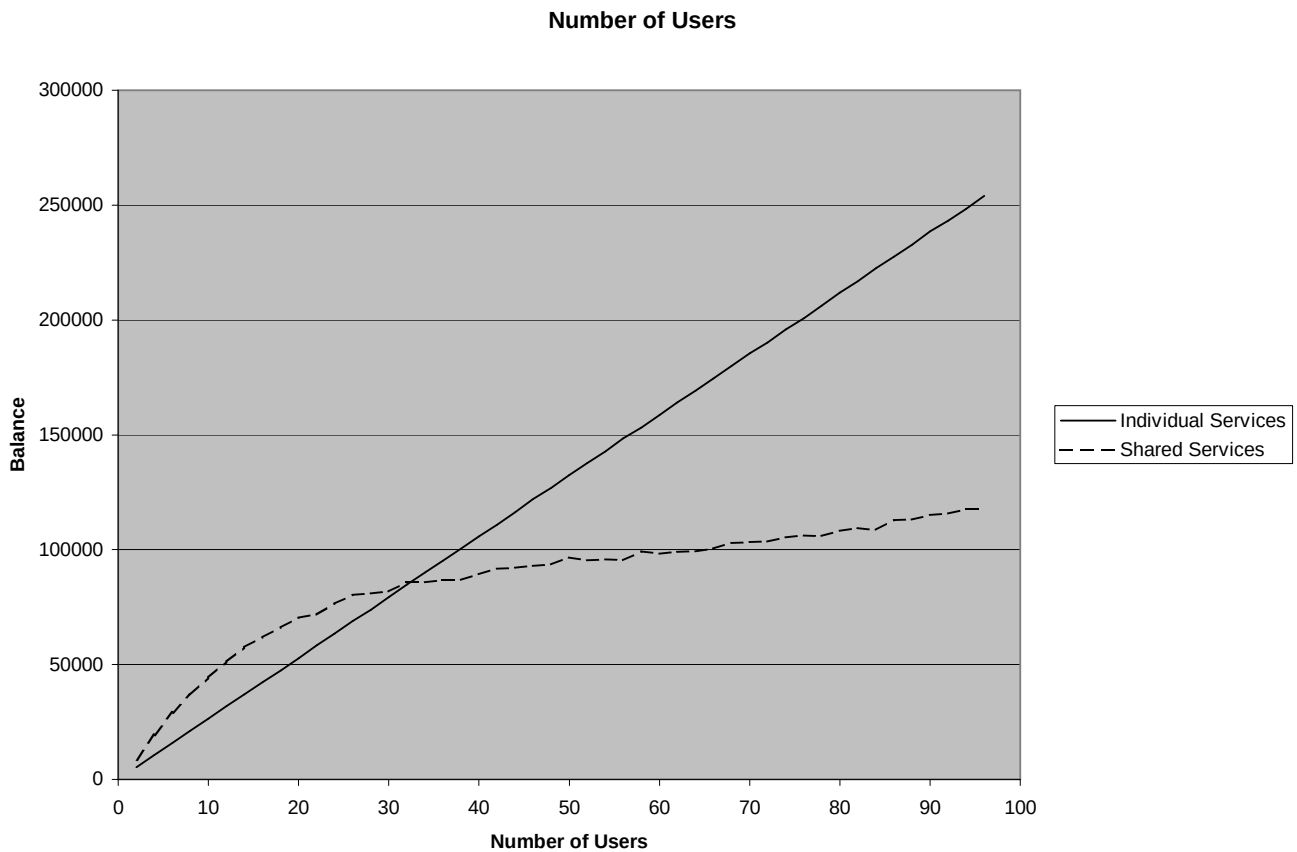
Experiment Three

- Shared vs individual services as external attacks increase



Experiment Four

- Shared vs individual services as organization grows



Further Work?

- I've been considering how best to release this as an open-source project
- Let me know if you have any ideas for researching using this program

 Bishop, M. & Klein, D. V. (1995).
Computers and Security14, **14** (3), 233–249.

 Kuo, C., Romanosky, S., & Cranor, L. F. (2006).
In: *SOUPS '06: Proceedings of the second symposium on Usable privacy and security* pp. 67–78, New York, NY, USA: ACM Press.

 Leyden, J. (2003).
The Register, .

 Proctor, R. W., Lien, M.-C., Vu, K.-P. L., Schultz, E. E., & Salvendy, G. (2002).
Behavior Research Methods, Instruments, and Computers, **34** (2), 163–169.

 Robert M. Polstra, I. (2005).

In: *InfoSecCD '05: Proceedings of the 2nd annual conference on Information security curriculum development* pp. 135–138, New York, NY, USA: ACM Press.


Shay, R. & Bertino, E. (2009).
Int. J. Inf. Sec. **8** (4), 275–289.


Summers, W. C. & Bosworth, E. (2004).
In: *WISICT '04: Proceedings of the winter international synposium on Information and communication technologies* pp. 1–6, Trinity College Dublin.


Wikipedia (a).
http://en.wikipedia.org/wiki/Entropy_(information_theory).