Buying time - What is your data worth?

"The power which money gives is that of brute force; it is the power of the bludgeon and the bayonet."

--- William Cobbett

"When in doubt, use brute force."

--- Ken Thompson
Who am I?

Adam Bregenzer
(adam@bregenzer.net, adan@sli.cc)

Member of Kaos Theory and DC404
Developer for Anonym.OS and SAMAEL
Developer for GroupHug.us
Distributed What?

- Scalable password cracking
- Can use word lists, character brute forcing, rainbow tables, or anything you can think of!
- Can produce – the answer, rainbow tables, or again, anything you can think of!
- Provides a flexible framework for cracking passwords across a network of computers.
So, why do I care?

- Brute forcing is an “assumed risk” and often dismissed.
- The ability to rent computers, or access to grid computing and storage means that processing power is “infinite”.
- CPU = hard dollar cost, therefore password cracking has a hard dollar cost.
- Your passwords have a fixed, decreasing cost that is based on their complexity and application.
How is this useful?

- Password strength is one way of measuring the safety of your data.
- Practical security can be measured as the relationship between the efforts required to break your security and the value of your data.
- Therefore the value of your data is analogous to your password's strength, or price.
This is BAD!

- Distributed attacks against password hashes rapidly reduce this cost.
- As do Rainbow tables
- Additionally, Moore's Law means this cost will halve every two years.
- Botnets today have enough processing power to crack your passwords!
A practical model - libattkthread

- Helper code to facilitate multi-threaded cracking
- Only need to write a simple function to process cracking a single word.
- Asynchronus, interruptable, extendable.
- Resulting library can be extended to take advantage of distributed processing framework
Quick Tech Review

Password Attacks
Different types – brute force, dictionary, rainbow tables
Prevention mechanisms – Salts, limited retries

Distributed Computing
“embarrassingly parallel”
distributed.net
Similar tools
djohn
john's external:parallel
Access Data's DNA
libattackthread Design

- Attack Initialization – attack_st structure
  - Readers and Writers
  - Number of threads
  - Cracking function
  - Callback
- Starting and stopping the attack
- Checking the status of the attack
Implementing libattkthread

- Building an attack function
- Bringing in words
- Writing out hashes
- Initializing and starting an attack
- Building an attack library
Running an Attack

- Processing user values
- Running the attack
- Signal handling
- Building the executable
Implementing the Distributed Attack Framework

- Building the module
- Starting the server
- Linking multiple servers
- Starting and monitoring an attack
Creating an Attack Module

- Using the template
- Defining the input values
- Creating the init and start methods
- Building the module
Installing and Running the Server

- Configuring it
- Starting it
- Linking it to another server
Advanced Techniques

- Logging in and starting an attack
- Watching and controlling the attack
Modifying libattkthread

- Making new readers and writers
- Other uses of the framework
Modifying the Distributed Framework

- Linked servers
- Secure communication
What next?

- P2P Distributed processing
- Advanced clustering
Demo

DEMO!
Thanks To

- Cowpatty
- Joshua Wright
- Renderman
- Byron for testing!
- kaos.theory
- DEFCON
Questions?

???