KoreLogic
SECURITY

Cracking Corporate Users' Passwords Made Easy

September 2011
ISSA, Richmond Chapter

Hank Leininger – KoreLogic

https://www.korelogic.com/

Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- Corporate password strength policies
- Anatomy of Password Cracking Tools
- Reality of Weak Corporate User Passwords
- DEFCON Crack Me If You Can contest, 2010 and 2011
- The Year of Pastebin

Background on me

Hank Leininger <hlein@korelogic.com>
BE5D FCCA 673B D18B 98A9 3175 896E 3D4A 1B4D C5AC

Played defense as a sysadmin / security admin since the mid 90's.

Had to learn how to think like the bad guys.

Decided that attackers had more fun.

Have been doing security consulting for a little over a decade; co-founded KoreLogic in 2004.

Also run the MARC mailing list archive site: http://marc.info/

Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- Corporate password strength policies
- Anatomy of Password Cracking Tools
- Reality of Weak Corporate User Passwords
- DEFCON Crack Me If You Can contest, 2010 and 2011
- The Year of Pastebin

Public Password Disclosures

There have been some really big, widely publicized password disclosures in recent years:

- RockYou, 2009: 32 million users' details including plaintext passwords
- Monster, 2009: 4.5 million users' encrypted passwords
- Nate / Cyworld, 2011: 35 million users' details
- Sega, 2011: 1.3 million users' encrypted passwords
- Gawker, 2010: 1.5 million users' encrypted passwords
- Sony
- Sony
- Also, Sony
- And an endless stream of small web forums/bulletin boards, such as roughly half of all phpBB websites ever made

An excellent collection of info about breaches of passwords, credit cards, etc is maintained at: http://datalossdb.org/

Public Password Disclosures – What do they tell us?

There are two big lessons here (although neither is really new):

- As an IT professional: users, when given no password strength requirements, will choose really, really awful passwords.
    - 12345
    - 123456
    - password
    - qwerty
    - firstnamelastname
    - companyname

- As a human: expect to be betrayed by every company you interact with online. That way you'll be pleasantly surprised 50% of the time...
  (But that isn't really our topic today.)

Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- **Corporate password strength policies**
- Anatomy of Password Cracking Tools
- Reality of Weak Corporate User Passwords
- DEFCON Crack Me If You Can contest, 2010 and 2011
- The Year of Pastebin

Corporate Password Strength Policies

That's why corporate environments always have password strength policies, right?

Minimum length, complexity rules (upper, lower, number, punctuation), and histories... those will keep us safe, right?

Corporate Password Strength Policies

That's why corporate environments always have password strength policies, right?

Minimum length, complexity rules (upper, lower, number, punctuation), and histories... those will keep us safe, right?

Wrong...

Corporate Password Strength Policies – Easy for Users to Evade

A nice, strong-looking policy such as:

- Minimum 9 characters
- Upper & lowercase, numbers, punctuation

Corporate Password Strength Policies – Easy for Users to Evade

A nice, strong-looking policy such as:

- Minimum 9 characters
- Upper & lowercase, numbers, punctuation

...Is perfectly happy to accept “P4ssword!”, J0hn.Sm1th”, etc.

Which—as we'll discuss later—are trivial to crack.

Corporate Password Strength Policies – Easy for Users to Evade

Password expiration and password histories can also work against you.

How about we force users to change their passwords every 30 days, and they can't reuse for 18 months? Sound good?

Corporate Password Strength Policies – Easy for Users to Evade

Password expiration and password histories can also work against you.

How about we force users to change their passwords every 30 days, and they can't reuse for 18 months? Sound good?

10% of them will currently have a password of “Sept2011!”, “Sept%2011”, or “Aug.2011!”

Even worse, once I know that, I can come back 18 months from now and guess it on the first try.

Corporate Password Strength Policies – Driving Users Crazy

What's an IT person to do? Right about now, cry into your pasta.

Meanwhile, users are ready to mutiny because of what they perceive as draconian, over-the-top security measures.

We'll come back to this...

Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- Corporate password strength policies
- Anatomy of Password Cracking Tools
- Reality of Weak Corporate User Passwords
- DEFCON Crack Me If You Can contest, 2010 and 2011
- The Year of Pastebin

Anatomy of Password Cracking

What is password cracking anyway?

If you know this one, now's a good time to check your email, but I'll review just in case.

Anatomy of Password Cracking – Online Attack vs Offline Attack

“Online” password-cracking typically means connecting to a target server over and over, and attempting to log in with guessed username & password pairs. This is noisy (generates logs), slow (relatively speaking), and likely to lock users out.

Usually, people just refer to this as login brute-forcing/guessing. “Password cracking” usually means “Offline” cracking, in which the attacker isn't connecting to the server over and over again.

Anatomy of Password Cracking – Offline Password Cracking

For a given system that encrypts users' passwords, the server converts a user's plaintext into a ciphertext using some specific encryption algorithm, and stores the ciphertext (/etc/shadow, Windows SAM, etc).

In an offline password-cracking attempt, an attacker has somehow gotten ahold of the encrypted hashes (the ciphertexts), and wants to determine the plaintexts.

Unless the cryptography is utterly, horribly broken (read: do-it-yourself), it's impossible with today's math to work backwards from a ciphertext to figure out what the original plaintext was.

- It's akin to being given the number 36, and not knowing if it was originally composed of 2x18, 3x12, 4x9, or 6x6.

Anatomy of Password Cracking – What is it?

So, the attacker has to guess (or bruteforce) plaintexts; and keep trying the same encryption function the server uses:

```
for each candidate_plaintext
    result = encryption_function(candidate_plaintext)
    done if (result matches the ciphertext)
```

Exactly how the encryption function works typically depends on the underlying crypto algorithm (DES, MD5, SHA1) and how it is applied:

- MD5(plaintext)
- SHA1(salt . plaintext)
- MD5(MD5(MD5(MD5(plaintext))))

Anatomy of Password Cracking – Cracking tools

So at a high level, conventional password cracking tools need:

- To support the hashing algorithm of your target ciphertext, and hopefully be fast at it.
- To be able to generate candidate plaintexts to try.

Some tools are good at both of these; some really only one or the other.

Anatomy of Password Cracking – Cracking tools

So at a high level, conventional password cracking tools need:

- To support the hashing algorithm of your target ciphertext, and hopefully be fast at it.
- To be able to generate candidate plaintexts to try.

Some tools are good at both of these; some really only one or the other.

(Some are only really good at parting you from your money.)

Anatomy of Password Cracking – Cracking tools

Password cracking tools were traditionally CPU-based (because everything was):

- John the Ripper

  Support for lots of different hash types
  Runs on lots of different operating systems
  Fast by CPU-cracker standards
  Excellent at candidate-generation
  Free and open-source
- Hashcat

  Support for a good variety of hash types
  Runs on Windows, Linux, some other UNIXes
  Free, but closed-source
- Cain & Able

  Support for even more hash types
  Windows-only
  Free but closed-source
- L0phtCrack

  Pretty

Anatomy of Password Cracking – Cracking tools

In recent years GPU-based password crackers have been developed. Historically, GPU-crackers were fast, but dumb: they could only handle a few hash types (raw MD5, SHA1) and could only bruteforce (aaaaaaa, aaaaaab ...). But that has been changing:

- oclHashcat suite
  - Limited algorithm support initially; more every month
  - Unbelievably fast: 10x – 1000x as fast as a CPU, depending on the algorithm.
  - Very smart at candidate generation
  - Runs on Windows and Linux
  - Free, closed-source
- Cryptohaze MultiForcer
  - Limited algorithm support
  - Very fast
  - “Smart” brute forcer

Anatomy of Password Cracking – Cracking tools

Rainbow-table cracking tools work differently.

Rainbow tables are pre-generated data files (often big – terabytes and terabytes) built using a whole lot (months, years, or decades) of CPU time.

Those tables are then used as a partial lookup table when cracking a given ciphertext, resulting in a *huge* speedup (seconds or minutes instead of hours, days, or months).

Rainbow tables can be *really* effective in some circumstances. But for some they are useless; for others they work but are not the best tool for the job.

Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- Corporate password strength policies
- Anatomy of Password Cracking Tools
- **Reality of Weak Corporate User Passwords**
- DEFCON Crack Me If You Can contest, 2010 and 2011
- The year of Pastebin

Reality of Weak Corporate User Passwords

Users tend to behave predictably when creating plaintexts that pass strength rules, such as:

- **``[A-Z][a-z]+[0-9]+[!-/:-@^-`]``**
  (Word with first letter capitalized, 2-4 numbers, punctuation)
- **``[A-Z][a-z]+[!-/:-@^-`][0-9]+``**
- **``[0-9]+[A-Z][a-z]+[!-/:-@^-`]``**
- **``[0-9]+[!-/:-@^-`][A-Z][a-z]+``**
- Including the company name, city, local sports teams
- Finger patterns, like:

```
!@#$     edx
qwer     rfc
Asdf     fjdksl
zxcv     1!2@3#4$
```

These permutations on dictionaries can be tried in seconds, and they usually work on the majority of users' passwords.

Reality of Weak Corporate User Passwords

We found that in companies that do not actively police for things like that, users passwords were *very* predictable.

So we'd recommend, and sometimes help them set up password cracking programs internally, etc...

Reality of Weak Corporate User Passwords – What can you do?

There's no easy answer, but these help:

- Pro-actively crack your organization's passwords; force changes for the weakest.
- Include the “why” in password-strength user education
- Make secure password-storage a purchasing requirement—use salts, difficult algorithms, etc. (Unfortunately this would rule out MS Windows...)
- If users need to maintain lots of different sets of credentials, look for tools to facilitate that—GPG, keepass, various commercial options.
- Push for stronger authentication methods
  - Public key (SSH authorized_keys, certificates – dare I say PKI?)
  - 2-factor using hardware tokens, SMS to mobile devices, etc.

Reality of Weak Corporate User Passwords

But we saw few people talking about this publicly, and new clients were always surprised when we pointed it out.

We wanted to change that.

Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- Corporate password strength policies
- Anatomy of Password Cracking Tools
- Reality of Weak Corporate User Passwords
- **DEFCON Crack Me If You Can contest, 2010 and 2011**
- The Year of Pastebin

DEFCON Crack Me If You Can 2010

KoreLogic sponsored a password cracking contest at DEFCON 18 in 2010, designed to highlight and get people talking openly about what we'd seen happening within corporations.

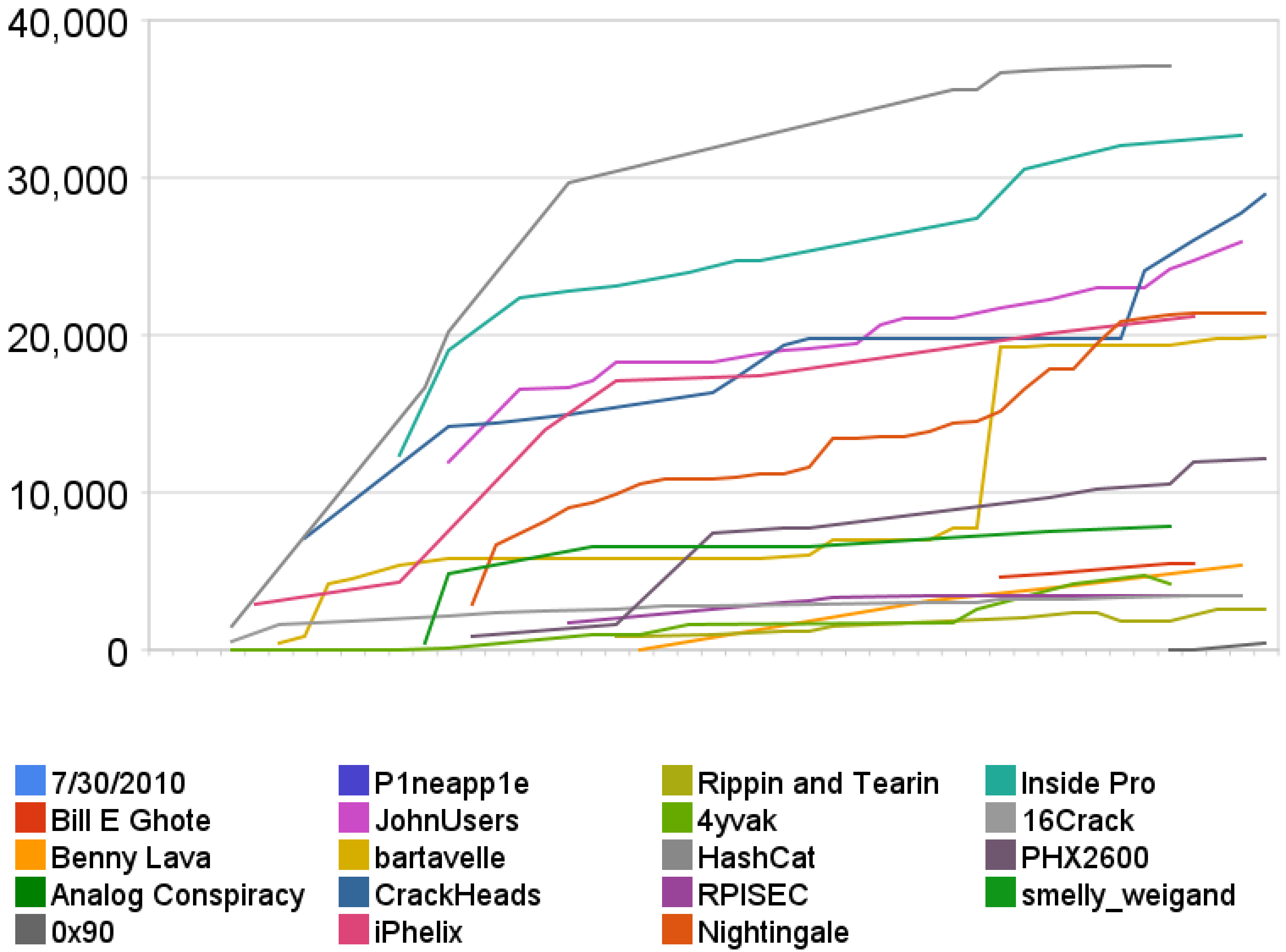
- We created ~54,000 plaintext passwords, based on every behavior we'd seen in common use within companies where we'd done pentests, ranged from very easy to very hard.
- We hashed them using the kinds of encryption types we actually see in corporations—Windows LanMan & NTMD4, UNIX DES, FreeBSD MD5, SHA and Salted SHA used by LDAP servers, etc.
- A percentage of these could be cracked using rainbow tables, “dumb” brute force, etc.
- But most couldn't. The only way to win was to be smart.

DEFCON Crack Me If You Can 2010

The goal of the contest was simple: Crack the most passwords in a 48-hour period, using any resources the teams had legitimate access to, including remote computers and members.

The catch was, in order to win they had to be willing to “show their work”: write up their methodology, and publish it to the world.

DEFCON Crack Me If You Can 2010



DEFCON Crack Me If You Can 2010

The bottom line: it worked!

- Teams did creative problem-solving, figured out the behavior patterns we had embedded in the data.
- Some new software was developed by teams on-the-fly, which was later released.

Also, afterwards we released a whole bunch of specialized dictionaries and password cracking rulesets (for John the Ripper and for the Hashcat suite) that had gone into creating the plaintexts.

You can read more about the 2010 contest, including our rulesets, and writeups from the teams at:

http://contest-2010.korelogic.com/

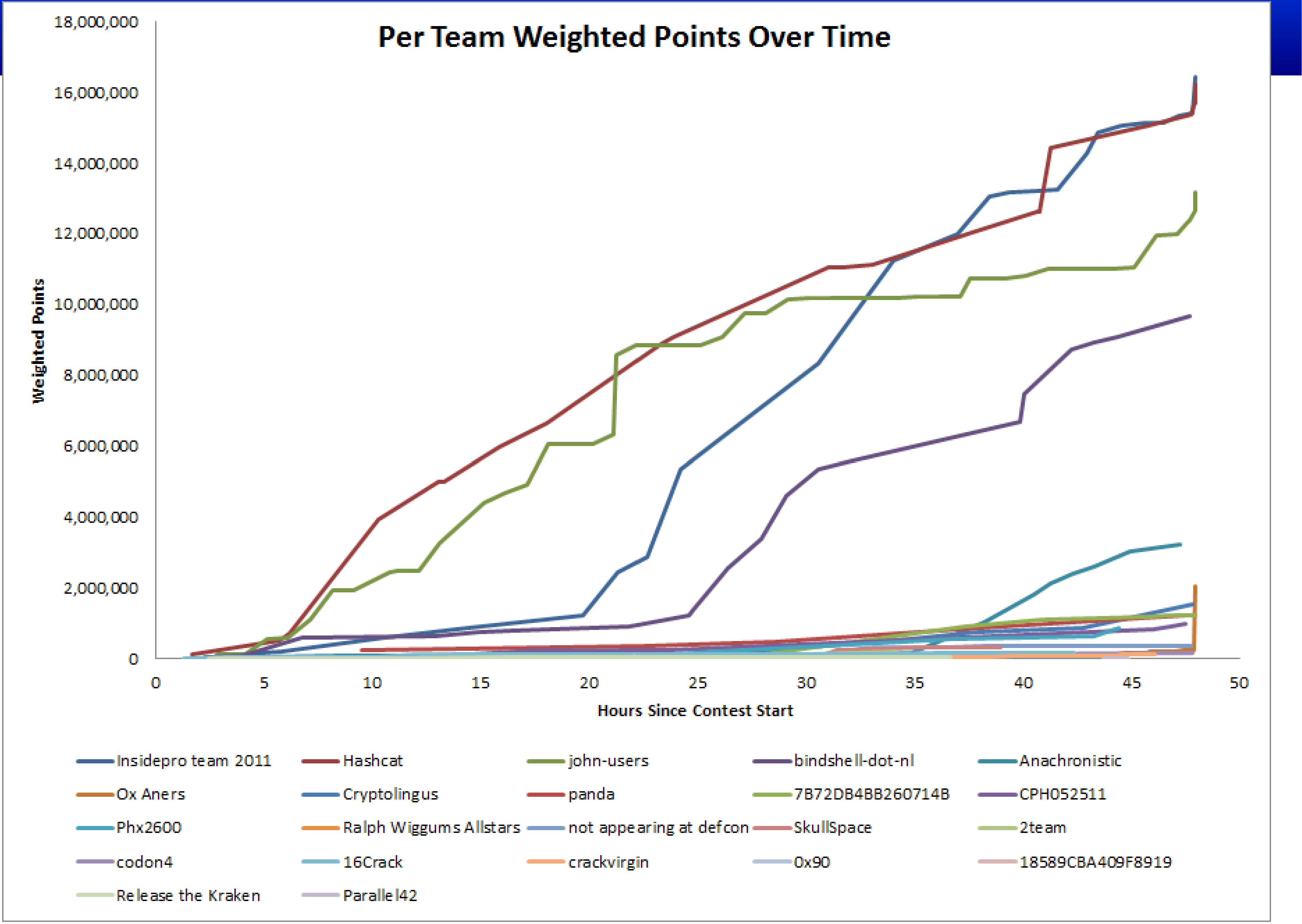
DEFCON Crack Me If You Can 2011

Last month we ran the contest again at DEFCON 19.

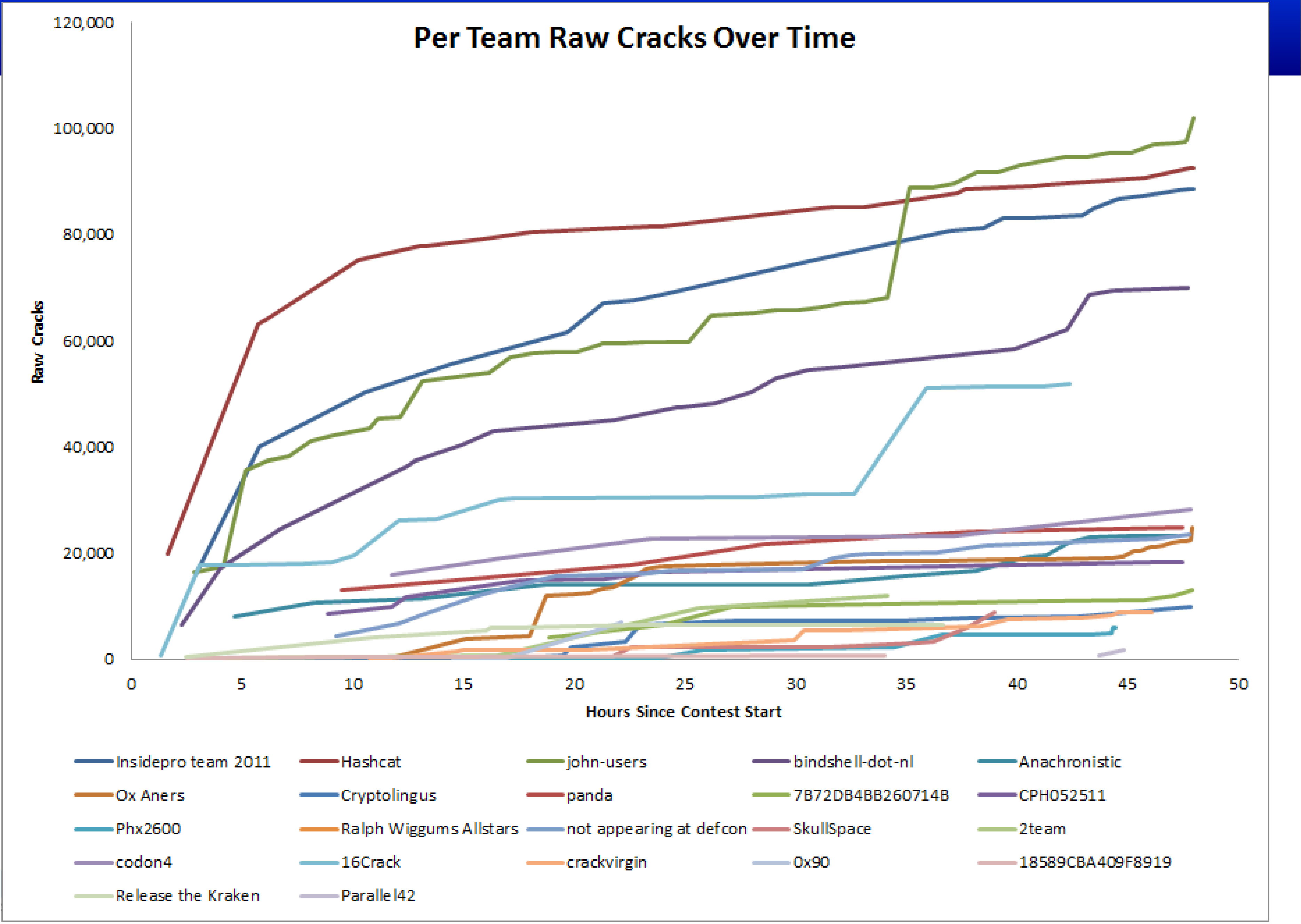
This time we emphasized other things we wanted to see get more attention, such as:

- Passphrases: combining multiple dictionary words (or words from multiple dictionaries)
- More hash types, with more points awarded for the harder ones
- Encrypted container files: we included some encrypted .zip's, .rar's, .dmg (OSX disk image file)

DEFCON Crack Me If You Can 2011



DEFCON Crack Me If You Can 2011



Agenda

- Background on me (mercifully short)
- Public password disclosures (rockyou, etc)
- Corporate password strength policies
- Anatomy of Password Cracking Tools
- Reality of Weak Corporate User Passwords
- DEFCON Crack Me If You Can contest, 2010 and 2011
- **The Year of Pastebin**

The Year of Pastebin

The site you don't want to appear on used to be Attrition, because you'd be embarrassed.

Now it's pastebin, because it'll contain your CEO's password.

Attackers are sharing their hash dumps on sites like pastebin, paste2, etc. They are posting publicly, if not necessarily advertising.

There are also private file-sharing networks dedicated to lists of password hashes.

I analyzed all the public posts to pastebin.com in a 15-hour period earlier this week, and found about 50,000 plaintext and encrypted passwords in about 20 files. Not to mention some router configs (with passwords) and logs of compromising several sites, including some .gov sites.

The Year of Pastebin

Added to pastebin.com in a 15-hour period earlier this week:

UhxsD___	700+ email addresses and plaintext passwords
UPdW0___	**10,000+** MD5 hashes and corresponding plaintexts
PKUuy___	**3500+** SHA1 hashes
TA5Qp___	350+ SHA1 hashes and corresponding plaintexts
DxhkM___	160+ MD5 hashes and corresponding plaintexts
XBDsN___	350+ MD5 hashes and corresponding plaintexts
Bw5Lv___	**17,000+** MD5 hashes and corresponding plaintexts
QaqtF___	800+ Windows LanMan hashes and corresponding plaintexts
Vt98c___	590+ SHA1 hashes and corresponding plaintexts
QBUhB___	50+ SHA1 hashes and corresponding plaintexts
D4EiN___	plaintext usernames and passwords for several websites
Apg1f___	**2100+** MD5 hashes and corresponding plaintexts
K7fF7___	150+ MD5 hashes and corresponding plaintexts
LFHCX___	**5800+** SHA1 hashes and corresponding plaintexts
5Z5yW___	250+ MD5 hashes and corresponding plaintexts
MhcSx___	80+ SHA1 hashes and corresponding plaintexts
ADp4H___	**6500+** MD5 hashes and corresponding plaintexts

Other Reading

- My coworker Rick Redman gave a talk about Advanced Password Cracking techniques at the ISSA Summit in Baltimore in 2010; the slides are linked from:
  http://infosec-summit.issa-balt.org/html/2010_agenda.html

  Rick goes into much more detail about advanced John the Ripper techniques, the various rules we've written for it, and how to write your own.
- For a nice quantitative analysis of the strength (or lack thereof) of common websites' password authentication, see:
  "The password thicket: technical and market failures in human authentication on the web”:
  http://preibusch.de/publications/password_market/
- All the team writeups under http://contest-2010.korelogic.com/ and http://contest.korelogic.com/; we've also published a lot of our rules and dictionaries on the contest-2010 site.
- A good free PAM module to do password/passphrase strength checking and enforcement tool, written by the author of John the Ripper:
  http://www.openwall.com/passwdqc/