Slide 1

Not Your Grandpa's Password Policy

KEVIN NEELY, MANAGER – INFORMATION SECURITY, PURE STORAGE

PURESTORAGE®

Who likes to change their password on an arbitrarily regular basis?

About me

git diff ~/.plan

```
--- pre-2015
+++ current

@@ Overview @@

 - SysAdmin: Active Directory, Exchange, Citrix, Bind, etc.

 - Attorney

 - Incident response lead at ($COMPANY - 2)

 - Security Architect at ($COMPANY - 1)

 + Security lead at Pure Storage

 + BSidesSF volunteer

 + @ktneely on Twitter
```

Talk Overview and Agenda

I want to security like it's 1999

- Demonstrate the ineffectiveness of common password policies
- Describe the ease with which passwords can be discovered
- Show the current state of password use
- Review attack modes
- Review Pure's approach to passwords
- Demonstrate effectiveness of *not* forcing password changes

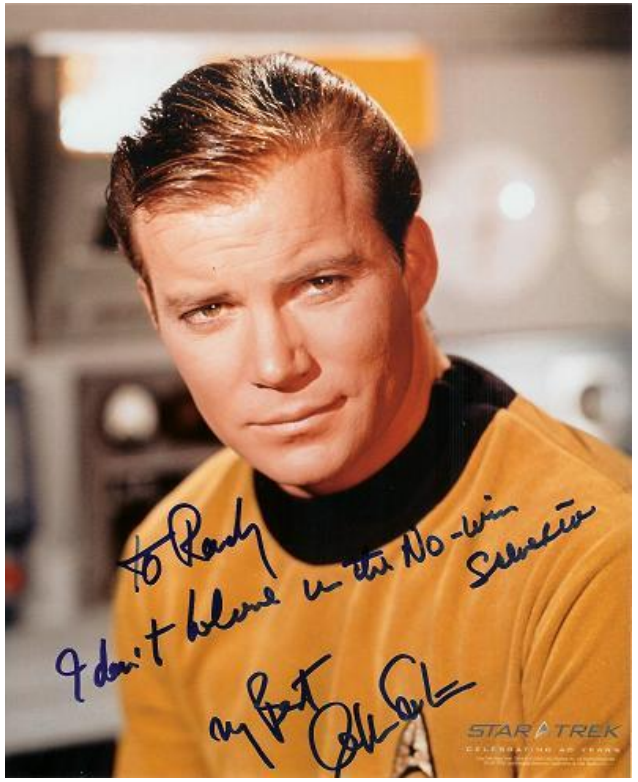
Slides, Notes, and Walkthrough:
https://github.com/ktneely/password-audit

I’m not going to bore you with deep, technical specifics, instead, I will cover password cracking concepts and how we used them to implement this approach,
Provide an overview & framework to get started, how we implemented, and the results we saw.
Specifics to get past the speedbumps
Slides are available… with notes for the talk
Includes detailed walkthrough and scripts to get you up and running quickly

Why Do This?

It's really a win-win

- Implements the new NIST guidance
- An easy-to-implement cutting-edge approach
- Be a hero: Eliminate password changes
- Actually *measure* security awareness
- And most importantly: It's fun!
  - Provides blue team members with a challenging attack activity
  - Build a password-cracking rig!
  - *Really* get to know your co-workers!



PURESTORAGE®

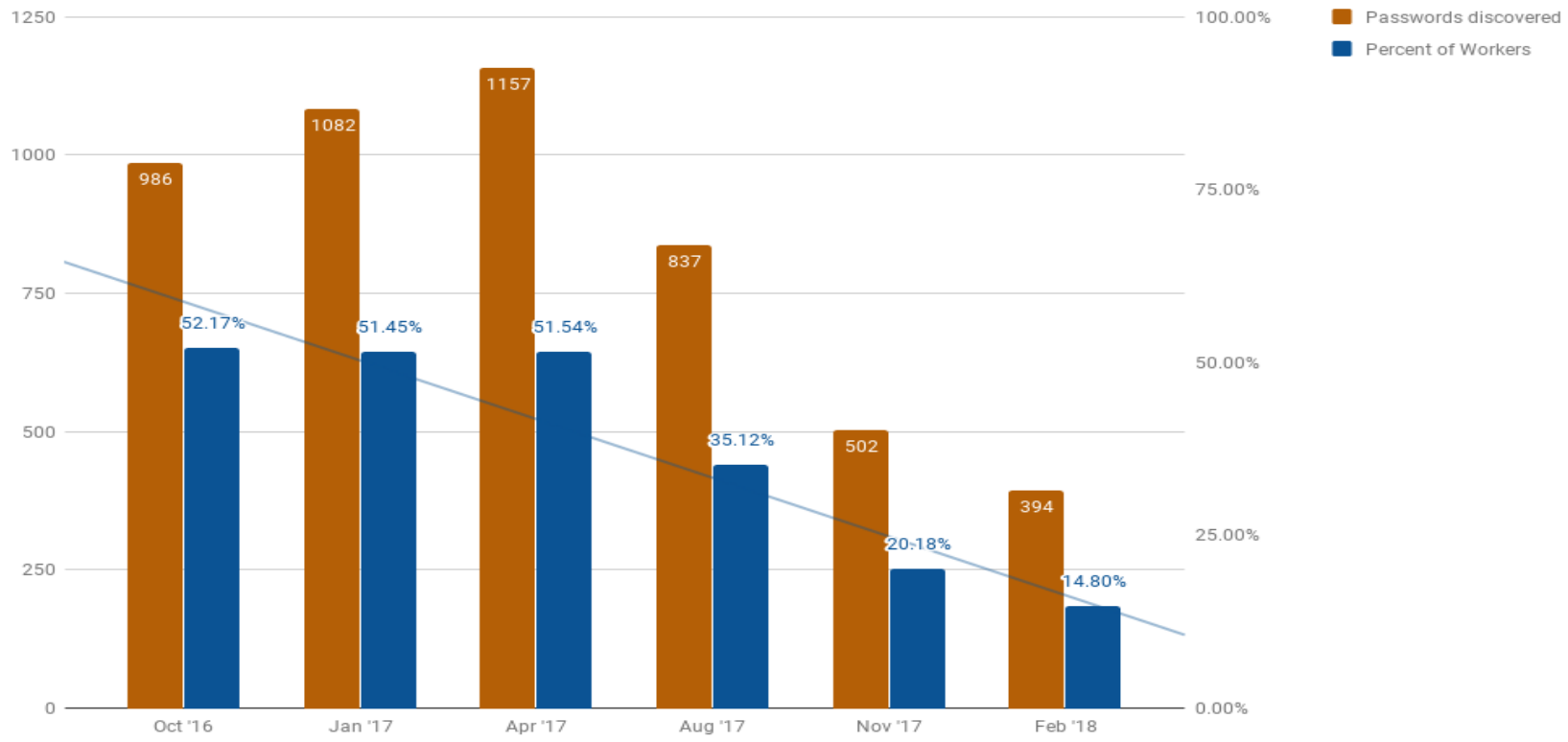
Even NIST no longer recommends regular changing of passwords: https://www.grahamcluley.com/new-nist-guidelines-do-away-with-periodic-password-changes/

- Get a one-up on the pen-testers
-

Slide 5

Program Effectiveness

When they see the benefit, users will choose more complex passwords



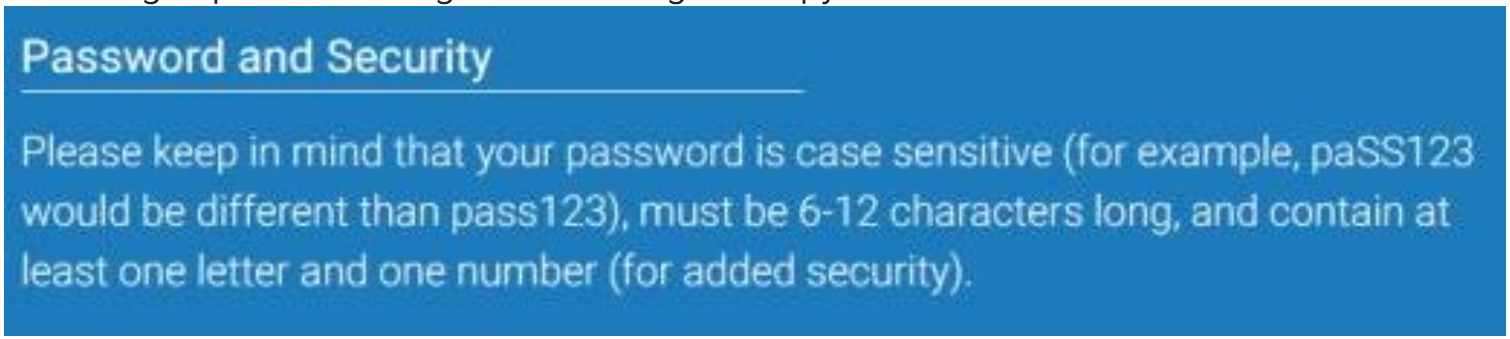
As people learn that they benefit from creating a strong password, they start to actually create them.

Slide 6

MOU3

Common Password Policies - edit

Do nothing to promote strength or encourage entropy



Not OK: Password

OK: P@55w0rd

Policy has no effect, time to crack either: < 1 sec

PURESTORAGE®

Who thinks 12 character strings with three character types makes a strong password?

This is the Password policy from turnitin.com

People tend to choose poor passwords
simple substitutions such as a ‘0’ for an ‘o’.
We take a list of common words, and apply simple transformations

We require 3 of: cap, lower, digit, special char, and at least 9 characters

With average hardware:
8 char: less than a day
10 char: 159 days through entire keyspace
11 characters: 10 years

Slide 7

#InfoSec Twitter Agrees!

You may not know them, but there are people that share your vision

We're not forcing people to change their password just because 90 days have passed.

PASSWORD CRACKING APPROACH

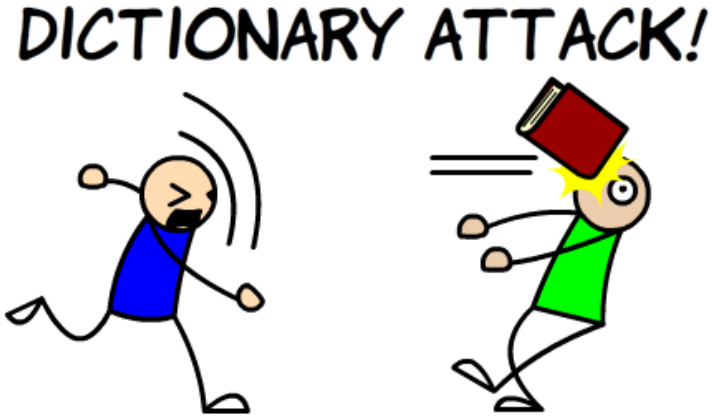
PURESTORAGE®

Has anyone cracked passwords before?

MQU4
KN1

Prerequisites and Equipment

A pinch of technology, a dash of psychology, and a crinkle of intuition



PURESTORAGE®

Things we need:

- Wordlists / dictionaries
- Password cracking tool
- GPU (we use Azure / AWS)
- Understanding your user base (geo-location, language, culture)

Password Cracking Tools

More than one way to skin this cat

- Linux & Windows
- Free
- Forums
- Rules and masks provided

- Linux & Mac OS X
- Free & Paid versions
- Forums
- Wordlists

- Windows
- Network-aware
- No active development

Overview of tools to crack passwords

- Hashcat https://hashcat.net
- John http://www.openwall.com/john/
- Cain http://www.oxid.it/cain.html

Why Hashcat?

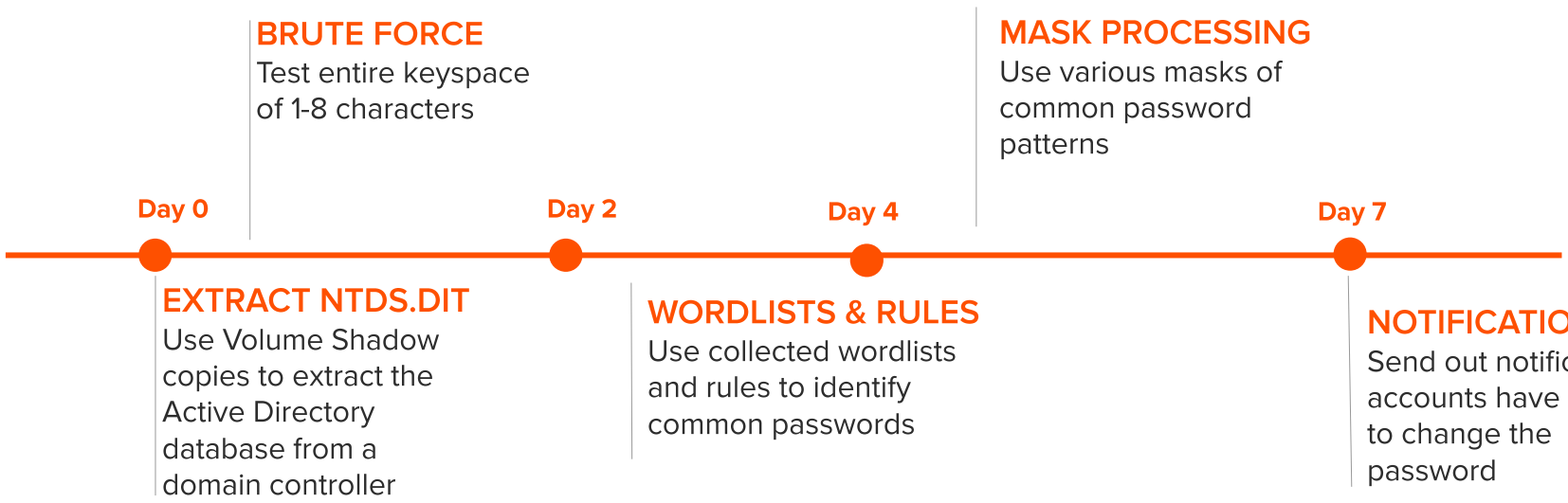
- Hybrid mask attack
- Active community and forums

Slide 11

MOU5

Analysis Timeline

This is how we do it..



PURESTORAGE®

Typical 7-day progression of the password process. We're going to dive into each of these areas over the next few slides.

Totally flexible, but we use the first two days gathering new tools or techniques while the brute force test chugs in the background.

Link to extract the NTDS.dit from Active Directory: https://github.com/ktneely/password-audit/blob/master/doc/extract-hashes.md

Ask a question: how many people are familiar with this process of extracting the hashes.

Brute Force pass

Hitting the problem head-on

PURESTORAGE®

The first step is the easiest, but potentially the most time-consuming. Do what makes sense for your environment, budget, and computer power.

How many people have >= 10 char password?

Even if your settings do not allow for short passwords, there may be legacy accounts or administrator overrides of which you're not aware.

Testing all possibilities from 1-8 characters takes less than two days, so we kick this off right away.
Alternatively, you can run this in parallel, if you have the $$ for two cracking instances.

In 7 runs (almost 2 years), we discovered:

- 37 7-character passwords
- 785 8-character passwords
- That’s over 15% of all discovered passwords.

Analysis technique - Wordlists

We create and download various wordlists

- Released passwords from breaches (Thank you, Troy!)
- Previously discovered passwords
- Dictionaries
- Lists of things (movies, places, sports teams, etc.)
- Tailored lists with Wordsmith (from https://popped.io)

PURESTORAGE®

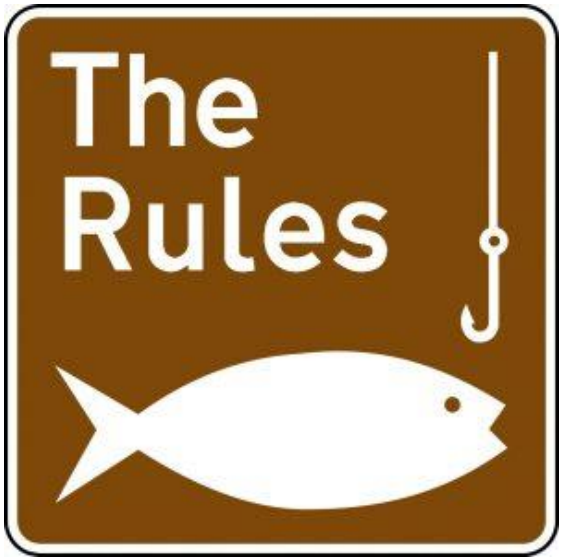
Wordlists form the basis of analysis. In the simplest form, they can be used “straight”, looking for just a match. While brute force is running is a great time to go find some long lists of words.

Downloadable English dictionary:
Password lists from Skull Security:
https://wiki.skullsecurity.org/Passwords
Troy’s released passwords:
https://haveibeenpwned.com/Passwords
Probable passwords: https://github.com/berzerk0/Probable-Wordlists
Wordsmith: https://github.com/skahwah/wordsmith

Analysis technique - Rules

Rules modify the wordlist fed into hashcat

- **Toggles**: a becomes ^
- **Case-swapping**: a becomes A
- **Truncation**: drop the beginning or end of a word
- **Append**: similar to wordlist + mask, but more granular
- **Duplicate word**: word becomes wordword or worddrow
- **Duplicate char**: Alaska becomes Alaskaa



Hashcat can use a rules file, which specifies a number of rules to apply to the wordlist. Here are some examples.

This could be axed

Analysis technique - Masks

I. In Hashcat:

- ?a means any character
- ?l and ?u mean lower and upper case, respectively.
- ?d means digit
- ?u?l?l?l?l?l?l?l?d?d would match/discover Password17 & PStorage99

II. Can be combined with wordlists, such as: Dictionary + ?a?a?a?a

- storage!123
- UnicornP00p
- Drive12!@

III. Masks can also be used to create limited brute force:

?u?l?l?l?l?l?l?l2018– amazing how many 12-character passwords you can get with this!

PURESTORAGE®

The first part describes how to use hashcat's character variables.

Second part

- Take a dictionary with straight English words
- Test for combinations of any four characters afterward
- Would catch year, compound words, and strange strings people thing are secure

Third part:

- People construct passwords in predictable patterns.
- Using masks enables the analyst to search a significant portion of likely passwords in a given keyspace.

- First character upper, followed by all lower, followed by this year can capture a significant number of passwords.

Everyone loves examples

CLI FTW

Analysis step	Command Line
Brute force entire 1-8 character keyspace	`./hashcat64.bin -a 3 -i -m 1000 /ntpass.out --username ?a?a?a?a?a?a?a`
Test passwords with the RockYou database, including toggling characters for case, number substitutions, etc.	`./hashcat64.bin -a 0 -m 1000 --username -r /rules/toggles3.rule /ntpass.out /Wordlists/rockyou.txt`
Test passwords with a large dictionary and incrementally postpend up to four characters	`./hashcat64.bin -a 6 -i -m 1000 --username /ntpass.out /Wordlists/2B-probable.txt ?a?a?a?a`
Test passwords with a small dictionary and prepend exactly four characters	`./hashcat64.bin -a 7 -m 1000 --username ntpass.out ?a?a?a?a /Wordlists/Top95Thousand-probable.txt`

Four example password tests where, using a progression of hashcat attack types.

In slower and redundant hardware scenarios, you can perform the brute force on one system in parallel with the other tests on another system.

Explanation of the command lines:
General
-a = attack mode (the specific method Hashcat will use to test the passwords)
-i = incremental (for masks, it will try the first, then the first 2, and so on until testing the entire mask
-m = format of the password hashes (1000 is active directory)
-r = Rule to be used by Hashcat

ntpass.out is the list of hashes to be cracked

- This tests all possible passwords from 1-8 characters
- This uses the rockyou wordlist and a built-in ruleset that modifies the strings in the wordlist, in order to test variations. (such as changing ‘o’ to ‘0’)
- This takes a wordlist of 2 billion possible passwords and incrementally appends 1-4 of any characters
- This takes a wordlist of 95 thousand possible passwords and prepends every possible combination of 4 characters.

Slide 17

POSITIVE OUTCOMES

FINDING A METHOD IN THE MADNESS

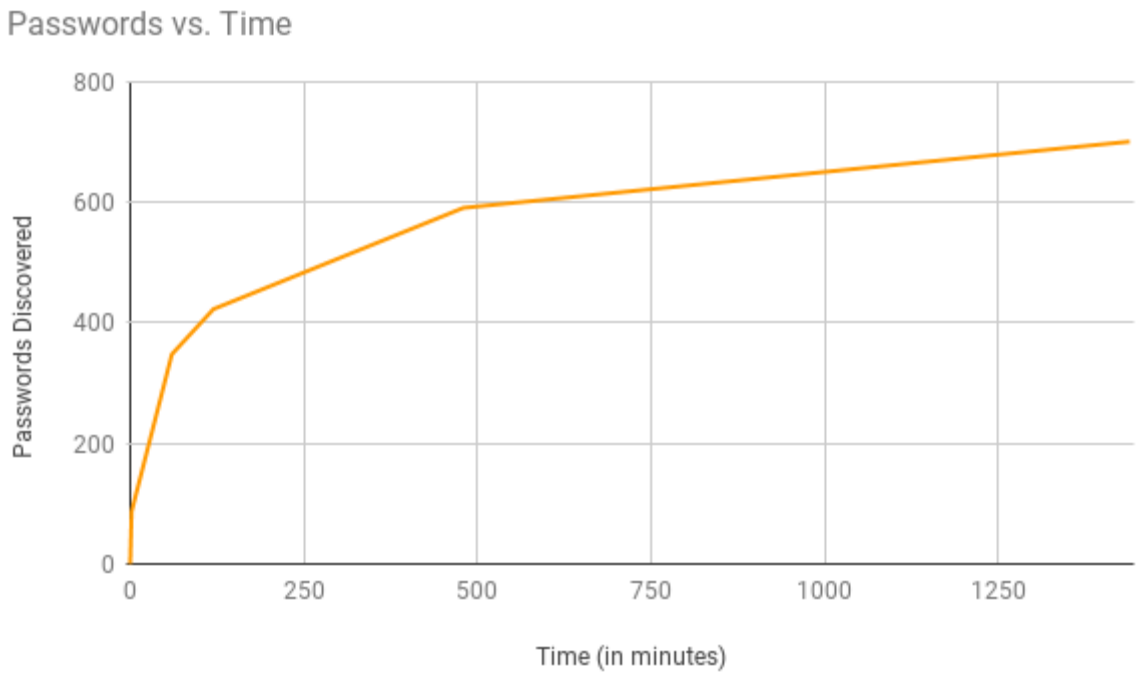
PURESTORAGE®

Slide 18

MOU7

Passwords discovered over time

All the good stuff happens at the beginning



PURESTORAGE®

After aggregating the data, based upon time to discovery

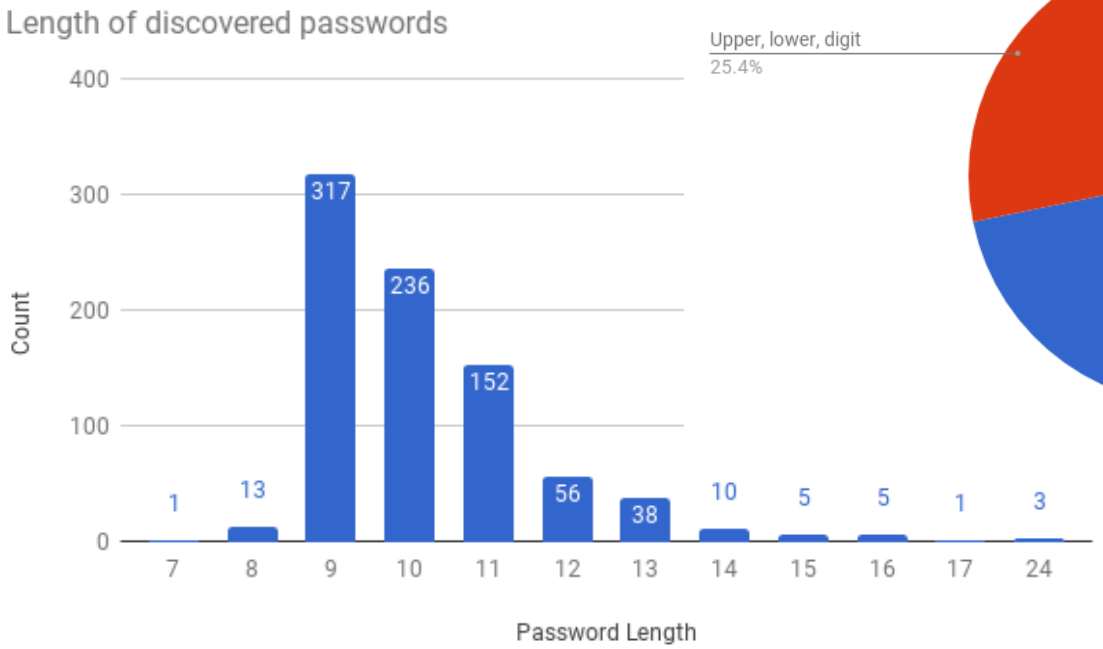
X axis is time in minutes
Y axis is number of passwords discovered

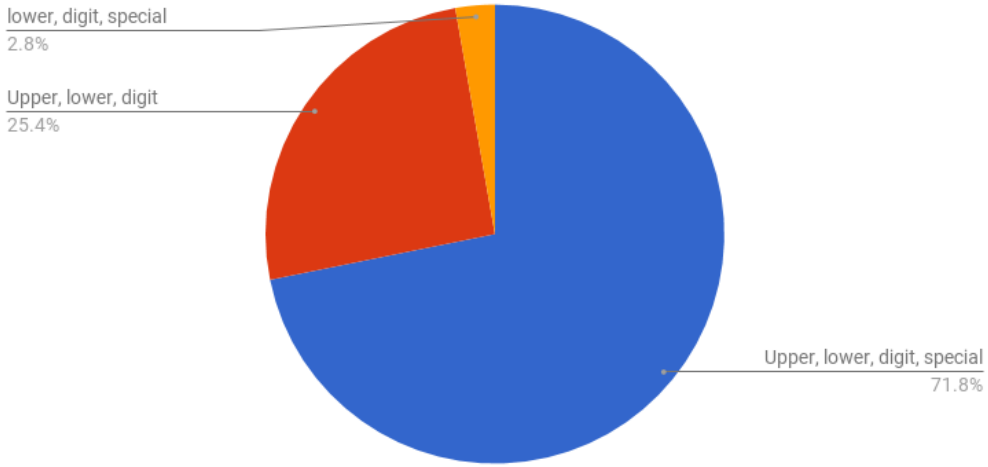
At the 2 hour mark, we've discovered about half of what we'll find in the total of 7 days (this actually assumes a different order than what we really do)

Slide 19

Password Composition

Length doesn't equal strength, but it helps! Poor placement of special characters is merely wasted effort.





PURESTORAGE®

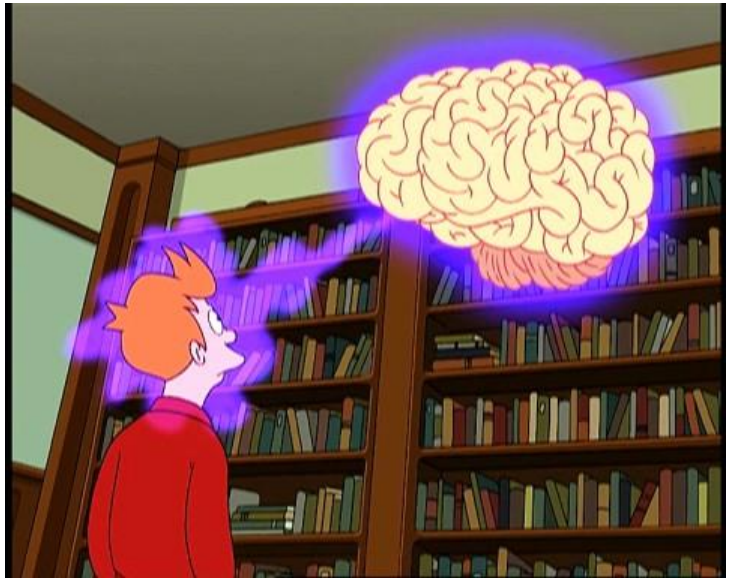
9 character is the minimum, so this is somewhat self-selecting
Even by using all 4 of the “complexity” characteristics, we discover these passwords, so merely having these is not a measure of strength

[] (change these to the global stats)

Common Passwords

Minds think alike

Rank	Password	Count
1	1qaz@WSX	63
2	Ch1nat0wn	27
3	Purit@n1	26
4	M1ss1ngCat	25
5	PureStorage#1	23
6	Pure!12345	20
7	Mousey09	16
8	Pure123	14



PURESTORAGE®

The most common passwords after two years of testing passwords. Our employees really love their company!

Password Similarity

Literally the definition of insanity.

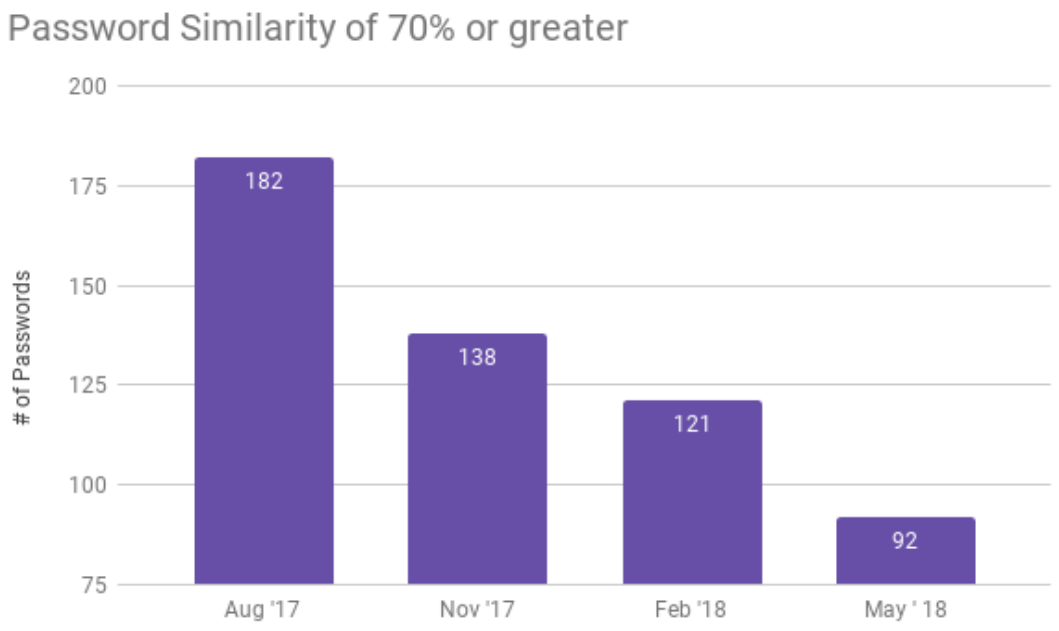
Examples

95% Similar:

- Panther22
- Panther22@

70% Similar:

- Davealice777
- Davealice888



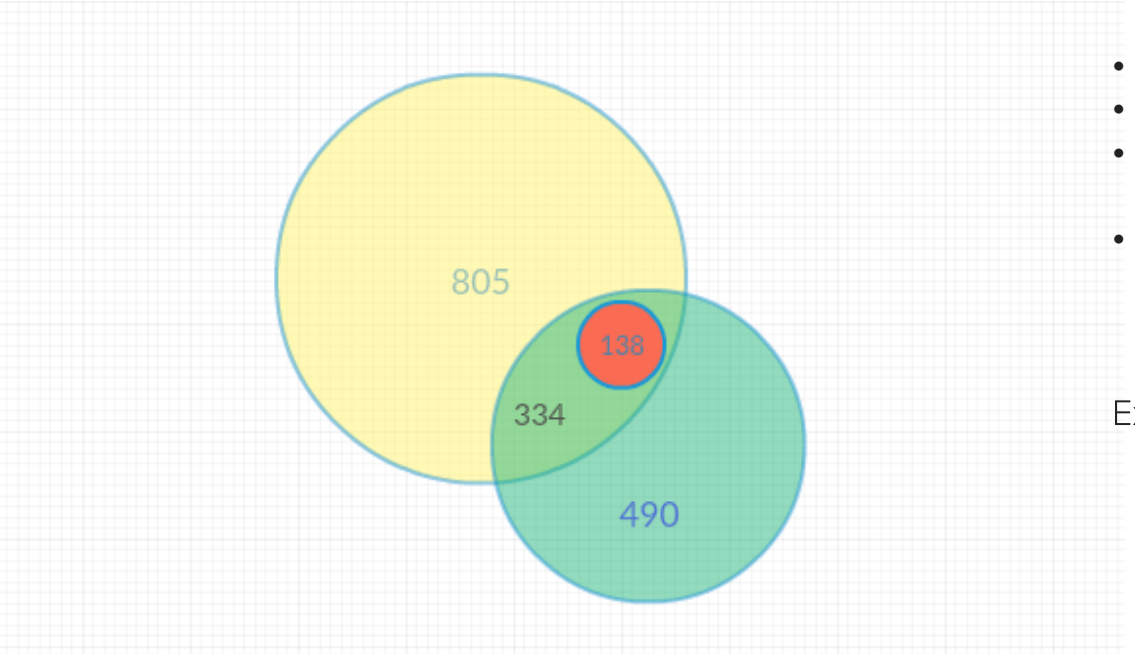
PURESTORAGE®

After running this a few times, we discovered that some people were essentially creating the same password, over and over
I wrote a script (included) that calculates the similarity between your previous and current password.

If it is greater than 70%, we send you an email stating the similarity and **include the old password in the email**.
Seeing one's password generates a visceral reaction, but it has also resulted in a dramatic reduction of recurrence.

Also, we check identical hashes in multiple accounts owned by the same person and force a change for both, regardless of strength.

Repeat Offenders



- 805 – Discovered Aug 2017
- 490 – Discovered Nov 2017
- 334 – People failing in both runs
- 138 – passwords that were basically the same both times

Example: Blizzard1 – Blizzard1!

I compared the passwords discovered in November against those in August for recurred identifications. Of the 490 in November, there are 334 accounts with a failing password that were *also* in the August discoveries. Of these, 138 are using a password that is substantially similar to the previous one, with a similarity rating of 70% or greater.

Don’t Forget – Policy, Not Technology

Use MFA– Skip SMS

Does anyone here ever have to speak with their internal or external auditors?

Your SOX or compliance officers will get hung up on “policy” but mean Active Directory *Group Policy*. Communicate to them that those settings are merely a way to encourage

Also: keep in mind that the password testing is only one part of the puzzle. Organizations should also have:

- multi-factor authentication
- minimum settings in AD or application
- Monitoring for aberrant behavior

Thank You!

It's been a pleasure. Really. Not even joking here.

Contact:

- Kevin Neely
- @ktneely on the Twitters
- Slides, notes, and scripts to get you started

https://github.com/ktneely/password-audit