Password Correlation Analysis

by Jonathan Staples

Disclaimer: This analytical technique is theoretical and should be treated as such. I make no guarantees as to potential accuracy of this method or legality of collecting/retaining data necessary for analysis; take this for educational purposes only.

Overview

At this point in time, late 2017, we live in a world where massive data breaches are commonplace. The Equifax breach impacts nearly every adult, and breaches of user credentials at large sites like Yahoo, LinkedIn, and MySpace have provided hackers with a treasure trove of credentials to use for account takeover attacks. By most accounts, data breaches that expose user credentials are bad, but as with most things this assessment boils down to a matter of perspective. The massive numbers of user credentials available, nearly 4.8 billion according to Troy Hunt's "Have I Been Pwned" data, is great for hackers. It may also be useful for security researchers both in terms of analysis of passwords in general, and potentially linking handles based on password reuse. In this article I'm going to focus on the latter as I suspect it may be helpful to law enforcement and others who are trying to identify individuals by linking online handles together to build a more complete picture of person or group.

Password Theory

"Hold up, are you saying we should be looking at the passwords instead of handles?"

Yes and no. The basic premise of a password is that it should be something that is unique to the user so that not everyone is using the same thing for a password. Unique passwords prevent easy cracking of your accounts via dictionary attacks, and many sites and services require certain things in your password to increase the length or complexity. This is done for your own protection, but in the process it pushes users towards creating something that is unique to them. Despite the benefits to a strong password, there are some people who will still use generic looking passwords as this list of common passwords shows:

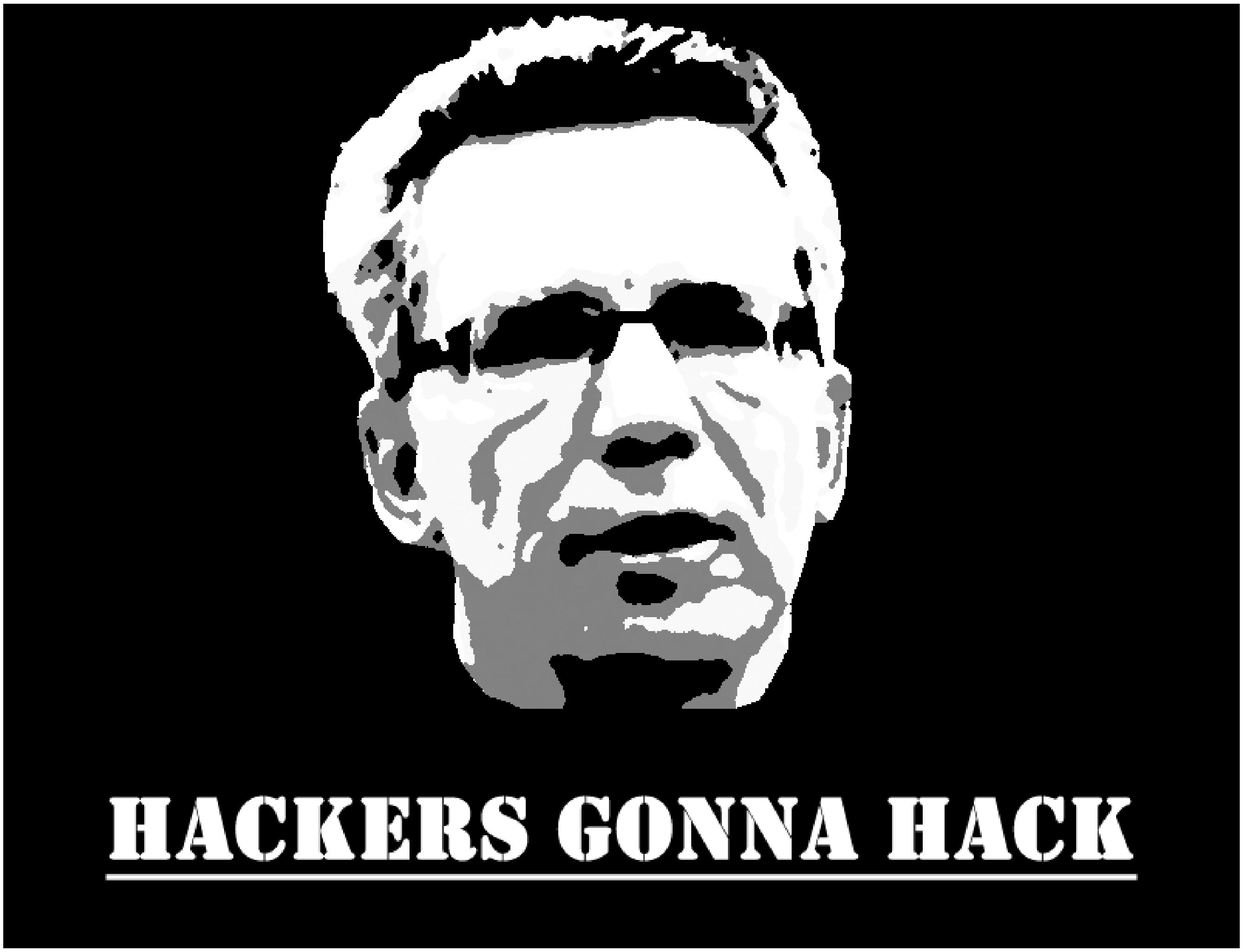
- 123456789
- qwerty
- 12345678
- 111111
- 1234567890
- 1234567

- password
- 123123
- 987654321

For the curious, that was a sample taken from a study of the top passwords exposed in 2016, not data from 1999. As an aside, we don't know how many of those passwords belong to bots and generic accounts so there may be hope that they aren't being used by real people, but even if they are those aren't the passwords we're concerned with for this article. We want to focus on the unique passwords, whose complexity make it unlikely that they will be reused by others. Passwords like “18atcskd2w” or “3rjs1la7qe” are what we would be more interested in when seeking to link accounts via passwords.

Preying on Human Weakness

Let's not mince words here, for this to work we are relying on two basic principles and both take advantage of the fact that people are flawed. The first is that people are lazy, and when it comes to passwords that laziness is an issue because people will reuse the same passwords in multiple places. This fundamental flaw is why account takeover via credential stuffing works so well: the largest contributor to your account being compromised is likely that you used the same password on some other site you probably forgot about and that other site was popped. Borrowing on the idea that the bad guys are wolves and the good guys are sheepdogs protecting the flock, the sheepdog too can use that laziness to our advantage.



The second principle is that, as Thomas de Maizière somewhat put it, hackers are gonna hack. Relevant to us is that hackers often have no real loyalty and are gonna hack other hackers, hacker forums, dark web marketplaces, and other points of interest. Credentials and other data stolen in these hacks often also find their way to the web just as data from LinkedIn did, meaning that in addition to credentials of normal people we are likely to find credentials of hackers in these lists somewhere. So if we take that there are at least 4.8 billion credentials floating around, and these are just the credentials that were databased by Hunt, we can safely assume that at least some of these credentials belong to hackers, criminals, terrorists, and other people or groups of interest. Understanding these two flaws allows us to leverage those to our advantage and go hunt wolves instead of waiting for them to come to us.

What Are We Looking At?

With no consistent standard of password storage we see all kinds of things in the wild – passwords stored in plaintext, encrypted with weak algorithms, encrypted with strong algorithms, and some methods that are too crazy to consider trying to analyze. Going from an attacker perspective, here is a sample of what we might see in a data dump online (actual data found on Pastebin.com):

```
IVINAIREN@GMAIL.COM:80767d8fb8a088b550fd17252e78881d
ucclesya@gmail.com:7c29ea38a59f4bf911610e0e708be120
olenka_b88@bigmir.net:d056f9eeabc7fba9a3309639ac8ea434
natakom@ukr.net:ba9daa79030ac6ff51330390edd2447f
```

That is a sample of a simple combo list, or a list of usernames & passwords, where the passwords are encrypted using the MD5 algorithm. Lists like that are easy to find and even large data sets, like LinkedIn, are often encoded with weak algorithms or have a shortcut that allows the password to be cracked quickly. For example when Ashley Madison was popped in 2015, the passwords stored in that database were encrypted using bcrypt, a strong algorithm. Researchers were still able to crack them by attacking the login tokens instead which were only protected by the MD5 algorithm, enabling them to crack the passwords for more than 10 million accounts in a mere 10 days. Deriving the plaintext variant of a password is essential for our comparison, as different sites may include different processes in their password hashing algorithm, making it difficult to compare hashes directly with a high degree of certainty.

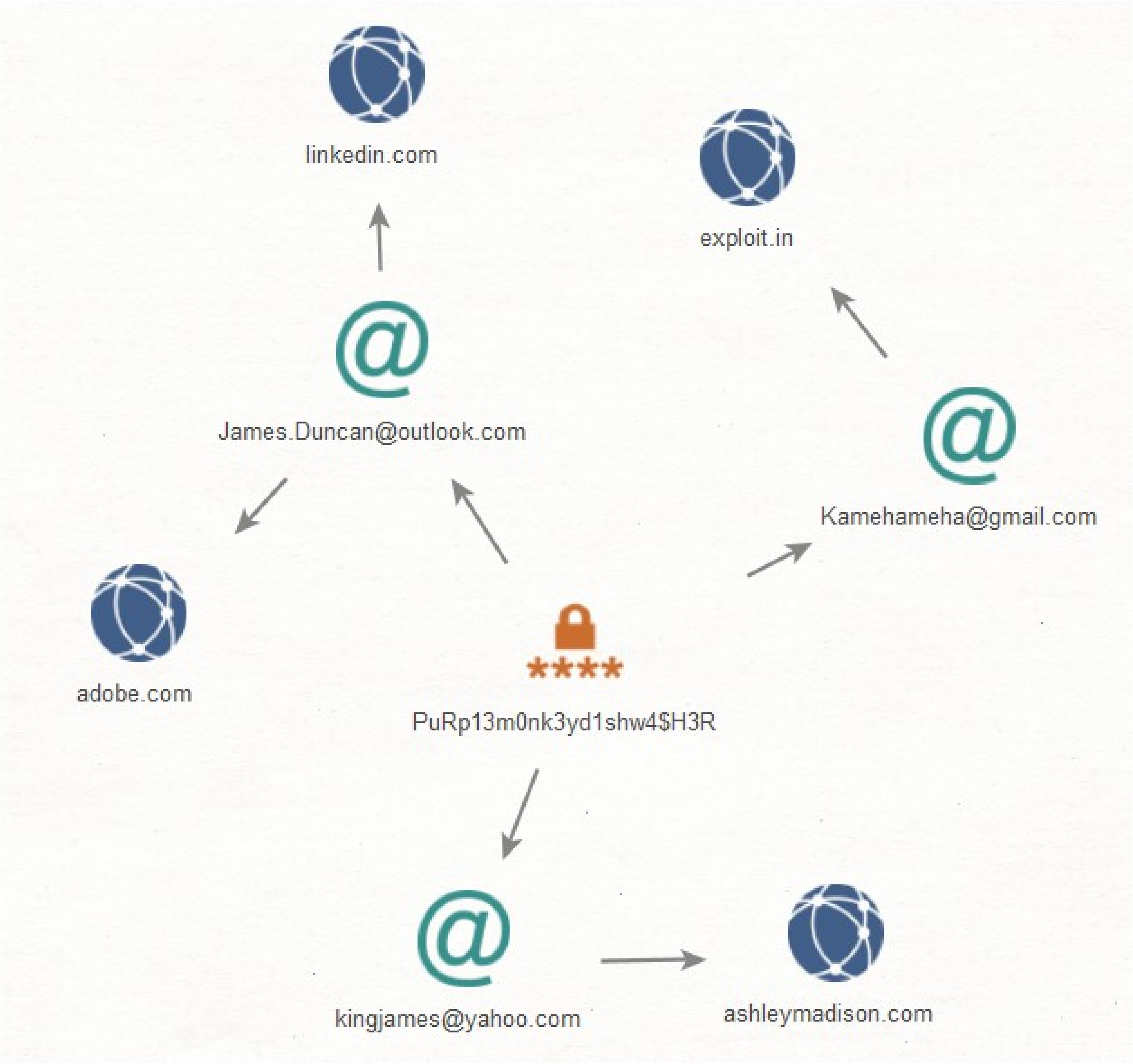
How you get to plaintext passwords is beyond the scope of this article and really depends on what resources you have available or are willing to commit to the topic (ex. Do you crack them yourself or look for versions that are already cracked?). More importantly, there are some legal questions you should probably ask yourself before you go any further because you can quickly get in trouble if you just start compiling breach data and cracking passwords. There is a certain level of risk involved beyond this point, so exercise caution.

We're Good With the Risk, Now What?

Now we're going to proceed with the assumption that you have some sort of database that contains a list of usernames, the breach that username came from, and the password associated with that breach. From here you can take one of two approaches:

Method 1: Targeting a Specific Handle

In this case the assumption is you already know a specific user handle who you want to get more information about, and this person may have great OPSEC so you have no leads. For instance, let's say we have this user known to have logged in to Exploit.in with the handle Kamehameha, and his email address is kamehameha@gmail.com. His OPSEC is great and his password is both complex and unique, making it a great candidate for analysis. Pivoting off the password in our database, we look for other breaches that included the same password and extract the user info from that query. As an example, we might find the following results:



Well we might be getting somewhere now, our resulting diagram shows the same password was used for accounts that were involved in several other breaches. As an analyst I could make the argument that the target's real name is James Duncan and his professional email address is james.duncan@outlook.com as the email address and password were the same for sites that are typically used for professional reasons. Trying to figure out if the other accounts are his as well, we can look at it from two directions. Looking at the graph from one direction, Kamehameha was a Hawaiian king, so the kingjames@yahoo.com account could be a combination of his real name and his hacker moniker. Looking at it from the other direction, we could argue that the target views himself somewhat highly by applying the “king” title to his own name, thus it makes sense that he would use the name of a king for a different account. We now have a data correlation point in that the same unique password was used for all three accounts, and a contextual correlation in the names the target is choosing to use.

If we look closer at those accounts, we might pick up some additional leads including alternative email addresses, profiles, forums, or other resources we can now investigate. Outside of our data set we can go look at domain registration, forums, and other websites to build a more complete picture of the individual to see what things make sense for that person and weed out the things that don't. We can't guarantee that it's the same person behind all those accounts, but if we want to be more like V we should not believe in chance and do some actual analysis. Besides, we've already said we were out of leads earlier so why not have a look?

This method may work well for individuals and small groups, but not necessarily for large groups of accounts as it may generate too much noise to effectively analyze.

Method 2: Shake the Tree and See What Drops Out

Perhaps a less useful time investment, though arguably more interesting, would be to basically shake the tree and see what falls out. By this I mean to compare all your passwords and organize them in order of the most common. Then skip the top few because they are most likely bots, generic accounts, and a group of accounts that are likely to be unrelated or at least have too much noise to be of any use. Once you get down your list to the more interesting passwords it can reveal some interesting data including:

- Identifying people who have multiple accounts at the same sites
- Identifying people who are active across a variety of sites

- Identifying groups of people who share the same passwords
- Identifying people who use similar passwords but have minor changes (you can use REGEX queries or wild cards to help figure this out)

and of course:

- Finding funny passwords for the lulz

Sounds Good, But Why?

Now that we've stepped through the method a little we come to the question of how is this useful, to which I offer the following:

For Offensive Players

If you're on offense, this can be a useful tool in furthering an investigation or intelligence collection against a target. Law enforcement targeting a specific hacker or other individual can use this method to potentially identify additional leads in their case that they might not otherwise have if the suspect is using good OPSEC. This is a funny thing about OPSEC in a lot of cases – many people focus on OPSEC in terms of their username/email address/handle, and assume that since no one sees the password then the password is a safe element, not a point of possible identification. Also, not many people are thinking about password correlation as a means of analysis, so the risk of reusing the same password is currently fairly low in terms of having your password give up multiple identities. Going a step further, if you collect multiple password samples for a targeted individual it can give you an idea of how they construct their passwords or provide clues to their personal life, which can also be helpful in an investigation.

For Defensive Players

Props to you for making it this far in the article and thinking about this from a defender perspective. In this case you have a distinct advantage: you probably have a database of users and passwords so you can apply these techniques to your own data set without having to worry too much about whether or not you have a right to that data. You may even be able to compare password hashes to find matches as you would know your own algorithm, so you don't have to break the data back down to a plaintext format (avoiding those risks as well). As a defender, you may be able to do some correlation to discover:

- Accounts created by bots that use the same passwords
- Accounts involved in fraud that use the same password as a legit user, potentially linking the two
- Common passwords that satisfy your password requirements – which you can then blacklist to prevent users from using since they are common and/or easy to guess

- Find users who have created new accounts, allowing you to potentially identify orphaned accounts and nominate them for closure

At the end of the day…

How and why you might use this technique depends highly on you, your organization, what information you have access too, what your goals are, and a number of other factors. The purpose of writing this was to point it out as a possible technique, “another tool for the old tool bag” so to speak, and in a sense serve as a word of caution. If the data to be analyzed is easily accessible, and indeed a lot of it is, and the technique may have some benefit to offensive players, then we have to remember that the wolves are on offense as well and that should impact our own behavior. Foreign intelligence services, hackers, and other wolves may try these same techniques to identify you at some point. Obviously using the same passwords across sites is just bad practice, but if you are doing it then know you are at risk of having all those accounts linked at some point. Whether or not that's something you want to worry about I leave up to you.