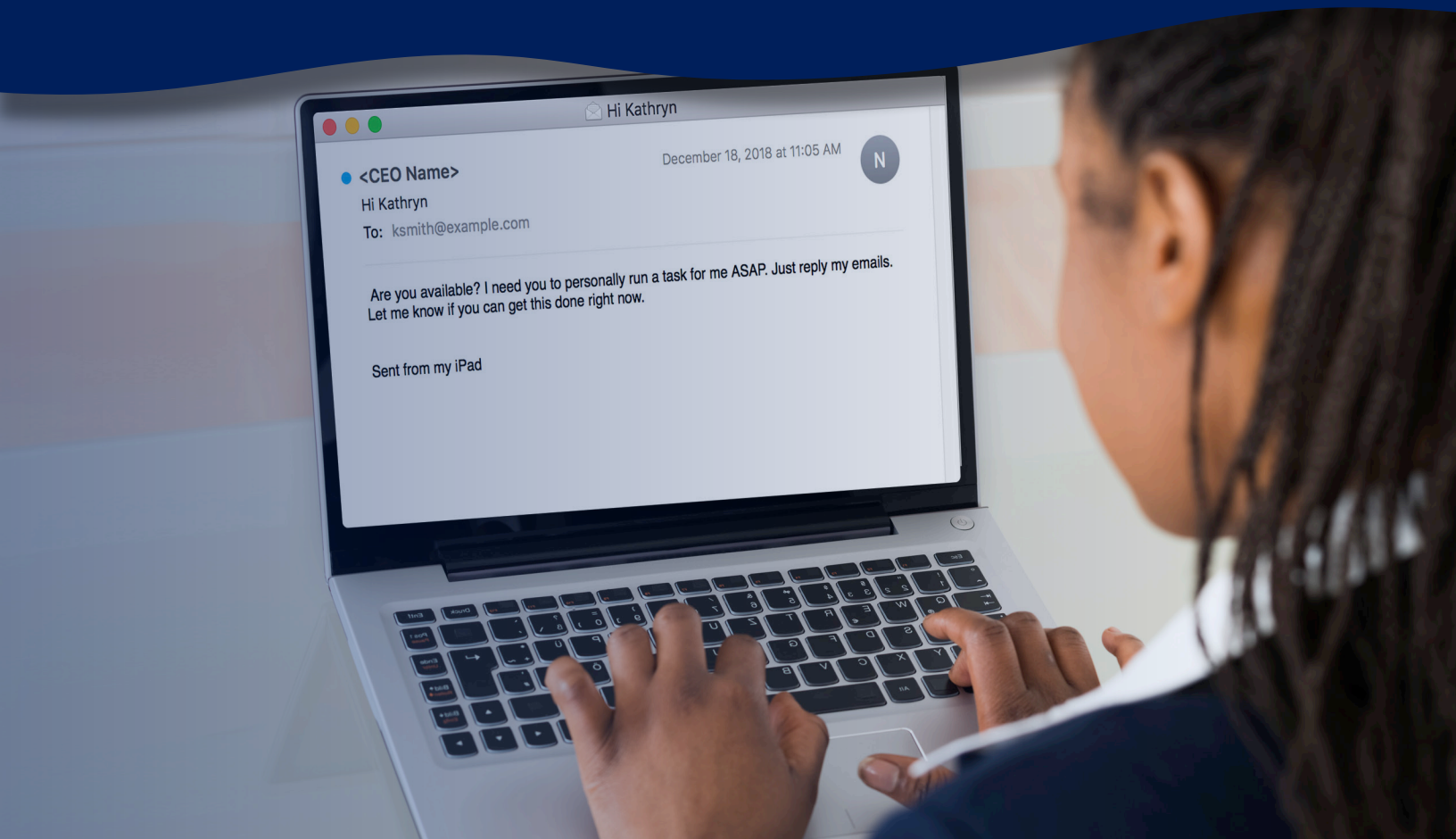
UNDERSTANDING PHISHING

INKY®

Account Takeover

One of the most pernicious forms of phishing involves account takeover. In these attacks a bad actor harvests, guesses, or brute-forces the password for an email account, and then uses this account to impersonate a person or brand... often with devastating results. In this report we examine real-world account takeover scenarios, why they are difficult for SEGs to identify, and how INKY catches them using sender profiling and stylometry.



Looking for a Few Good Email Accounts

What's an email account worth? Since there are over four billion of them worldwide, you might assume "not very much". But it's important to recognize that not all email accounts are created equal in this regard.

A random Gmail account? Maybe not so valuable. But a high-reputation corporate or .edu account? Solid gold in the hands of a sophisticated attacker.

In earlier installments of our *Understanding Phishing* series we've covered credential harvesting attacks, where attackers send fake branded emails directing victims to real-looking login pages for cloud services like O365. In this report we'll look at how the attackers use these credentials to conduct particularly effective phishing attacks. And, fair warning: these are both the most damaging kinds of phish and the most likely to get past Secure Email Gateways.

> ***Attackers use credentials for high-quality email accounts to effect devastating phishing attacks that are very difficult for Secure Email Gateways to detect.***

Let's first how the bad guys execute an account takeover ("ATO") phishing scam.

INKY™

Oh No! ATO

First recognize that every phishing email has to come from some source email account: some mail server has to initiate an SMTP session with one or more other mail servers to cause the phishing email to get delivered to its intended victim. So the phisher needs some mail server to send his evil emails.

Where can he get this mail server? Broadly, he has three options: he can set up his own mail server, he can use a free shared mail infrastructure like Gmail, or he can take over someone's legitimate account on their organization's mail server. At INKY we see vast amounts of all three kinds, but the latter ATO type is in many ways the nastiest, for several reasons.

For years SEGs have been using reputational information to block spam, malware, and phishing. Things like DNS block lists track servers known to send bad mail and are updated in real time based on end-user spam reporting. But an attacker using an ATO account hosted on, say, a corporate mail server benefits from the stellar reputation of that server. No corporate mail server with a competent admin will remain on a DNS black list for long, because its presence there will harm the company's reputation and ability to communicate.

ATO accounts can be especially devastating when the attacker not only relies on the good reputation of the compromised account, but also uses it to impersonate the person who legitimately controls the account. Imagine receiving an email from a person at a vendor you know well, making an unusual request: you may be talking to an impostor! This case is especially hard to detect because the email really does come from the legitimate sender's account. Later we'll explain how INKY uses stylometry and other sender profiling mechanisms to identify these.

Attackers use ATO to evade SEG reputational filters and even impersonate the specific people whose accounts they've compromised.

Another aspect of ATO that's important to keep clear is: *whose* account is being taken over? The key distinction for security professionals here is *my own accounts* vs. *other organizations' accounts*. The correct way to protect your own email accounts from being taken over is to use multi-factor authentication (MFA). If you do not already implement and mandate MFA for your own email accounts, stop reading this and do so immediately! It is foolish to rely on a phishing detection tool to detect this kind of "first party" ATO after the fact, when MFA with an authenticator app can completely prevent it.

It's *third party* ATO that you need to detect after it's occurred. This is because while *you* control authentication to your email accounts, you can't realistically force all the third parties you interact with to implement MFA or similar controls.

The primary ATO threat you face is attackers using compromised third party accounts to phish your users. Make sure your SEG or mail protection properly distinguishes first-party and third-party ATO scenarios, and be sure you understand what controls you have in place for each.

Before we dive into some real-world examples of third-party ATO in action, a brief aside on how attackers compromise accounts in practice.

INKY™

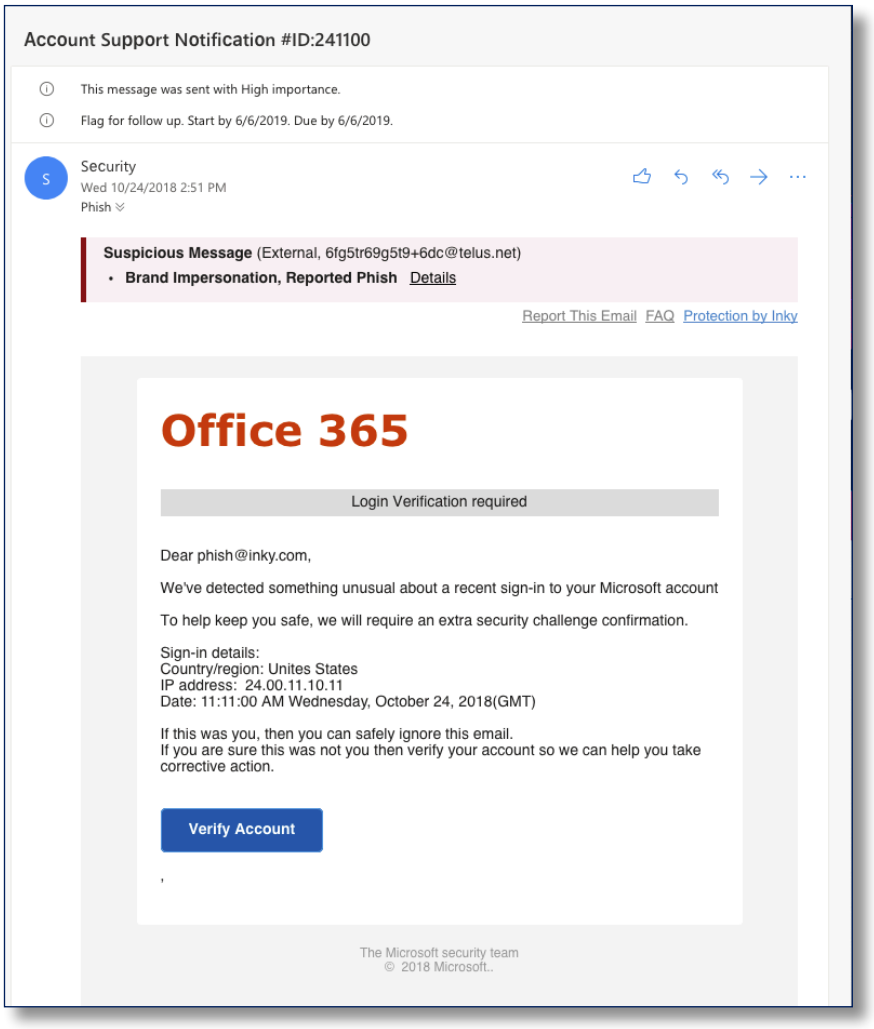
I Can Haz Password?

If you want to make it easy for a bad guy to take over your account, use a bad password. Even an incompetent attacker can breach an account with an easily guessed password like 123456 or password1. National Institute of Standards and Technology (NIST) even maintains a list of bad passwords to avoid and JavaScript code to check it.

More sophisticated bad guys will use *brute force* or *dictionary-based* attacks. These involve simply trying to log in over and over again with a large number of passwords generated from patterns or dictionaries. In fact, attackers use the bad passwords list themselves to compromise accounts secured by these passwords! As a result, *all password-based authentication systems* should enforce a delay between attempts, along with a maximum bad login limit.

Open source packages like zxcvbn by Dropbox can actually estimate how long a given password would take to brute force; the most secure password-based authentication systems will incorporate metrics like these and disallow users from choosing easily brute-forced candidates.

By far the easiest way to get a user's password, however, is to fool them into just telling you what it is via credential harvesting. The most common case is a fake cloud service login page. A phishing mail like the one below leads the victim to a real-looking O365 login page that is in fact hosted by the attacker, and logs all the email address and password combinations people enter into it. We cover this topic in more detail in *Understanding Phishing - Credential Harvesting*.



Attackers can also grab passwords by sniffing them out of network traffic or via key-loggers that watch what users type, but as these methods installing malware behind the firewall, they are harder to execute and more likely used by actors with significant resources, and for specialized, highly targeted attacks. Bottom line, though: there are several easy ways for criminals to get passwords.

Assume attackers can easily obtain the passwords for your own users' accounts and for the accounts' of third parties your users exchange email with. Implement MFA to prevent takeover of your user's accounts when the password is compromised.

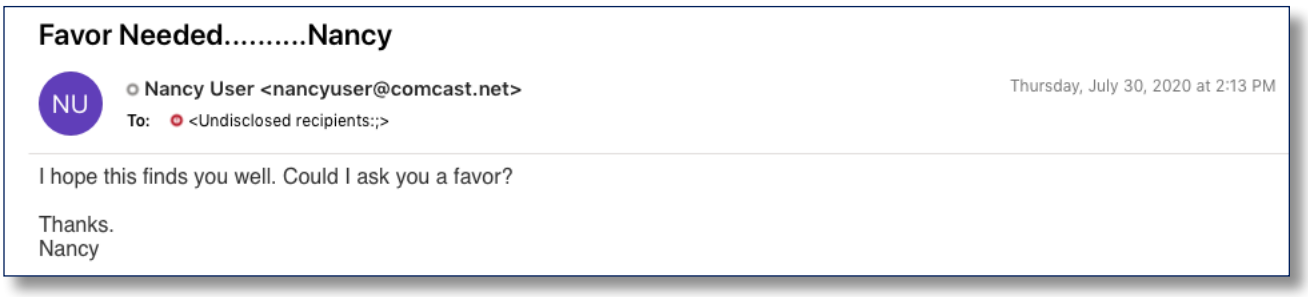
Third Party ATO

We've now looked at how and why attackers compromise email accounts, and how to protect your own users from ATO with MFA. That leaves third party accounts: how do we protect ourselves from *other people's email accounts* getting taken over?

This is where a phishing prevention solution like INKY is crucial; while we can't stop third party ATO phish at the source, we *can* recognize them on the way in from the internet and block them before they hit the intended victim's inbox.

To understand how INKY identifies third-party ATO emails, we'll now examine a few real-world examples that INKY has caught in detail. As always, we have not modified these in any way other than to redact PII.

One of our users received this mail sent from a person named Nancy whom she knew well:



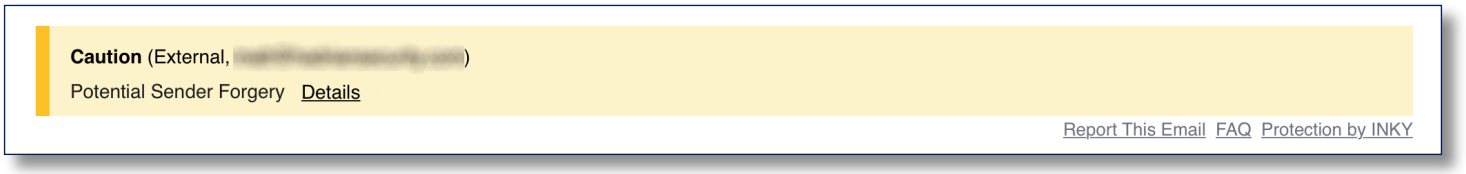
You can see immediately the challenge these kinds of third party ATO emails pose: there is very little content here to analyze, and the victim is predisposed to think the mail is legitimate because it's from the account of a familiar contact.

Furthermore, when an email comes from the impersonated individual's actual account like this one does, the mail headers are generally of little use, since they'll look the same as they do for legitimate mail (with a few possible exceptions).

Cases like these are where we need to rely on *stylometry*. Originally developed to determine authorship of disputed manuscripts — "Did Shakespeare really write *Edward III*?" — stylometry uses stylistic features of text like vocabulary, sentence structure, and even punctuation to model an author's typical writing style.

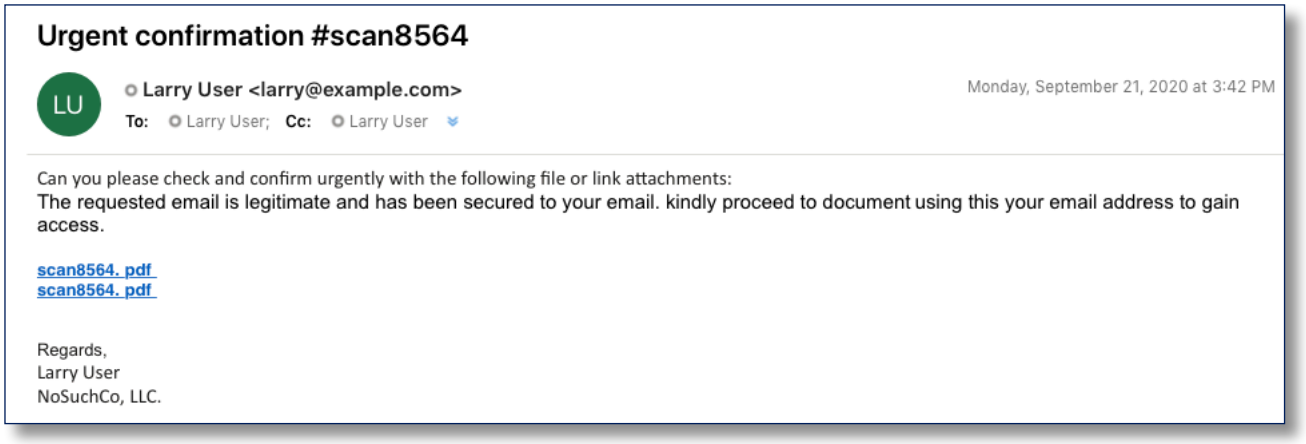
INKY uses both header features and stylometric features to create and maintain a profile of each sender whose email it has ever seen. This profile, in essence, captures what that sender's "normal mail" looks like.

While it's often difficult — in some cases theoretically impossible — to *prove* that a mail is or is not from the claimed sender, INKY can often develop a pretty good *heuristic* sense: while it can't necessarily be sure, INKY can develop a strong enough belief that an email is an outlier that the mail warrants a yellow warning banner telling the user the mail is a potential sender forgery:



INKY looks at many different stylometric features, but one that stands out here is the sender's use of unusual punctuation in the subject line; it's likely the real Nancy has never sent mail with lots of periods in the subject line. Similarly, the attacker's lack of greeting and "Thanks. Nancy" closing text may differ from what the real Nancy usually uses. Finally, at least one header feature may be relevant here in INKY's determination that this mail is probably not really from Nancy: the `To:` line is empty. (Note that mail clients often display this as "undisclosed recipients" rather a blank `To:` line.)

Here's another example of a third-party ATO email INKY caught:



Urgent confirmation #scan8564

LU

Larry User <larry@example.com>

Monday, September 21, 2020 at 3:42 PM

To: Larry User; Cc: Larry User

Can you please check and confirm urgently with the following file or link attachments:
The requested email is legitimate and has been secured to your email. kindly proceed to document using this your email address to gain access.

scan8564. pdf
scan8564. pdf

Regards,
Larry User
NoSuchCo, LLC.

Like the previous "fake Nancy" example this "fake Larry" example has many of the hallmarks of third-party ATO phishing: in particular, it has urgency written all over it, so to speak; includes stilted language and poor punctuation; and links to URLs which are probably hosting malware (but, importantly, which weren't on any threat list when INKY received this zero-day). There's even a strange font change, and the fact that the mail is both `To:` and `From:` Larry. Despite these signals, this mail went through the upstream SEG unscathed (in this case, Barracuda).

INKY, by contrast, was skeptical that this was really from Larry because the stylometry varied significantly from that in Larry's normal emails. Sort of like we might doubt the authenticity of that Shakespeare play, INKY uses similar techniques here to shed light on the questionable authorship of this email.

How INKY Solves It

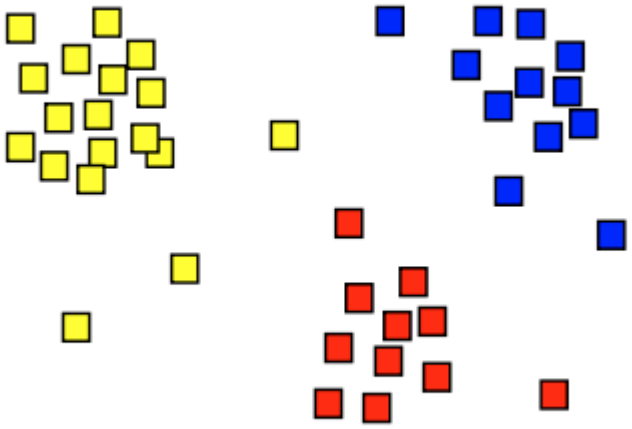
Hopefully by now you have some intuition about how INKY uses stylometry and other features from emails to build sender profiles and catch attempts to forge mail from known users — even when those forgeries come from the real senders' actual email accounts.

This still doesn't really explain *how* INKY uses these features to make a determination about a given email. In this section we'll introduce the concept of *clustering* and explain how INKY applies this machine learning technique to identify third-party ATO attempts.

Clustering is a machine learning approach that allows us to build models that can intelligently group data points together in ways that are meaningful to humans. Consider trying to create software that recognizes handwritten digits like these from the MNIST data set:

Notice that while all these digits look different, each row simply has different renderings of the same digit; we can say that each row represents a cluster, each corresponding to a particular digit. Essentially, each image in a row is more similar to the other images in its row than it is to images in other rows. We capture this notion of similarity with a *distance metric:* the closer together two images are according to this metric, the more similar they are.

If this sounds abstract, it is: finding a good distance metric that captures the essence of what we humans perceive as digit similarity is tricky, and part of the art of using clustering is picking just the right "features" to extract from the examples so that the "metric space" these features establish works well for what you are hoping the software will learn. While it's impossible to summarize how this is done here, this visual from the Wikipedia page on clustering may help:



Imagine that each digit in the MNIST set corresponds to one of the colored squares in the image above. Notice how the yellow ones form a sort of cluster? Think of those as the MNIST one digits. Likewise, think of the red squares as the MNIST four digits. The yellow and red squares very near other tell us that under this distance metric, there's

a one digit example that is very close ("similar") to a four digit example. Look at the ninth 4 from the left in the MNIST data image to see what example four digit it might be.

Now here's the mental leap of faith you need to take: while the cluster map above is two dimensional — each square has a vertical and horizontal coordinate, and the clusters are areal — in real machine learning applications the clustering happens in much higher-dimensional metric spaces. Trying to imaging the image above with a thousand dimensions instead of two is impossible for our human brains, but the mathematics of clustering algorithms generalizes to these higher dimensions and works in a similar way. These many additional dimensions come from the many features we extract from each example — in this case, from each handwritten digit image.

Hopefully now it makes sense how INKY uses header and stylometric features to cluster emails. Instead of images of handwritten digits, INKY is clustering emails, with the goal that emails that seem to be written by a particular sender should mostly cluster together. So a "normal mail" from Nancy should appear near other mails Nancy has sent in the past *in this very high dimensional metric space.*

The trick lies in determining *what* features to extract and *how* to use them to map each email onto a point in a space where nearness corresponds to the notion of "similar to mail sent by the same sender". This is where the secret sauce lies in all machine learning approaches: finding features and mappings that work well for a given problem is as much art as science, and can take a lot of iteration.

At INKY we have spent many years experimenting with different features, models, and approaches, and we continue to do so. But at the end of the day, we're simply trying to make INKY learn about email senders and what their mail looks like, so we can spot attempts to forge mail from one of those senders.

INKY™

Conclusion

Account takeover is both pervasive and hugely destructive. Third party ATO phishing scams target your users, are impossible for you to prevent, and are difficult to detect.

INKY has applied recent innovations in machine learning like stylometry and clustering to create individual sender profiles and block third party ATO and other impersonation attempts.

To see INKY in action request a live demo, or sign up for a free trial today.

We're passionate about email.

Want to talk about an issue you're facing in email security at your organization?

Request a demo today

www.inky.com