Ichthyology: Phishing as a Science

Karla Burnett, karla@stripe.com
Security Engineer, Stripe

Phishing is traditionally seen as the easy way out when conducting penetration tests: of course you can socially engineer your way in, that's always possible. This dangerous mentality leads companies to designate phishing "out of scope" when conducting Red Team exercises, and leads Blue Teams into the lax attitude that phishing is inevitable, with training the only defense.

This is a naïve view of phishing to have in 2017. There have been countless damaging phishing campaigns against companies in the last 5 years, from the 2011 RSA master key leak[1], to the hacking of Sony by the North Korean government in 2014[2], to the 2017 Google Docs OAuth worm[3]. Organizations who rely on phishing training as their only defense turn a blind eye to the impracticalities of such training, including how effectively the information is retained, and how realistic it is to train all employees on a regular basis.

This paper will categorize phishing campaigns based on their attack mechanism, then will discuss a series of internal phishing campaigns run against Stripe employees, using these as case studies of the ways in which effective attacks bypass training. I'll discuss the psychology of phishing, and why training is often used to solve the wrong problem, then I'll present a solution in real-world use at Stripe that prevents credential phishing for protected services.

DISCLAIMER

All companies have human weaknesses, and Stripe is no exception. Each of these campaigns was performed against Stripe employees on their work machines. They're described here, along with lessons learned from performing them, so that everyone in the field can benefit.

TYPES OF PHISHING

Before we can begin to discuss the science of phishing, we first need to categorize the types of phishing that are currently prevalent. This is typically done by breaking campaigns up by the types of people they target, with "spear phishing" targeting an individual, and "whaling" targeting an individual with a lot of power in a company, such as a CEO. Unfortunately, this places the focus on the person being targeted, rather than the type of attack being used.

Instead, it is more valuable for both defenders and attackers to categorize phishing by the type of attack being performed, specifically into three categories:

- **Action phishing** campaigns rely on tricking a target into providing information that is in itself valuable, or to take a particular valuable action. An email spoofed from the CEO of a company to another employee, asking them to transfer a large sum of money offshore, would fall into this category.

  Prominent examples of action phishing campaigns include the ICANN 2014 CZDS spear phishing campaign[4], the 2015 Xoom CFO scam[5], and an assortment of payroll[6] and tax[7] phishing attacks that have happened in recent years.

  Action phishing is typically mitigated with training on how to identify a phishing email.

- **Exploit phishing** relies not on any wrong-doing on the employee's part, but on the use of an unpatched machine, or a 0 day vulnerability in software they have installed. An email from a shipping company with a malicious PDF attachment would fall into this category.

  This type of phishing led to the 2011 leak of the RSA master key[8], the 2014 Dyre malware[9], and the 2016 Fedex malware[10].

  Exploit phishing is usually mitigated by ensuring your organizations' machines are kept patched, and limiting the installation of traditionally exploited applications, such as Acrobat Reader and Flash.

- **Credential phishing** attempts to harvest credentials for online services, which can then be maliciously accessed as a direct source of information, or as a method of pivoting from one service to another. An email that pretends to link to a webmail provider, but links instead to a cloned page in order to steal usernames and passwords, would be an example of this type of phishing.

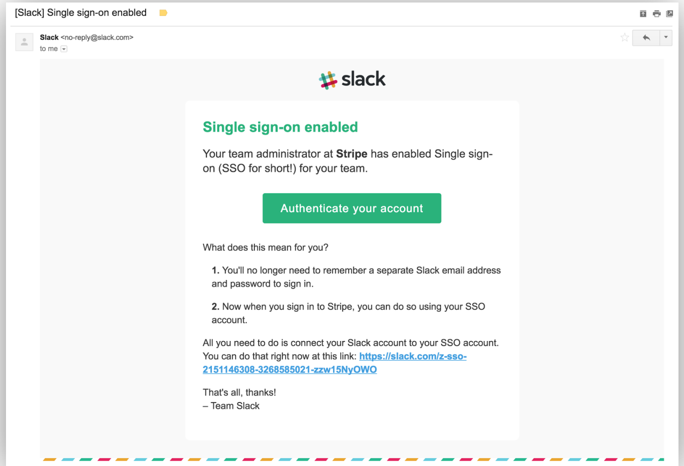
  The 2014 Sony Apple ID phishing campaign[11], the 2016 Hacking of the US Democratic National Committee's Gmail[12], and the 2017 Google Docs worm[13], are all examples of credential phishing in play.

  Credential phishing is often mitigated with training on how to identify a phishing email.

Although each type of phishing presents different dangers to an organization, the least well mitigated by training and a strict machine update policy is credential phishing. Employees enter credentials on a regular basis, and are routinely prompted for them by external services outside your company's control. As discussed later, under the psychology of phishing, the routine nature of this action makes it difficult for training on these actions to be effective. As such, this is the type of phishing this paper focuses on.

CASE STUDY 1: SLACK

Our initial phishing campaign was conducted in very late 2014, as a direct response to the Sony Apple ID attack. We'd recently triggered a legitimate email from Slack to all our users by turning SSO on, so decided to clone that email in order to determine just how vulnerable we would be to an attack similar to the Sony one.



[Slack] Single sign-on enabled

Slack <no-reply@slack.com>
to me

Single sign-on enabled

Your team administrator at **Stripe** has enabled Single sign-on (SSO for short!) for your team.

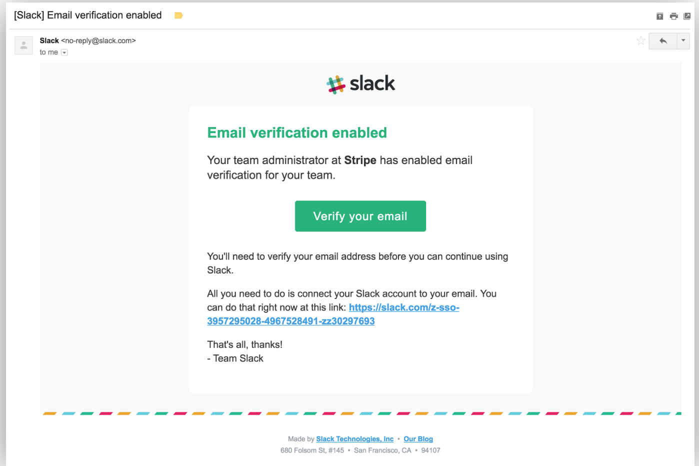
What does this mean for you?

1. You'll no longer need to remember a separate Slack email address and password to sign in.

2. Now when you sign in to Stripe, you can do so using your SSO account.

All you need to do is connect your Slack account to your SSO account. You can do that right now at this link: https://slack.com/z-sso-2151146308-3268585021-zzw15NyOWO

That's all, thanks!
– Team Slack



[Slack] Email verification enabled

Slack <no-reply@slack.com>
to me

Email verification enabled

Your team administrator at **Stripe** has enabled email verification for your team.

You'll need to verify your email address before you can continue using Slack.

All you need to do is connect your Slack account to your email. You can do that right now at this link: https://slack.com/z-sso-3957295028-4967528491-zz30297693

That's all, thanks!
- Team Slack

Made by Slack Technologies, Inc • Our Blog
680 Folsom St, #145 • San Francisco, CA • 94107

Left: original Slack "Email verification" request. Right: cloned phishing email.

We sent this campaign to all current Stripe employees, and saw good conversion rates from the email to the website (epitomized by the VP of Engineering falling for the email immediately after being told it would be sent to him!), but poor conversion once users landed on the website.

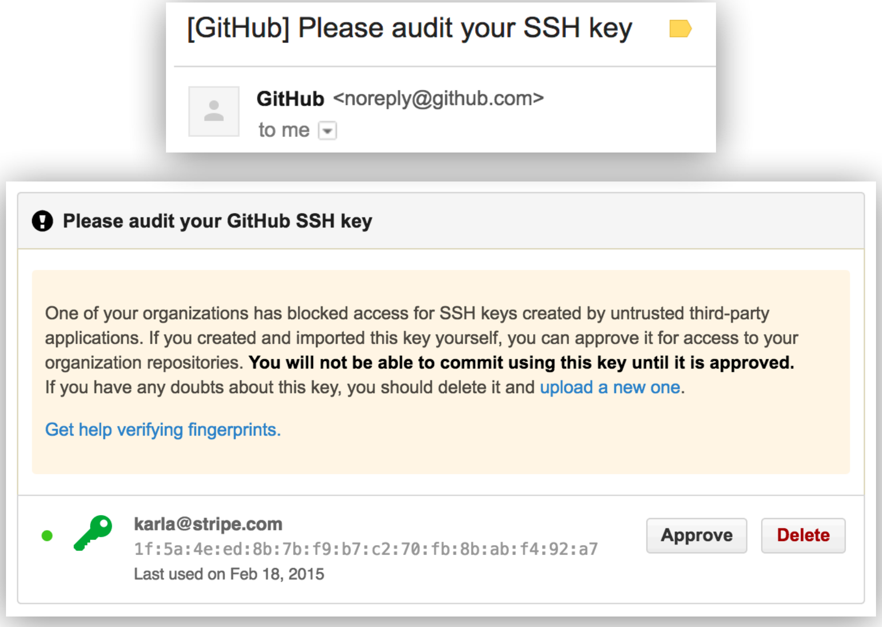
This wasn't surprising - short of vetting the email headers by hand and verifying that the HTTPS link pointed to where it said it did, there was nothing to tell a user that the cloned phishing email was not legitimate. At the time, Slack employed no SPF[14] or DMARC[15] protection on their domain, allowing emails to be spoofed from their legitimate `no-reply@slack.com` address. The email style was cloned directly from the real email, while the content contained no misspelled words, a common tell that people use to identify phishing emails.

However, the low conversion rates on the actual website suggested that users were paying attention - the presence of an incorrect domain, and the lack of an HTTPS connection alerted them to the scam. Unfortunately, while these signals might be work for a poorly executed attack, they would be easy to work around by a motivated attacker.

CASE STUDY 2: GITHUB

With that in mind, several months after the first campaign we ran a second, more well-constructed one. We took what we'd learned from our initial attempt and hosted the phishing page on an HTTPS domain, and in fact managed to host it on a domain controlled by the company whose login details we were spoofing.

We chose to target GitHub for this attack, since there was a clear path from committing code to our repositories to gaining code execution on a production server. We cloned an existing message from GitHub about third party keys, converted it to an HTML email, then scraped users' public keys from GitHub's API to convincingly provide their real SSH key fingerprints in the email.



[GitHub] Please audit your SSH key

GitHub <noreply@github.com>
to me

Please audit your GitHub SSH key

One of your organizations has blocked access for SSH keys created by untrusted third-party applications. If you created and imported this key yourself, you can approve it for access to your organization repositories. **You will not be able to commit using this key until it is approved.** If you have any doubts about this key, you should delete it and upload a new one.

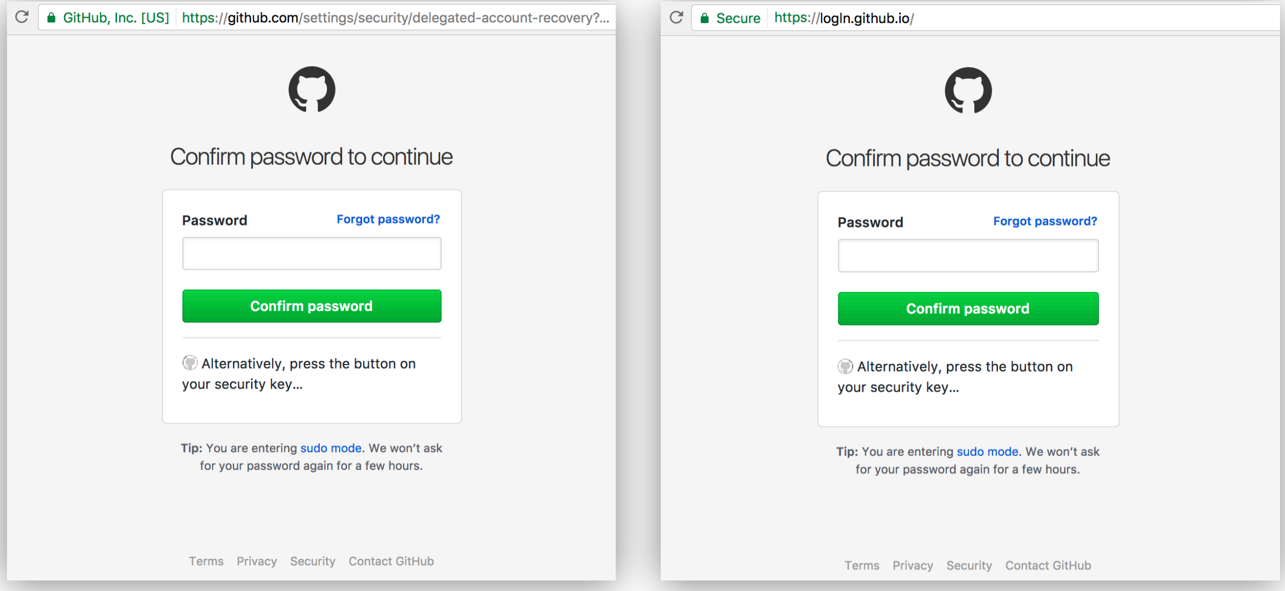
Get help verifying fingerprints.

karla@stripe.com
`1f:5a:4e:ed:8b:7b:f9:b7:c2:70:fb:8b:ab:f4:92:a7`
Last used on Feb 18, 2015

Top: Email subject and delivery information. Bottom: Email content.

We knew from our previous campaign that we'd want to host the phishing page itself on an HTTPS domain. Helpfully, GitHub operate a service called GitHub Pages, which allows users to host basic websites at `username.github.io`. In the past, these pages had been hosted at `username.github.com`, and at the time of this campaign, pages hosted on `nonexistent.github.com` would redirect to `nonexistent.github.io` with no interstitial, for compatibility with old links. While GitHub restricted the creation of usernames such as `www` and `login`, registration of similar usernames such as `logln` was not prevented.

As such, hovering over either the "Approve" or "Delete" buttons in the email would show you a link hosted on `logln.github.com`.



Alternatively, press the button on your security key...

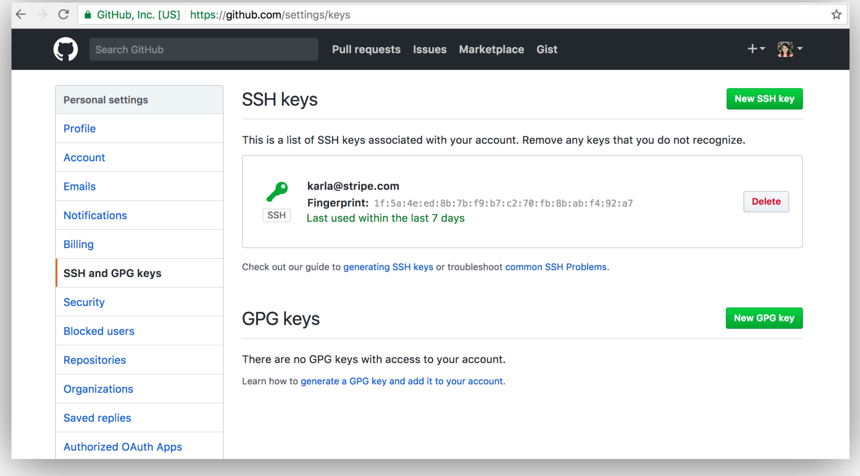
Tip: You are entering sudo mode. We won't ask for your password again for a few hours.

Alternatively, press the button on your security key...

Tip: You are entering sudo mode. We won't ask for your password again for a few hours.

Left: original GitHub "upgrade privilege" page. Right: cloned page, hosted on GitHub Pages.

Finally, after authentication users would be redirected to the legitimate GitHub key settings page. This meant that users who were uneasy about the domain they had authorized would be reassured by the presence of a legitimate eventual redirect.



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

karla@stripe.com
Fingerprint: 1f:5a:4e:ed:8b:7b:f9:b7:c2:70:fb:8b:ab:f4:92:a7
Last used within the last 7 days

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

GPG keys

There are no GPG keys with access to your account.

Learn how to generate a GPG key and add it to your account.

Legitimate GitHub key settings page

This campaign provided a number of interesting data points worth discussing. First, GitHub exclusively sends its transactional account emails in plaintext, presumably catering to their developer-heavy user base. However, typically the conversion rates for “pretty” HTML emails are higher, hence their common use in marketing campaigns.

We decided to test which property mattered more, the similarity to existing emails, or the “prettiness” of the emails. We sent half of our employees a plaintext email, and half an HTML email whose style had been copied from GitHub website, then compared conversion rates on each. We saw a low 10% of users convert from email view to click on the plaintext email, but a much higher 50% of users convert on the HTML email.

We also added tracking analytics to the phishing page we hosted. In particular, we tracked views, keystrokes, and form submissions. This data showed us that of the users who submitted their credentials, nearly half had copied and pasted their password into that field. This suggested that users were using password managers without browser integrations, or that they were ignoring the advice of their password manager, and manually copying their credentials into a suspicious page.

Even more worryingly, we saw very low reporting rates for this campaign, with only a small number of users flagging either the email or the website as suspicious.

THE PSYCHOLOGY OF PHISHING

These results were very worrying to us - not only could our employees not detect a phishing campaign before falling for it, they also failed to detect it when their password manager prompted them that the domain didn’t match. Not only that, but the employees falling for these campaigns were just as likely to be in technical positions as not.

To understand why this is, we first need to understand the way in which the human mind perceives the world. In his book, *Thinking, Fast and Slow*[16], Daniel Kahneman posits that the brain has two ways of operating, which he calls System 1 and System 2.

System 1 operates quickly, below what we’d consider the level of conscious thought, and is heavily pattern driven. It’s the part of your brain that “just knows” what the right decision is, or swerves and brakes when someone ahead of you on the road stops suddenly.

On the other hand, System 2 is much more deliberate. When you weigh up hefty financial decisions, like whether to make a given investment, or make a list of pros and cons, you’re using your System 2 method of thinking.

SYSTEM 1	SYSTEM 2
Fast	Slow
Instinctive	Methodical
Emotional	Rational
Gullible	Skeptical

System 1 plays a very important role in our lives, making unimportant decisions for us quickly, and freeing our minds up to focus on those things that do really matter. But this system of operation has a flaw: it is both emotional and gullible, making it susceptible to being tricked. Unfortunately, thanks to the sheer volume of email each of us receive in a day, we end up reading mail using our System 1 thinking, not our System 2. This means we evaluate the legitimacy of emails based on pattern recognition: does this look more like the type of phishing emails I have seen before, or like a real email I have received from this service. For a well-constructed phishing campaign, the answer will always be the latter - the lack of spelling mistakes, the “prettiness” of the email, and the similarities to other emails sent by the service will fool System 1 into thinking the email is legitimate.

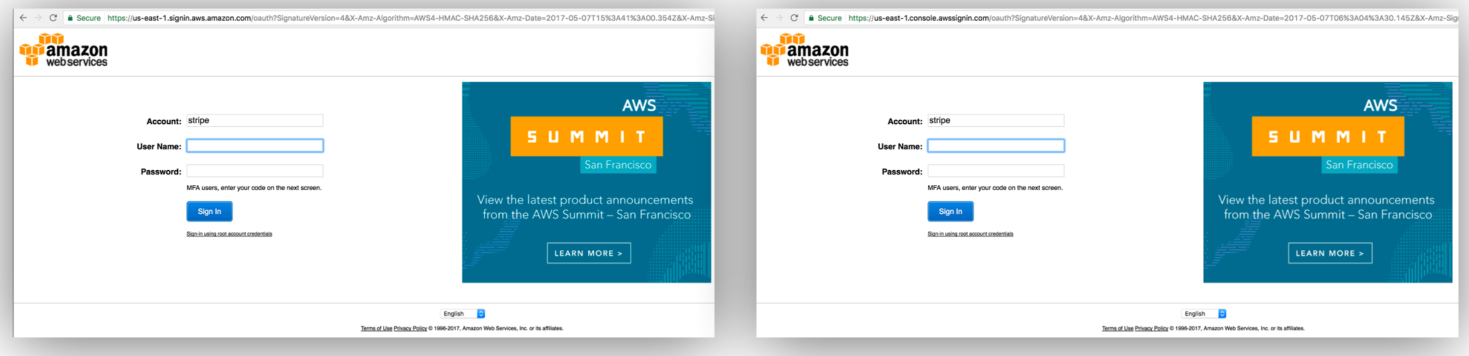
This explains why the technical sophistication of users has little bearing on how likely they are to fall for a well-crafted phishing email: the ability to identify whether or not the headers of an email are legitimate, or whether a domain belongs to the organization you expect, is useful only once you are using your System 2 mind.

CASE STUDY 3: AWS

As a final experiment, we wanted to test how much retention we would see after a session of interactive phishing training. To begin with, we trained the entire company on the phishing campaigns we had run in the past, and how difficult it could be to identify them.

We also presented employees with a series of other fake phishing pages, and had them guess which of them were real and fake. We highlighted that URLs that were complex were more difficult to detect as suspicious than URLs that were shorter.

Three months later, we sent out a phishing email linking to an AWS phishing page, one of the sites we had shown as being difficult to detect as fraudulent during phishing training. This page asked not just for a username and password, but also for a two factor code, before showing a generic success message.



Secure https://us-east-1.signin.aws.amazon.com/oauth?SignatureVersion=4&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=2017-05-07T15%3A41%3A00.354Z&X-Amz-Si

Account: stripe

User Name:

Password:

MFA users, enter your code on the next screen.

Secure https://us-east-1.console.awssignin.com/oauth?SignatureVersion=4&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=2017-05-07T06%3A04%3A30.145Z&X-Amz-Sig

Account: stripe

User Name:

Password:

MFA users, enter your code on the next screen.

amazon web services
Sign In
Sign-in using root account credentials
AWS
SUMMIT
San Francisco
View the latest product announcements from the AWS Summit – San Francisco
LEARN MORE >
English
Terms of Use Privacy Policy © 1996-2017, Amazon Web Services, Inc. or its affiliates.

Left: original AWS IAM login page. Right: cloned login page.

We sent this email only to those employees who might reasonably be expected to have access to AWS. Of those who opened the email, 40% clicked through to the phishing page. Two thirds of those users went on to enter both their password and their two factor credentials into the page. This represented an overall conversion rate from email open to a full set of credentials entered of more than 25%, just 3 months after employees had been trained that this particular page was difficult to identify phishing on. Furthermore, many employees reported after the fact that they had set the email aside to take action on later, and would have fallen for it if we had allowed the experiment to run for longer.

DOING BETTER THAN TRAINING

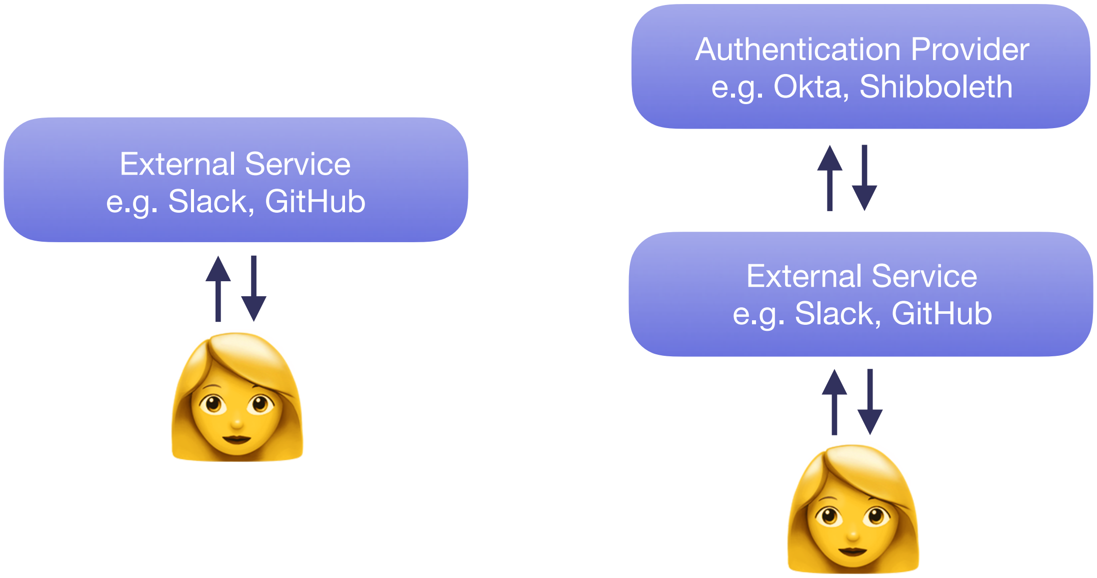
These case studies are damning of two common pieces of industry advice: that both user training and two factor authentication are effective mitigations against phishing. Although user training has proved successful in academic studies[17], it seems ineffective when more sophisticated attacks are used. Similarly, although two factor authentication increases the technical competence required of an attacker, since the attack must now be conducted online rather than offline, it is no real deterrent to someone motivated. Even SMS 2FA credentials can be stolen, by triggering a login request to the legitimate service once a user's username and password have been stolen. This request will trigger an SMS to be sent to the user, which can then be phished in a second step.

At Stripe, rather than focusing on mitigating more basic attacks with phishing training, we decided to invest our time in preventing credential phishing entirely. We did this using a combination of Single Sign On (SSO), SSL client certificates, and Universal Second Factor (U2F), and now believe that it is impossible to phish for Stripe credentials for protected services.

For completeness, we give a brief overview of each technology below, before discussing their use as a combined solution.

SSO

SSO is a technology that allows services to delegate their authentication to another authentication provider. For example, a typical Slack user enters their username and password on a `slack.com` subdomain, and is then authenticated to just that service. When SSO is enabled by an organization's administrator, a user who wants to login to Slack is redirected to another authentication provider. Once that authentication provider has established the user's identity, either through the presence of an existing session, or by triggering their own authentication flow, they return a signed attestation of user identity back to the original service.



Left: typical authentication flow. Right: SSO authentication flow.

SSL CLIENT CERTIFICATES

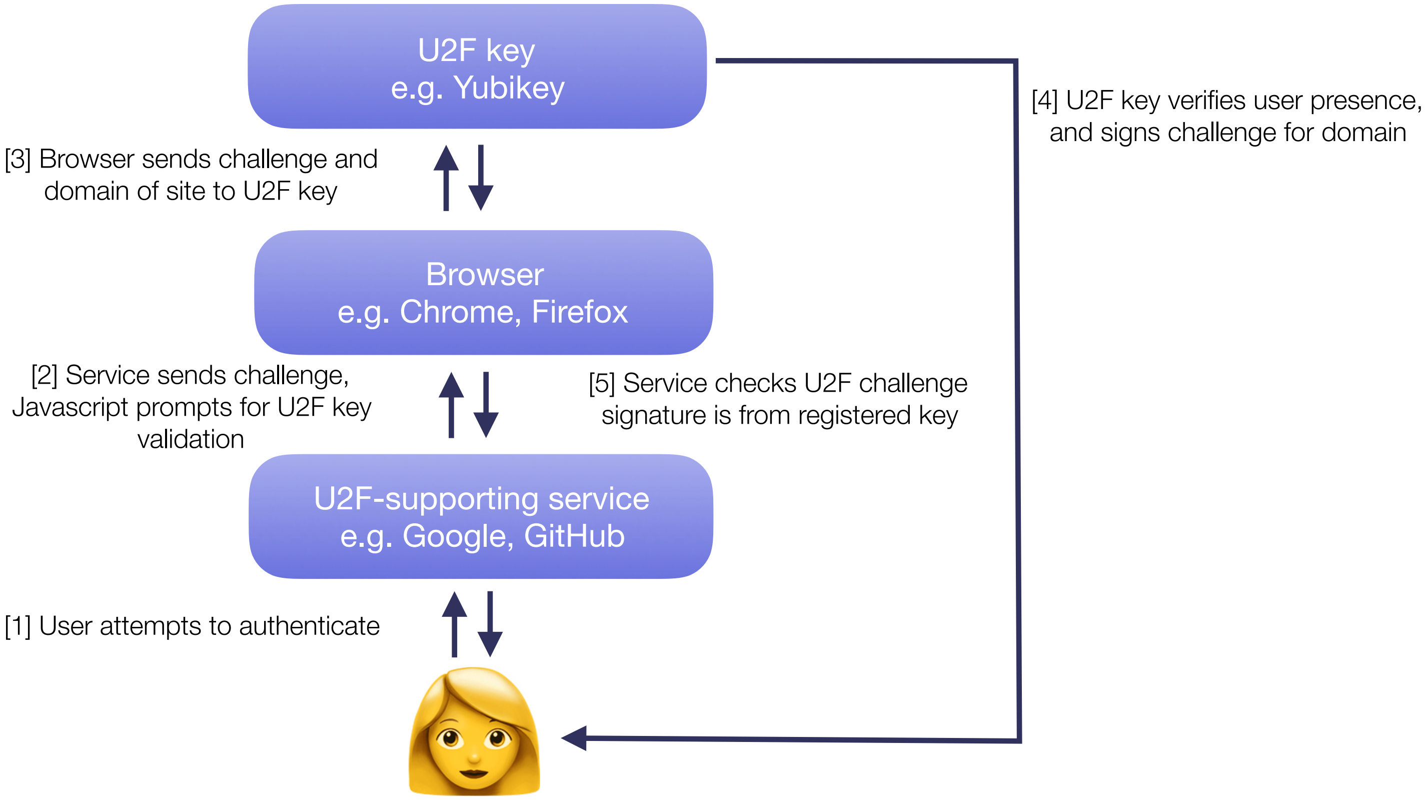
SSL client certificates are the lesser used counterpart to the SSL server certificates we use every day to browse the web. In addition to having the server provide a certificate to the client that can be chained to a trusted root, it is possible for the client to do the same for the server.

This can be configured in most browsers to happen completely transparently, leading to a user experience that is second to none: an employee authenticating with a client certificate to a website is completely unaware that they are doing so. This provides the knock-on effect that users have no password they could be tricked into revealing, short of extracting their client certificate from a trusted store.

U2F

Universal Second Factor is a protocol released relatively recently that provides a unified protocol for physical second factor tokens. Most notably, it allows a user with a single external security key (typically connected over USB or Bluetooth) to use their key on an infinite number of websites, while cryptographically tying each credential to the requesting domain.

Similar to other two factor authentication methods, it provides a single-use token that can be used in addition to another credentials to authenticate a user. However, unlike other 2FA methods, even if a malicious domain convinces a user to authenticate with U2F, that token will be invalid on the legitimate domain. This is because the signatures a U2F device generates for a particular challenge are tied to the requesting domain, as reported by the user's browser.


U2F key
e.g. Yubikey
[4] U2F key verifies user presence, and signs challenge for domain
[3] Browser sends challenge and domain of site to U2F key
Browser
e.g. Chrome, Firefox
[2] Service sends challenge, Javascript prompts for U2F key validation
[5] Service checks U2F challenge signature is from registered key
U2F-supporting service
e.g. Google, GitHub
[1] User attempts to authenticate

U2F authentication flow

PUTTING IT ALL TOGETHER

Combined, these technologies allow an organization to cryptographically tie both long-lived and single-use credentials to a particular website, and in doing so, to completely prevent credential phishing.

For example, consider the first case study, in which Slack was targeted. Instead of configuring Slack to perform its own authentication, an organization would instead setup SSO on Slack, delegating authentication to an SSO provider. That SSO provider would be configured to validate both a client certificate provided by the employee's browser, and to ensure that the user was physically present by requiring validation from a registered U2F key. Even if a user were to attempt to enter their Slack credentials, they would have no clue what password to enter in a password input, and any signature generated by their U2F key would be for the phishing domain, not the real `slack.com`.

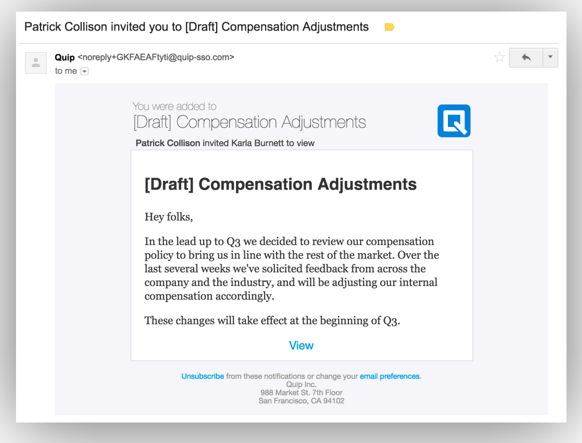
Much of this protection could be afforded solely with the use of a client certificate for authentication, and SSO to allow that type of authentication on a wide variety of sites, so why the need for U2F? One unfortunate downside of client certificates is that they require credentials to be left on an employee's machine, accessible to previously authorized programs without human intervention. The addition of U2F to the authentication flow means that a human must be physically present for credentials to be compromised, increasing the time a malicious actor must have code execution on the machine.

BONUS CASE STUDY: GOOGLE

Although we've found this strategy very effective at preventing credential phishing at Stripe, it does require close attention be paid to edge cases. The loss of login credentials is often not the only way for users to unintentionally give others access to their accounts, and other methods may not delegate to your authentication provider.

For example, Google provides a flow for users to grant access to particular aspects of their accounts using OAuth. This flow was abused in early 2017 to spread a worm that purported to be a shared Google Doc, but really requested full access to your email. Although this particular worm was not harmful, the mechanism it used to spread could easily have been abused by a more malicious actor.

Fortunately for Stripe, we had already conducted this exact campaign internally, albeit a different content hosting provider, who we were using at the time. This meant that we already had defenses in place that prevented our employees from granting sensitive privileges to unknown applications[18].



Patrick Collison invited you to [Draft] Compensation Adjustments

Quip <noreply+GKFAEAFtyti@quip-sso.com>
to me

You were added to
[Draft] Compensation Adjustments
Patrick Collison invited Karla Burnett to view

[Draft] Compensation Adjustments

Hey folks,

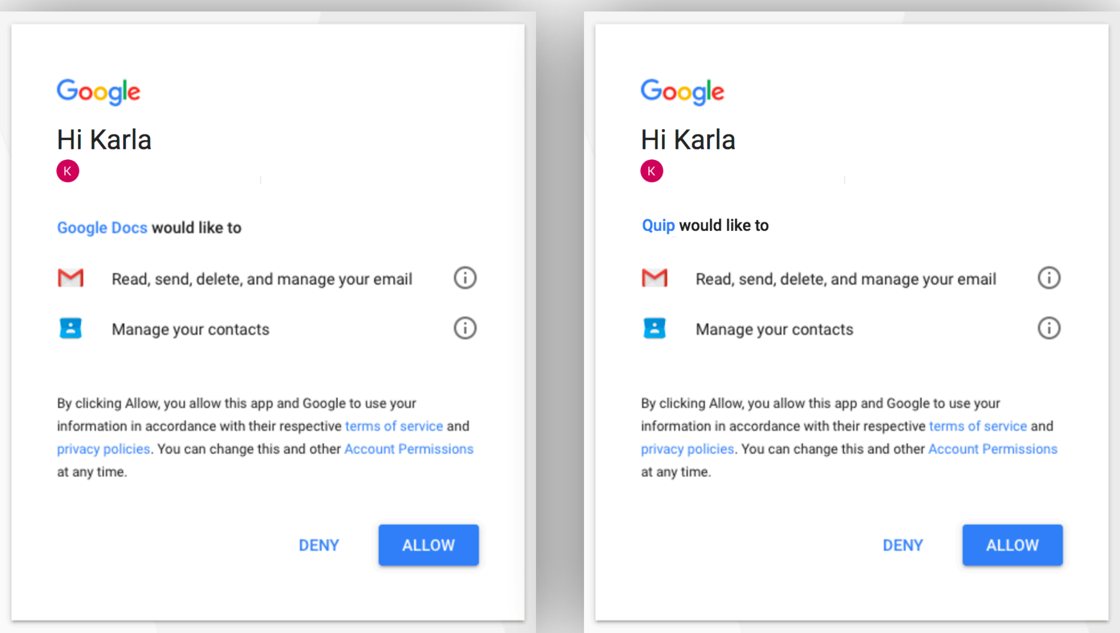
In the lead up to Q3 we decided to review our compensation policy to bring us in line with the rest of the market. Over the last several weeks we've solicited feedback from across the company and the industry, and will be adjusting our internal compensation accordingly.

These changes will take effect at the beginning of Q3.

View

Unsubscribe from these notifications or change your email preferences.
Quip Inc.
988 Market St. 7th Floor
San Francisco, CA 94102

Spoofed CEO email



Google

Hi Karla

K

Google Docs would like to

Read, send, delete, and manage your email

Manage your contacts

By clicking Allow, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other Account Permissions at any time.

Google

Hi Karla

K

Quip would like to

Read, send, delete, and manage your email

Manage your contacts

By clicking Allow, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other Account Permissions at any time.

Left: Google Docs worm OAuth request. Right: Quip phishing campaign request.

This attack underscores the importance of ensuring that you audit all authentication flows available for services you use, to guarantee that users can grant only the expected permissions.

IN CONCLUSION

Although it is easy for attackers to view phishing as something that only the less technically competent will fall for, without careful consideration of the authentication strategies we put in place, we are all vulnerable to a sophisticated attack.

Defenders should not take the easy way out and argue that phishing training is all they can do to protect their users. Red team exercises should include phishing as a valid entry path, and organizations should mitigate or prevent different types of phishing based on the class of attack being used, not the person targeted.

Modern technologies such as U2F, when combined with those that have seen longer use, such as SSO and SSL client certificates, provide effective ways of preventing credential phishing. While credential phishing can be completely prevented, and exploit phishing heavily mitigated by enforcing strict update policies, some training will continue to be necessary for action based phishing attacks. Organizations can reduce the need for this training by limiting the ability for humans to perform one-off dangerous actions, and training that does exist should emphasize that out of band authentication should always be performed for these sensitive actions, regardless of the apparent legitimacy of a given email.

REFERENCES

[1] The Register: RSA explains how attackers breached its systems, April 4, 2011
https://www.theregister.co.uk/2011/04/04/rsa_hack_howdunnit/

[2] Computer World: Sony hackers targeted employees with fake Apple ID emails, April 23, 2015
http://www.computerworld.com/article/2913805/cybercrime-hacking/sony-hackers-targeted-employees-with-fake-apple-id-emails.html

[3] BBC News: Google Docs users hit by phishing scam, May 4, 2017
http://www.bbc.com/news/business-39798022

[4] PC World: ICANN data compromised in spearphishing attack, December 17, 2014
http://www.pcworld.com/article/2860792/icann-data-compromised-in-spearphishing-attack.html

[5] CNBC: Xoom says $30.8 mln transferred fraudulently to overseas accounts, January 6, 2015
http://www.cnbc.com/2015/01/06/xoom-says-308-mln-transferred-fraudulently-to-overseas-accounts.html

[6] CNN Money: Snapchat employee fell for phishing scam, February 29, 2016
http://money.cnn.com/2016/02/29/technology/snapchat-phishing-scam/index.html

[7] Krebs on Security: Seagate Phish Exposes All Employee W-2’s, March 16, 2016
https://krebsonsecurity.com/2016/03/seagate-phish-exposes-all-employee-w-2s/

[8] The Register: RSA explains how attackers breached its systems, April 4, 2011
https://www.theregister.co.uk/2011/04/04/rsa_hack_howdunnit/

[9] SC Magazine: US-CERT warns of phishing campaign spreading Dyre, October 28, 2014
https://www.scmagazine.com/us-cert-warns-of-phishing-campaign-spreading-dyre/article/538336/

[10] SC Magazine: New phishing scam targets FedEx customers, March 22, 2016
https://www.scmagazine.com/new-phishing-scam-targets-fedex-customers/article/529027/

[11] Computer World: Sony hackers targeted employees with fake Apple ID emails, April 23, 2015
http://www.computerworld.com/article/2913805/cybercrime-hacking/sony-hackers-targeted-employees-with-fake-apple-id-emails.html

[12] The Guardian: Top Democrat's emails hacked by Russia after aide made typo, investigation finds, December 14, 2016
https://www.theguardian.com/us-news/2016/dec/14/dnc-hillary-clinton-emails-hacked-russia-aide-typo-investigation-finds

[13] BBC News: Google Docs users hit by phishing scam, May 4, 2017
http://www.bbc.com/news/business-39798022

[14] Sender Policy Framework, an email validation system that allows domains to specify the IP addresses that may send mail on their behalf. http://www.openspf.org/ has more.

[15] Domain Message Authentication, Reporting and Conformance. An email validation system that allows domains to provide rules for how mail that is not correctly DKIM-signed by them should be treated. https://dmarc.org/ has more.

[16] Thinking, Fast and Slow, Daniel Kahneman, Farrar, Straus & Giroux, New York, 2011

[17] Lessons From a Real World Evaluation of Anti-Phishing Training, (Kumaraguru, Sheng, Acquisti, Cranor, Hong), Carnegie Mellon University, 2008.
http://www.cs.cmu.edu/afs/cs/Web/People/ponguru/eCrime_APWG_08.pdf

[18] Google has since released this feature publicly:
https://www.blog.google/products/g-suite/manage-access-third-party-apps-new-g-suite-security-controls/