Tripwire: Inferring Internet Site Compromise

Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker and Alex C. Snoeren
UC San Diego
{jdeblasio,savage,voelker,snoeren}@cs.ucsd.edu

ABSTRACT

Password reuse has been long understood as a problem: credentials stolen from one site may be leveraged to gain access to another site for which they share a password. Indeed, it is broadly understood that attackers exploit this fact and routinely leverage credentials extracted from a site they have breached to access high-value accounts at other sites (e.g., email accounts). However, as a consequence of such acts, this same phenomena of password reuse attacks can be harnessed to indirectly infer site compromises—even those that would otherwise be unknown. In this paper we describe such a measurement technique, in which unique honey accounts are registered with individual third-party websites, and thus access to an email account provides indirect evidence of credentials theft at the corresponding website. We describe a prototype system, called Tripwire, that implements this technique using an automated Web account registration system combined with email account access data from a major email provider. In a pilot study monitoring more than 2,300 sites over a year, we have detected 19 site compromises, including what appears to be a plaintext password compromise at an Alexa top-500 site with more than 45 million active users.

CCS CONCEPTS

• **Security and privacy** → *Intrusion detection systems*; *Authentication*; *Web application security*; *Phishing*; Social network security and privacy; • **Social and professional topics** → **Computer crime**;

KEYWORDS

Password Reuse, Website Compromise, Cybercrime, Webmail

ACM Reference Format:
Joe DeBlasio, Stefan Savage, Geoffrey M. Voelker and Alex C. Snoeren. 2017. Tripwire: Inferring Internet Site Compromise. In *Proceedings of IMC '17, London, United Kingdom, November 1–3, 2017,* 14 pages.
https://doi.org/10.1145/3131365.3131391

1 INTRODUCTION

Virtually all online information services, whether email, social networks or e-commerce platforms, rely on user names and passwords to authenticate their users and limit access to content or capabilities. Thus, attackers seeking to compromise such accounts need only acquire the user's credentials.

While there are a range of vectors by which account credentials can be compromised—including phishing, brute force and malware—perhaps the most pernicious arises from the confluence of data breaches and account reuse. In this scenario, an unrelated site is compromised such that all of its user accounts and passwords (or, more commonly, password hashes) are exposed. An attacker can then leverage this information to access any other accounts a user may have using the same credentials (a situation exacerbated by the widespread use of a user's email address as their standard user name) [3, 8, 14]. In one recent study, Das et al. estimated that over 40% of users reuse passwords [27] and our own anecdotal experience with stolen bulk account data suggests that up to 20% of stolen credentials may share a password with their primary email account.

Moreover, opportunities for such attacks abound, with reports of data breaches now commonplace: in just the past year, reports have surfaced of 117 million account credentials stolen from Linkedin [10] and 360 million from Myspace [4]. In 2014, Hold Security reported obtaining credentials for more than a billion users from breaches on several Internet services [39]. Indeed, the market for stolen credentials is thriving, with credentials being sold in bulk for under a penny a piece [41]. The value of these credentials lies in their ability to be used across sites, enabling account compromise at sites otherwise wholly unaffected by the unrelated original site's compromise.

The most sensitive and important credentials on offer are those associated with major email providers (e.g., Gmail, Live/Hotmail, Yahoo, etc.) because in modern usage it is these accounts that are the foundation for one's Internet footprint. In particular, online services commonly require an email address to register, to reconfirm accounts, to communicate key information and to reset or recover passwords. Thus, access to someone's email account can be sufficient to gain access to a broad array of other services as well. Indeed, while email accounts with such services are routinely compromised en masse, only one major email provider has had a public breach to date (Yahoo, in late 2014 [21]).

Unfortunately, the password reuse problem is not easy to address. While it can be mitigated by the use of password managers and two-factor authentication, these require mass transitions in user behavior that have proved difficult to achieve quickly. Absent that, most providers behave reactively. Once it becomes known that a site is compromised, the site operators will commonly reset the accounts of their users. As well, many large service providers will reset or lock down the accounts of their customers known to have accounts with a compromised site (on the presumption of password reuse).

However, such responses are predicated on knowing of the compromise. In many cases such compromises may never be discovered, let alone become public. Small sites may lack the staff and instrumentation to detect compromises and even large, well-managed sites

IMC '17, November 1–3, 2017, London, United Kingdom

ACM ISBN 978-1-4503-5118-8/17/11…$15.00
https://doi.org/10.1145/3131365.3131391

have no easy way to identify the source of a breach when their accounts are compromised via password reuse. Further, attackers are incentivized to be quiet about successful breaches; if the breach is known publicly, users and service providers will take steps to mitigate the effects, thus devaluing the attacker's cache of credentials. Indeed, the 1.2-billion account credentials recovered by Hold Security are reported to have come from compromises of over 400,000 distinct, relatively unknown sites.

Given this reality, a critical issue is being able to determine when credentials breaches occur, thus opening up other sites to password reuse attacks. In this paper we describe a technique for inferring the occurrence of such breaches (both large and small) without requiring any special access to Internet sites or their hosting infrastructure. Our measurement approach detects site compromises externally by exploiting precisely the attacker's interest in the password reuse vector. In particular, by registering honey accounts at Internet sites using unique email addresses, we place our own accounts at risk, indistinguishable from any other user's account at each site under observation. By further arranging that each of these accounts shares a unique password with its corresponding email account, we create a clear password reuse attack opportunity. If any of these email accounts is ever accessed, such action provides strong and singular evidence of a compromise at the corresponding site. This approach allows a wide array of Internet sites to be efficiently monitored for compromises and admits no false positives—presuming the email provider itself is not compromised.

We have built a prototype system, called Tripwire, to implement this technique, which automatically crawls and registers accounts in this matter. We partnered with a major email provider to conduct a pilot study of this approach covering approximately 2,300 sites. Over a year's time, we discovered evidence of compromise at 19 sites, all but one of which were previously undisclosed to the best of our knowledge. The sites at which we detected breaches range in size from very small to a large publicly-traded company with more than 45 million active customers at the time of compromise. Moreover, by controlling the form of the passwords we can determine whether a compromise is consistent with a dictionary attack on password hashes (and thus users with strong passwords may have been protected) or whether the attacker was able to obtain the passwords in cleartext (and a strong password would not have helped).

In the remainder of this paper, we discuss previous work related to the Tripwire technique, the ethical considerations that guide our study, and our account creation, registration, and monitoring methodology. We then quantify the effectiveness of our pilot monitoring, and report our qualitative experience disclosing our findings to each of the affected sites, including how they receive such evidence and the extra-technical challenges in using it to change behavior. We conclude with a discussion of the complexities of automated account registration and the challenges in scaling such a service further.

2 RELATED WORK

The Tripwire measurement technique fundamentally depends upon attackers stealing email account credentials on one site and then taking advantage of shared password behavior to access the stolen account on the email provider. Researchers have repeatedly found that users reuse their passwords across multiple services [27, 31], and that they have accounts on at least 25 distinct online sites [15, 17, 29], with some estimates putting the number over 100 sites [12]. Given this landscape, it seems likely attackers will continue to exploit this reuse to try to take over additional accounts that might be of greater value [34]. Indeed, there have been several recent reports of attackers taking a large list of usernames and passwords acquired at one service and trying those credentials at another [3, 8, 9].

Previous work has also shown how to detect sites vulnerable to attack [28, 30, 36], defend against those attacks [22, 23], and evaluate the risk of compromise to a site or predict whether a site will be compromised in the future [40, 42]. Yet Canali et al. found that few shared hosting providers or "add-on" security services managed to detect even simple site compromises, despite direct server access [24]. Tripwire offers an advantage over these and other schemes because it can be deployed and operated by a third party not affiliated with the websites or their users. Moreover, it is able to detect the effects of both online (e.g., key logging) and offline (e.g., external database dump) attacks, relying only on the integrity of a major, independent email provider, not of the sites being measured.

The use of a designated decoy device or account to detect attacks against a service is a classic security mechanism [25], and honeypot accounts have been used to observe recent attacker behavior in general [38] and, in particular, have been suggested as a means of detecting compromise on an online service since at least 2008 [32]. A mechanism of honeypotting has also been suggested to detect password bruteforcing [35]. All of these systems rely on some aspect of the underlying service being measured having not been compromised, and we believe Tripwire is the first use of honeypots where no part of the system under measurement needs to be trusted.

3 ETHICAL CONSIDERATIONS

Before detailing our system and methodology, it is important to discuss the ethics and potential for harm associated with our study. First, while we obtained the full consent and cooperation of our partner email provider, we do not seek the consent of the websites that we monitor. It is both impractical and potentially damaging to the scientific validity of our study for us to seek prior consent from websites before registering accounts. In particular, sites might opt-out in a biased way (e.g., those who suspect their security to be flawed might wish to avoid being included), or choose to handle our accounts in a special fashion that would break the link between our account disclosure and site compromise.

Largely because we lack informed consent of the websites under test, we do not undertake our study glibly or without significant deliberation. In our view, there are two distinct issues: one of ethics and potential harm, the other of liability.

It is our belief that the potential direct harm we can cause to a site by attempting to register for a few (at most three) accounts that are rarely, if at all, accessed subsequently is limited to the small amounts of storage and load associated with these actions. Our automated crawler was rate-limited to attempt page loads no faster than every three seconds—and typically much slower than that due to intentional processing delays. Only three sites received more than eight registration attempts from our crawler[1]—with the

[1]Due to crawler debugging, the three most frequently accessed sites were contacted 16, 15, and 9 times.

overwhelming majority of sites receiving two or fewer attempts—a load unlikely to burden even tiny sites.

However, there are also indirect harms which may result to the brand or reputation of such sites if the knowledge of their breaches were to become known. For this reason, we have explicitly obscured the identify of the websites at which we detect compromise. Balanced against these potential harms is the concrete benefit to sites arising from earlier knowledge of a data breach and the benefit to consumers from earlier notification of their credentials being compromised. As detailed below, we attempted to notify the operators of all the sites where we detected compromise.

As we discuss at length later, however, our passive monitoring approach—specifically, one that provides concrete evidence of a compromise but no information regarding the exploit or mechanism employed such as pen testing or similar invasive methodologies would provide—can place notified site operators in a challenging position. Disclosing a compromise or forcing a password reset is, at least for some, perceived as a risky move that could drive users away from a service [6]. Depending on jurisdiction, however, sites may be required to notify users of any known compromise to their service [2]. In cases where a site is unable to independently corroborate a compromise, they are forced to choose between their own investigation and our evidence. Further, without being able to find the source of the compromise, they have no ability to assure users that future compromises will not occur.

On the legal side, we consulted extensively with general counsel and acted with the permission and knowledge of our administration. While we make no attempt to explicitly check the terms of service for each site in our study, it is quite likely that one or more aspects of our methodology are contrary to policies on some sites (e.g., sites frequently disallow "automated registration"). Even if not explicitly disallowed, bot activity is plainly discouraged by many sites through the use of CAPTCHAs and other Turing-test-like aspects of their registration processes, which we intentionally try to overcome. Moreover, if sites asked for personal information as part of the registration process, we provided fictitious details. Nevertheless, counsel advised us that the legal risk was low and outweighed by the scientific merits of the work and, moreover, that both the absence of real damages to any party and the limitations on the enforceability of terms-of-service contracts minimize even these limited risks.

Finally, we note that there are no human subject concerns in this study: all of the information we are providing is fictitious, and no human (other than perhaps an attacker who compromised a website under study) ever interacts with the email addresses or the accounts.[2]

4 METHODOLOGY

Conceptually, Tripwire consists of two distinct phases: account registration and monitoring. We designed an automated web crawler to register for accounts, and then partnered with a major email provider to monitor activity at the associated email accounts. Here we describe how we created and populated the email accounts used by Tripwire, the way in which we interact with the email provider, and the operation of our web crawling infrastructure.

[2]Aside from a handful of phone calls to the numbers associated with our accounts (see Section 5.2.2).

4.1 Account and identity management

Tripwire ensures that each account maintains a one-to-one mapping to an identity. These identities minimally consist of an email address and password, though many sites require additional information.

4.1.1 Identities. Tripwire identities must not easily be distinguishable from organically created accounts so that attackers cannot selectively avoid them. Hence, we created a database of fictitious identities and associate each with an email account and password at our email provider that were designed to look as organic as possible. Tripwire identities have full names, addresses, phone numbers, dates of birth, employers, etc. We generate names from sets of real names, and addresses are syntactically and semantically valid (although not necessarily extant) US street addresses [7]. Identities have real US phone numbers under our control. No site saw the same phone number more than once.

We generate usernames and email addresses to look plausible, yet be very unlikely to be taken. We generate the local-part of email addresses in the form of an adjective, a noun, and a four-digit number (e.g., `ArguableGem8317`), and then use the first 14 characters as the username at sites that require a username distinct from the email address. (Experience shows that many sites limit the username length.) Since the email provider does not create email accounts for us when there already exists an account with that username, we use this check as a heuristic for probable availability of a given username on all other services. This allows us to reduce the complexity of the crawler (by allowing us to assume that the username is available).

4.1.2 Passwords. We created accounts with two types of passwords to distinguish the types of compromises that may occur. "Hard" passwords are random alpha-numeric, mixed-case, ten-character strings without special characters (e.g., `i5Nss87yf`). "Easy" passwords are eight-character strings combining a single, seven-character dictionary word with its first letter capitalized, followed by a single digit (e.g., `Website1`). Easy passwords are deliberately easy to crack in a brute-force fashion, while hard passwords are designed to be as difficult to brute-force as common password policy constraints would allow.

Nearly every website we crawled permits eight-character passwords, and many require at least eight characters. The hard password ten-character length is a balance between a desire for long, complicated passwords, and the need to support websites with short maximum password lengths. Passwords do not contain non-alphanumeric characters as, in our experience, few websites require special characters in the password, while several do not support them. These assumptions simplify our crawler by not having to consider password policy when trying to create an account.

Tripwire typically registers for multiple accounts on a site by first attempting to register an account with a hard password. If Tripwire believes that registration succeeded, it enqueues up to two additional registration attempts with differing password types. If we later detect compromises for a site only on accounts with easy passwords (or where those accounts are compromised much earlier than ones with hard passwords), it would suggest that, while the website's database was breached, the website's passwords were well hashed. If Tripwire also detects activity on accounts with hard passwords, then it

suggests that the site was either storing passwords in plaintext, using a weak hash, or the compromise was able to bypass the hashing step.

4.2 Interaction with the email provider

We approached our email provider with the idea of Tripwire because Tripwire works best with a sufficiently attractive target email account. The email provider was only involved in providing accounts to our system, and was not aware of what accounts were used on what services.

We provided a list of identities in advance to our email provider, who then created the corresponding accounts unless they collided with a pre-existing account or violated the provider's naming policies. All email accounts were created with their corresponding name (in case an attacker sought to validate the authenticity of the accounts by checking the personal information at the email provider against the website's records) and forwarded any mail received to our own mail server, where we stored and parsed incoming messages for registration information. Since forwarding addresses are visible in the web interface of our email provider, we used forwarding addresses of accounts at one of a small number of domain names under our control who had their mail hosted by a third-party mail provider. This provider then forwarded messages to our mail server.

In addition to forwarding messages, the email provider notifies us of any successful logins in these otherwise-unused accounts. In particular, we receive sporadic dumps of login information for all of the identities we created, independent of whether they have been used to register accounts at any websites. Our provider is unaware of which accounts have been used, and which remain unassociated. The provider dumps provide timestamp, remote IP, and method (IMAP, POP, etc.) for any successful logins, but does not disclose failed attempts. We also maintain a set of control accounts that are not associated with a website, but into which we log in at our email provider from time to time. All such control login events have been accurately reported by our provider.

4.3 Crawler

To scale account registration, we developed a custom-built web crawler to automatically visit a given site and register for an account. The crawler attempts registrations on a 'best-effort' basis: the crawler explicitly does not attempt to support all of the site registration mechanisms encountered on the Web, as our experiment is designed only as a proof of concept and does not require complete coverage.

4.3.1 Registration. The crawler uses PhantomJS [33], a scriptable, headless web browser based on the WebKit engine [20]. It processes pages according to the flow shown in Figure 1. It attempts to identify a registration page on the site, and if successful, identify and fill each form field serially. If any stage fails, the crawler aborts with a corresponding error code. The crawler does not support any site whose registration system does not follow this basic flow, nor sites that use external account services such as those provided by Google or Facebook. The crawler relies on many hand-crafted heuristics to locate registration forms, fill them out, and submit them. These heuristics take the form of a series of weighted regular expressions and sets of DOM elements to which they apply. Our current heuristics are only designed to support sites written in English.

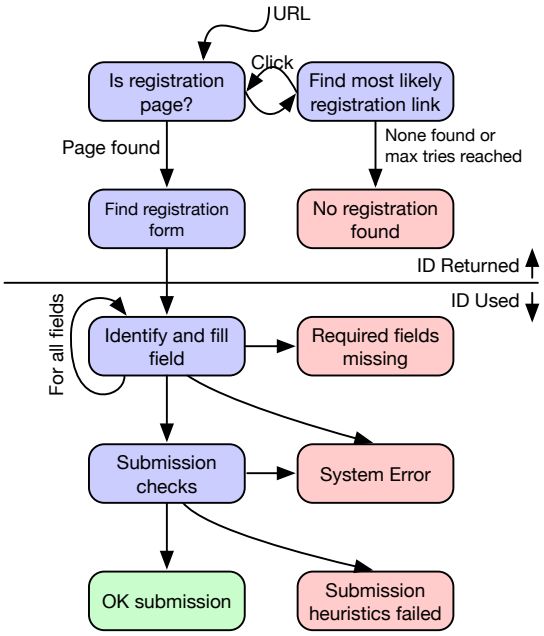

Figure 1: Control flow of the Tripwire crawler. Given a URL, the crawler returns a code indicating the reason for termination.

If an email address or password was ever shown to a site—regardless of Tripwire's assessment of the success of this submission—we "burn" the identity and forever associate it with that website. If a registration attempt fails, the identity used may be returned to the general pool to be used on another attempt only if neither the email address nor password were exposed to the website. The horizontal line in Figure 1 depicts the approximate point at which an identity typically becomes permanently associated with a site.

4.3.2 Bot-detection avoidance. The crawler bypasses some rudimentary bot-checking systems (such as CAPTCHAs or basic human-knowledge questions) by relying on a third-party CAPTCHA-solving service [5]. The crawler operates using a small network of web proxies that our group maintains solely for research purposes. These IPs are not meant to be unattributable—WHOIS records clearly state our institution name. They serve simply to decouple multiple registration attempts at the same site: websites receive at most one account registration from a given IP. We made no attempt to match the geolocation of the proxy IP to the address for a given identity, but in practice this did not seem to prevent registration.

4.3.3 Mail handling. Since many websites require confirmation of email addresses to create an account, our email provider forwards any messages delivered to our accounts to a mail server under our control. This server retains a copy of all messages received, and, as needed, processes incoming message contents. In particular, it processes all incoming messages to evaluate whether a message is associated with a recently-registered account, and, if so, if the message contains a validation link. If it does, the mail server loads the verification page and saves it for future debugging.

4.4 Interpreting account compromise

The key assumption with Tripwire is that a successful login reported by the email provider is the result of an attacker having stolen credentials from the site on which we registered with our email account. We end our methodology by arguing why we believe this assumption to be valid.

Acquiring the correct credentials requires collecting them either on a machine storing them, or in transit between credential holders. Credentials are stored in three places: within our own database, with the email provider, and with the site under measurement. We have striven to minimize the risk of leaking data from any of these sources. Our database is accessible only from a small number of servers on a small internal network, none of which provide externally accessible services. Communication between our servers is tunneled over SSH. Individual instances of the crawler have only the identity assigned to one site, so compromise of multiple identities would require full arbitrary code execution on the crawling machine.

The provider treats the email accounts used in Tripwire equivalently to their hundreds of millions of other accounts. The email provider has mechanisms to detect attempts to brute-force passwords. No known breaches of the email provider affected accounts used in Tripwire, and sensitive account credentials were only exchanged between the authors and the email provider via verified PGP.

Perhaps the most compelling evidence of the integrity of the Tripwire accounts is from the fact that no accounts were tripped that were not associated with Tripwire registrations. Tripwire has a database including more than 100,000 valid email addresses and passwords obtained from the provider that were monitored for logins, but were not registered with sites. The unused accounts conveniently serve as honeypot accounts to detect any compromise of the email provider or our own database since they are stored with the accounts used in the study, but have not been used for registrations. None of these unused accounts have ever been accessed.

For the sites under measurement, a possibility is that an attacker brute-forced our credentials without explicitly breaching a site, e.g., an attacker somehow guesses our usernames (or a site exposes them) and the site does not prevent brute-forcing attempts on its accounts. If so, then an attacker could conceivably have found the Tripwire username, brute-forced the password with the site, and then used those credentials in a password-reuse attack on the email provider. While unlikely, we consider this within the bounds of attacks that Tripwire should detect, and Tripwire would correctly declare a site as compromised in this situation.

In communicating with sites under measurement, the system used HTTPS when preferred by the site, validating certificates with a commonly accepted list of roots, and many of the tripped sites used HTTPS during the registration process. It is possible that an attacker may have actively impersonated a site during Tripwire's registration process. But we consider this threat to be an unlikely one, with only a few attacks of this kind having been seen in the wild, primarily due to targeted attacks by state-sponsored actors [16].

Finally, it is possible that a Tripwire account is stored in a sharded database on the site, and only a subset of the shards are compromised in an attack. If a Tripwire account is in an exposed shard, Tripwire indicates that a database breach occurred and still detected a significant compromise of the website under measurement. Conversely, if a Tripwire account is not in the shards exposed, then Tripwire will miss any attacks on the affected users (similar to a breach that did not result in password-reuse attacks). Registering for many additional accounts could reduce the possibility of being stored in an unbreached shard, but we consider this possibility to be remote, and additional registrations introduces ethical challenges that are not outweighed by the benefit to this rare case, especially given Tripwire's otherwise negligible false-positive rate.

5 ACCOUNT CREATION

We used the Tripwire crawler to register for accounts in batches between July 2014 and July 2016, with most occurring between January and March 2015. Tripwire made 65,413 distinct registration attempts across 33,634 different sites, using a total of 8,352 identities. We detail our validation methodology below. In our best estimate, Tripwire successfully registered for approximately 3,664 accounts on around 2,302 sites.

5.1 Website selection

We registered accounts primarily on four occasions from December 2014 through May of 2016. We initially seeded our crawler with the Alexa top-1,000 sites [1] combined with the Quantcast top-1,000 sites [18] (with duplicates removed) in December of 2014. Subsequent registrations occurred from January through March of 2015 covering the Alexa top-25,000 sites. In late November 2015, we attempted registrations on all sites in the Alexa top-30,000. Finally, in May 2016 we manually registered for accounts at all of the eligible Alexa top-500 sites to ensure good coverage of the most popular sites. In each case, we used the most up-to-date rankings available at the start of the registration window.

In all of the automated cases, we filtered URLs through a set of regular expressions to remove sites known to use common backends—e.g., `Amazon.com`, `Amazon.de`, etc., or Google, YouTube, Blogger, Blogspot, etc.—and others.

5.2 Registration attempts

Because our infrastructure has no automated way to validate registrations after it attempts to create them, there is uncertainty in the number of accounts and sites for which the crawler successfully registered. Hence, we rely on heuristics during the registration process, email-based indicators, and manual sampling to estimate success.

5.2.1 Crawler termination conditions. Figure 1 presents the termination conditions for the Tripwire crawler's execution across various sites. "Required fields missing" indicates that the registration form did not meet the conditions for a valid form (e.g., did not ask for both password and email), or the crawler was unable to recognize a sufficient number of fields to attempt registration. "Submission heuristics failed" corresponds to the case where the crawler submitted a registration, but suspects that it did not succeed, while "OK submission" indicates its heuristics suggest it did. Finally, "System Error" represents cases where the crawler was otherwise unable to process the site. We investigate the outcomes of the crawler in Section 7, though we note here that crawler outcome distributions were similar across Alexa ranks.

	Attempted					Estimated Valid			
Account Status	Hard	Easy	Total	Sites	Success	Hard	Easy	Total	Sites
Email verified	1,552	508	2,060	1,359	98%	1521	498	2,019 (55%)	1,332
Email received	128	51	179	106	82%	105	42	147 (4%)	87
OK submission	1,069	703	1,772	860	59%	631	414	1,045 (29%)	507
Bad heuristics/Fields missing	4,395	122	4,518	3,420	7%	308	9	317 (9%)	239
Manual	0	137	137	137	100%	0	137	137 (4%)	137
Total	7,144	1,521	8,666	5,882		2,565	1,100	3,665 (100%)	2,302

Table 1: Estimates of accounts created by account status.

5.2.2 Out-of-band confirmation. In addition to the heuristics Tripwire uses at registration time, some sites provide further confirmation of registration via email. If an email account receives an account verification message, we label it "Email verified". If the account receives email, but we do not recognize it as a verification message, we label the account "Email received". Over 47% of "OK submission" results at registration time triggered a verification message, and 4% more triggered at least some kind of email message. Fewer than 8% of the registration attempts of the other categories resulted in any kind of email message.

While it is possible that registrations could also be verified by phone, no phone verification occurred in our sample. We did receive 18 calls from seven distinct self-identifying sources ("Hi, this is John from site X") to our phone numbers that were directly attributable to the accounts we registered. (We received several additional phone calls, but they seemed to be wrong numbers or call-center scams that cannot be conclusively tied to the phone numbers used in Tripwire accounts.) All attributable calls were sales teams following up on what appear to be free-trial accounts for which Tripwire registered.

5.2.3 Success estimation. After accounting for email reception, we have five distinct outcome categories shown in Table 1. We manually tested a random sample from each category to determine their expected success rates as a basis for estimating the number of accounts and sites for which registration succeeded.

The "Attempted" columns on the left of Table 1 show the number of accounts and sites according to each registration attempt. "Hard" and "Easy" correspond to accounts created with those respective password strengths, and "Total" is their sum. "Sites" corresponds to the total number of sites on which we registered accounts (some sites had multiple accounts). We estimate the success rate in each category by selecting 50 random accounts and manually attempting to log in to the corresponding site. The "Success" column shows the success rate of these login attempts. The "Estimated Valid" columns on the right then show our estimates of the true success rates by discounting the "Attempted" columns by the login success rates.

Email verified. Our highest confidence bin for automated registrations is any account that received a recognized verification email. This category consists of 2,060 automated registrations. In our manual tests of a sample of accounts, they succeeded in 98% of cases, resulting in an estimated 2,019 accounts across 1,332 domains.[3]

Email received. An additional 179 registration attempts received email, but the message did not appear to require email verification. These accounts were valid in 82% of our tests, for an additional 147 accounts on 87 domains.

OK submission. In 1,772 registration attempts, our attempt passed all heuristics for success, but no email was received. In our sampling, 59% of these accounts exist, accounting for 1,045 more accounts on 507 domains.

Bad heuristics/Fields missing. The lowest-probability-of-success outcome is that the system exposed a username and/or password, but the system triggered a heuristic signaling failure or did not attempt to submit the form. In these cases, approximately 7% of attempts still succeed, for an additional 317 accounts on 239 domains.

Manual. Finally, we manually registered accounts at the 137 English-language sites accepting registrations among the Alexa top-500 sites (4% of all accounts registered).

[3]In the one "failure" case, the site in question is an app-development site partially hosted at GitHub with a local account registration. Tripwire successfully signed up for an account on GitHub instead of the site in question.

6 COMPROMISES DETECTED

At various points during our study, our email provider reported any successful login activity for Tripwire email accounts. (For non-technical reasons, we were unable to collect login information on a periodic or real-time basis.) As discussed in Section 4.4, we interpret a successful account login as indicating a compromise of the associated site. Among the estimated 2,302 sites with successful account registrations, Tripwire detected 19 such site compromises between June 2015 and February 1, 2017.

Figure 2 shows the login activity to email accounts stolen from the compromised sites across time. Each row corresponds to a compromised site, vertical ticks show when we registered for accounts on the site, squares show logins to email accounts with easy passwords, and triangles show logins to email accounts with hard passwords. They are sorted from the top according to time of first account login. Numbers to the right of each row indicate the total number of account logins for that site. The shaded region in Spring 2015 corresponds to a gap in our account login data. Due to a misunderstanding of the retention limits at the email provider, login activity was lost from March 20, 2015, through June 1, 2015. Although no logins were detected for more than a month after collection resumed, it is possible that additional sites were compromised and would have tripped our system during that time.

In the rest of this section, we characterize the sites that were compromised and detected by Tripwire, as well as other compromised sites during the same time frame. We then describe our results of disclosing the compromises to the sites. Finally, we summarize the activity of attackers who accessed the stolen email accounts.

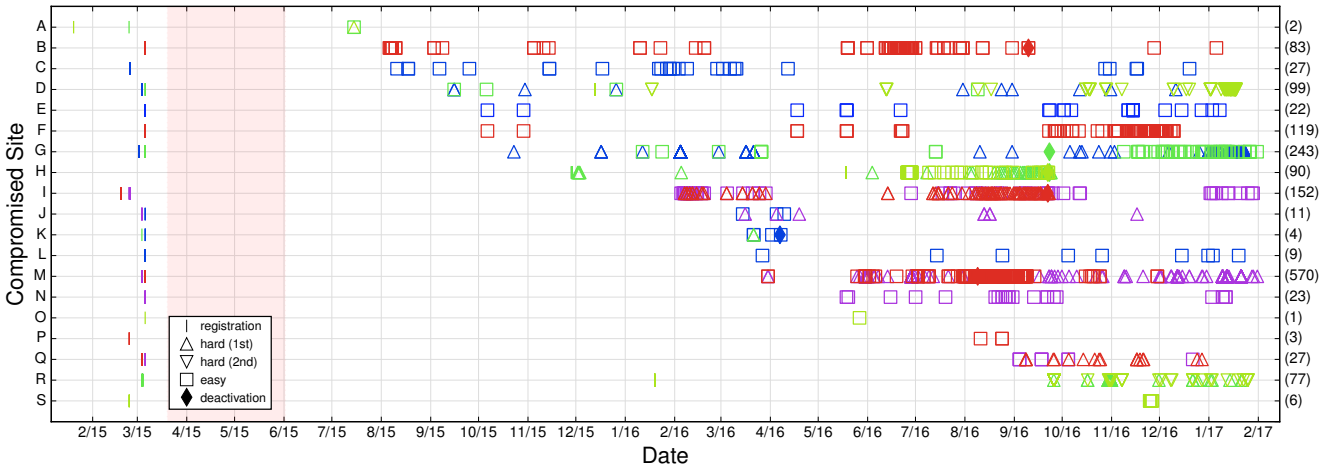

Figure 2: Registration and login activity to email accounts stolen from compromised sites. Each row corresponds to a compromised site, and different colors on the same row indicate activity on different accounts at that site. The numbers along the right y-axis indicate the total number of logins for that site across all accounts. The shaded region in Spring 2015 corresponds to a gap in our logs.

Site	Accounts accessed	Hard accessed	Category	Alexa rank
A	2 of 2	Y	Deals	500
B	1 of 2	N	Gaming	8500
C	1 of 2	N	BitTorrent	5500
D	3 of 3	Y	Wallpapers	20500
E	1 of 2	N	Gaming	16000
F	1 of 2	N	Gaming	18500
G	2 of 2	Y	RSS Feeds	17500
H	2 of 2	Y	Marketing	17500
I	2 of 2	Y	Horoscopes	7500
J	2 of 2	Y	Gaming	20500
K	2 of 2	Y	Classifieds	20500
L	1 of 3	N	Adult	11000
M	2 of 2	Y	Vacations	20000
N	1 of 2	N	Gaming	11500
O	1 of 2	N	Outdoors	18000
P	1 of 1	–	Adult	1500
Q	2 of 2	Y	Tourism Guide	22000
R	2 of 2	Y	Press Releases	22500
S	1 of 2	N	BTC Forum	4000

Table 2: Summary of sites with detected login activity. Rank at registration time rounded up to nearest 500. No 'hard' account was registered at site P.

6.1 Sites compromised

For each site, Table 2 shows the approximate Alexa rank, site category, the number of accounts created and accessed, and whether an account with a hard password was accessed. We explore what site characteristics appear to correlate with their compromise, how sites manage their account databases, and which compromises we detected were also disclosed by the sites themselves.

Overall, we find that while most of detected compromises are at small sites with few staff, Tripwire has also detected compromises on large sites as well. Tripwire detected both plaintext and hashed-password breaches, and has predominantly discovered breaches that have previously been undisclosed.

6.1.1 Site characteristics. The compromised sites cover a wide range in terms of popularity. The detected compromised sites are distributed throughout our covered site ranking, from a top-500 site through the full range of sites selected.

The most popular site compromised is a well-known American startup with more than 45 million active customers as of the quarter they were compromised (Site A). Sites E and F, owned by the same parent company, are a large gaming-services company well known within online gaming communities. We also detected compromise on a top-500 site in India, the top-ranked site in its category (Site I) which claims millions of installs of their app and more than 60 million visits to their site per month. Site P, a 'tube'-style pornography site, is a top-400 site in Germany. Site Q is owned by a company with a large portfolio of travel recommendation websites, claiming 40 million views across all sites every month.[4] Finally, site S, bitcointalk.org, is a prominent Bitcoin discussion forum that experienced a publicly acknowledged database breach in May of 2015. Contents of that breach were reportedly sold online in 2016. While the distribution looks somewhat skewed towards lower-ranked sites, there are too few sites to observe the distribution definitively.

The compromised sites comprise a variety of site categories, although gaming (i.e., sports or video games) is the most prevalent. These sites are fairly representative of sites with large user bases towards the tail of popularity.

Except sites A, E/F, Q and R, the remaining sites appear to be run by individuals or small teams. A few of the sites have not been meaningfully updated in several years, and site C has since disabled account registration. Most of the sites appear to have been created with good intentions for their stated purpose. Three sites (G, K, M), though not malicious per se, appear to have been created explicitly to generate ad revenue and offer services with little actual value.

6.1.2 Password management. The password strengths of the accounts provide insight into the password management practices of

[4]We did not register at any other sites owned by this company, so cannot speculate whether the compromise is limited to that site or spans across their properties.

the sites. For sites that only store hashed passwords, easy passwords can be guessed using dictionary attacks while hard passwords remain protected. For sites that store passwords in plain text, both easy and hard passwords are vulnerable.

In eight cases (sites B, C, E, F, L, N, O, S), our system registered for both an 'easy' and a 'hard' account at a site, but logins only occurred on the 'easy' accounts. This behavior suggests that these sites hash passwords sufficiently to at least delay the compromise of accounts with stronger passwords, or are leaking account credentials due to large-scale brute-forcing. Despite well-known security practices, we observed logins using 'hard' passwords on ten sites (including site A). These sites appear to have stored account passwords in the clear or used easily-reversed hashes. (For site P, we only successfully registered an account with an easy password.)

Our methodology only registered for accounts with easy passwords after it estimated that a hard registration succeeded. This biases our results to under-report compromises, as 'easy' passwords are more frequently compromised. Subsequent invocations of a Tripwire system should avoid this pitfall.

6.1.3 Breach indicators. Of the 19 sites that we detected as being compromised, we found only three with external indications of compromise.

As mentioned above, site A is a well-known, popular American startup. Around the time of our observed logins, several of their users on Twitter complained of their accounts on site A being compromised. One publication ran a story discussing the claims, but the site denied the allegations. We can find no further reporting on the issue, but our account logins on site A corroborate these reports, show evidence of attackers using stolen account information for password reuse attacks, and serves as an example of our system providing ground-truth evidence. Section 6.3 details our discussions with the site when we disclosed our account compromises; from their investigations, they reported finding no internal evidence of a site compromise but could provide no explanation for our results.

We also found a post on an unrelated forum claiming to provide a link to the user database of site L, a pornographic 'tube'-style site. We were unable to verify the availability of the database or the validity of the claim, but the posting time is consistent with the login attempts we see on account l1.

Finally, we detected an account compromise on site S, bitcoin talk.org. This site was known to be compromised as of May 2015, and there were subsequent reports of the database of hashed passwords being for sale on underground markets in 2016. Our detections are consistent with this timeframe, and consistent with the leak of hashed passwords [26].

We could find no evidence of disclosure of any of the other compromises. We provided the usernames we used on sites A–O to several major threat intelligence companies and online service providers in possession of large collections of compromised accounts, and none of the companies found any evidence of breaches.[5] We also searched a variety of public and private sources of compromised database dumps for evidence of our breaches without success.

6.1.4 Recovery from compromise. Although we found only three external indicators of account breaches, most of the sites appear to have either only been compromised at a single point in time, or were able to recover from the breaches. We registered for additional accounts on all sites except C (which no longer accepts registrations) and O–R (whose compromises had not yet been detected) as of mid-May 2016. To date, only our additional account at site H has been accessed and none others.

[5]These companies requested to not be named in exchange for their assistance.

6.2 Undetected compromises

It is natural with a system like Tripwire to want to calculate the proportion of compromises that Tripwire is able to detect (i.e. recall). Unfortunately, such a calculation is not possible in practice, as it is not possible to generate an accurate number of total compromises that have occurred in the open Internet. Further, Tripwire does not attempt to detect all compromises—it merely aims to expose compromises that otherwise would have gone undetected. It is still valuable, however, to understand why our implementation fails to detect an otherwise known breach. In this section we explore why Tripwire did not detect 50 recent data breaches as listed on a site that curates public data breaches [11] and, for each, examined why Tripwire did not detect a compromise.

6.2.1 Missed due to scale/scope. In 22 of 50 cases, the sites involved with the compromise were ranked too low according to Alexa to make our test corpus, despite many sites being quite large. Choosing sites to target is a non-trivial problem. While Alexa provides high-quality rankings for the most popular sites, our experience has shown it to be less reliable for the long tail of the distribution.

In seven instances, the sites involved with the compromise were not in English (six were Chinese-language sites and one was Russian). In both cases, a larger-scale and language-agnostic deployment of Tripwire may have been able to register for accounts without issue.

6.2.2 Missed due to technical challenge. In 14 of the 50 cases, the sites were missed due to limitations of the Tripwire prototype. In five cases, the crawler failed because it was not designed to handle multi-page registration forms. In four additional cases, the crawler failed a bot-detection check (e.g., a CAPTCHA image or a free-form common-knowledge question). In one case we successfully registered, but failed to properly verify the accounts by clicking on links sent to the email address.

In four cases, the crawler was unable to locate the registration page either due to it not being clearly accessible from the home page of the site, or because the registration page was not obvious based on the text of the page (e.g., because they relied on text embedded in images). These cases represent significant technical challenges which any implementation is likely to struggle with; however, they are not fundamentally at odds with the Tripwire approach either. For this limitation, it may be possible to rely on search engines to help locate the registration pages.

6.2.3 Missed due to inherent limitations. In six cases, Tripwire could not have registered for accounts either because the site required payment (two cases), or because the site did not support online registrations (four cases) because accounts are established by external means (e.g., being a customer of a bank). In one case, the site limited the length of the email address to fewer than 16 characters but the address we tried was 18 characters. These sites

are largely out of scope for Tripwire. While technically possible to automate payment, it is quite difficult to scale. Sites that have login systems but that do not support purely-online registration (e.g., many banks) are similarly out of scope.

6.3 Disclosure

Given the significance of account breaches, we contacted all sites from which attackers had gained access to our accounts (except for one case where the breach became publicly known). We disclosed our identities, methodology, and findings, and engaged with each site to the extent that they were willing. Although each site had its own unique situation, we can summarize our findings as follows:

- Six of eighteen sites responded to our disclosures.
- Sites that did respond typically responded quickly and took the disclosures seriously.
- Only one site directly corroborated a breach, and the compromise was previously known to them. Some sites acknowledged that security was not their highest priority. No sites that disputed our claim were able to explain how our accounts could otherwise have been compromised.
- No sites have notified users to date (although one site said they would force a password reset).

6.3.1 Disclosure Methodology. We disclosed to sites in two batches, with most occurring on September 7th, 2016, and sites compromised after that date on November 4th, 2016.

The first message we sent identified ourselves as researchers, explained that we believed that usernames and passwords at the site had been compromised, that we were willing to discuss details with the appropriate party, and asked if such a person could respond. We chose this phrasing both to provide confidence and to encourage a response. If we received a response from the site, we sent a second message explaining our methodology and some details of the specific compromise. Subsequent messages, and phone calls if requested, were answered as needed.

We chose recipient email addresses by looking for contact information on the site, emailing the registrants listed in WHOIS data, and emailing common email addresses that might be relevant (e.g. 'security@example.com', 'webmaster@example.com'). No site provided an obvious direct method for contacting the appropriate security contact. In each case, we emailed the complete set of addresses in case any individual address was invalid. We sent messages from the first author's institutional email address, with other authors' institutional addresses CC'd.

6.3.2 Sites without responses. Twelve sites — B, D, H, I, J, K, M, N, P, Q and R — did not respond to messages. Though we found additional contact information for a number of the sites, these messages also did not receive a response. A message to site I resulted in an automatic creation of an account at their internal ticketing system, but no response was ever generated. Site J had no MX record. Site M's email was forwarded to another domain that had expired and was purchased by a domain squatter. For site M, we also sent email to an email-to-SMS gateway address used as a contact in their WHOIS records.

6.3.3 Site A (Deals, Alexa rank 500). The head of security at site A responded within 10 minutes of our initial notification asking for details. Per their preference, most subsequent communications with them were either PGP encrypted or by phone.

Site A asked if we were willing to sign a mutual NDA, which we declined. Per their request, the authors met over the phone with the head of security of site A, an additional engineer, and a member of their in-house counsel. During this call, their head of security asked questions to vet the process of detection and our methodology.

Site A understandably lamented the significant delay between initial compromise and notification—an artifact of our specific measurement implementation, and worse for site A than for other sites. The operators of site A reported that, after our initial disclosure, they employed a third-party incident response team to investigate. Despite both internal and third-party efforts, they were unable to find internal evidence of the breach, but did not have an alternative explanation for how our accounts were compromised. Site A did acknowledge that they were aware of the article we made reference to in Section 6.1.3.

6.3.4 Site C (BitTorrent, Alexa rank 5500). Six days after notification, we received a request for more information, and the subsequent conversation provided details from the operator of site C. The operator explained that an attacker had managed to compromise the site sufficiently to create a competing clone in 2016. Our notification was the first indication to the site owner that the vulnerability had been used prior to 2016.

Site C's owner explained that until this year, passwords were simply hashed with MD5. When asked about whether they would be disclosing the attack to users, they indicated that there was no need, given that 'this information has already become public sine the hacker started a sote fork some months ago' [sic]. When asked about any technical countermeasures, the owner responded with 'sorry cannot tell. however be assured user are protected well'.

6.3.5 Sites E and F (Gaming, Alexa ranks 16000, 18500). Within 30 minutes of the initial message to the owner of sites E and F, the primary author received a voicemail from the in-house counsel of the company attempting to verify our identities. Our initial notification message did not include telephone contact information for the author, but that information was readily available via online search. Shortly after the initial voice message, we received an email message asking us to confirm via phone that we had indeed sent the message, and to read their responsible disclosure policy.

We received a follow-up from the head of security at the immediate parent company explaining that they were unable to corroborate the data from our study with any of their internal information, and expressed understandable frustration that so much time had passed between event and notification.

The company was very interested in obtaining all related information available, including communicating with the email provider. We provided them with timestamps and IP addresses associated with all relevant logins. Pages on their sites list usernames, and the company asked if these could have been used by an attacker to brute-force guess passwords either at the sites or the email provider. While our email provider provides checking against brute-forcing, sites E and F do not. But if indeed this is what occurred, then it represents a compromise consistent with Tripwire's goals.

6.3.6 Site G (RSS Feeds, Alexa rank 17500). The owner of site G responded three days after notification inquiring about our dataset. Upon explaining our data and methodology, the owner responded that, after looking for a while, he did find some SQL commands that were improperly escaped, and he knew that his server was under constant SSH brute-forcing attempts, but that he had not been aware of any prior breach. The owner also explained that he needed to update his installation of WordPress and that he would force a password reset after he had finished development. To date, a required password reset has yet to occur.

6.3.7 Site L (Adult, Alexa rank 11000). The owner/admin of Site L explained that he had started the site in 2007. Although he personally had 'a low level of IT knowledge', in April of 2015 he got rid of his system administrators due to their cost and because he felt they were making his job harder, not easier. Before recently migrating to a cloud provider, his site ran on approximately sixty dedicated servers. Since removing his system administrators, he has been running the site himself, and that 'being thrown in the deep end is an understatement'.

By the owner's own evaluation, security had not been a priority for the site: most of the code is from 2008, and requires PHP 5.3; passwords have only been stored in a hashed ('encrypted') form since 2015, but are still unsalted; the site suffers from some known XSS vulnerabilities that he has been intending to fix. The owner speculated that the compromise could be related to a large DDoS attack he experienced around the time of compromise which lasted several days.

He explained that he plans to prioritize salting passwords and upgrading his PHP and web server versions, although he was not presently planning on notifying users of the breach.

6.3.8 Site O (Outdoors, Alexa rank 18000). We received a response from site O less than 45 minutes after the initial notification was sent. This response, from the CEO of a competing site, explained that they had recently acquired site O from a major American travel-reviews company and that they had transferred accounts from site O to their own site in May of 2016 (the timeframe that our accounts were compromised). After we responded with our methodology and data for their site, the CEO responded saying that they were unaware of any account breach, but that they had performed a "lot of scripted testing" of logins onto their own site to ensure a smooth transition. Additional clarifications and questions regarding actions they planned on taking did not receive a reply, and users of site O have not been notified of the compromise.

6.3.9 Discussion. We believe that account information was stolen from the sites at which our registered accounts were accessed. As discussed in Section 4.4, we took many steps to ensure the integrity of our methodology, but we cannot categorically rule out the possibility that either the email provider or our own systems were compromised and that this was unwittingly the source of the account leaks.

However, the empirical evidence is inconsistent with the accounts being obtained via a breach other than at the sites at which they were registered. We had over 100,000 email accounts from the provider, only a subset of which were used to register accounts. Only a small number of those accounts were ever accessed, and all the ones accessed were used to register accounts. It also seems unlikely that an attacker would have defeated our operational security (or that of the email provider), obtained the account credentials, and then accessed only a fraction of the accounts acquired. Moreover, the odds that, in so doing, they would have happened to select just the accounts we used at these sites seems vanishingly small. Realistically, they would also need to know the sites at which we used each account, and have some reason to specifically target the accounts at those sites.

When engaging with the sites, only one of the sites we contacted (Site C) was able to confirm that their site experienced a breach, and in this case the breach was implicitly public since the site was illegally cloned by an attacker. Even in this case, though, the owner did not explicitly notify the site users that their account information had been stolen. All other sites were unable to confirm a breach. Yet, none of the sites were able to offer another explanation for how our account information could have been stolen, and in two cases we have other corroborating evidence (Section 6.1.3).

Given this situation, there are two immediate possibilities for why sites may not inform users about a breach. One is that the sites did not have sufficient information to corroborate the breach. Indeed, consider the perspective of the sites we contacted. The disclosures we provide inform sites that they have been breached, but do not give any information about how this occurred. Tripwire provides bounds of a compromise timeframe (between account registration and first login), but those bounds can be quite broad—in our study, this period was more than 18 months in the most extreme case. Further, while sites naturally asked to know which accounts on their service triggered detection, there is little information to be gleaned from these accounts, provided the compromise occurred after registration time. Such information provides sites with little insight on where to look for evidence of a compromise, nor how to prevent it from happening again.

Finally, even if a site believes Tripwire's evidence that a breach occurred, the specifics may not be sufficient to convince sites to incur the cost of acknowledging a breach. There are substantial potential legal and financial repercussions of publicly acknowledging a breach, particularly for sites run by businesses. The knowledge of a small number of leaked accounts, internally confirmed or not, may not constitute sufficient risk given the potential cost.

6.4 Attacker behavior

Lastly, we characterize the activity of attackers with the stolen email accounts [38]. In general, most attackers accessed the accounts repeatedly over the observation period. Although some accounts were shut down for sending spam, in many cases attackers have not taken active steps to use the accounts beyond siphoning email. Accounts appear to be accessed through a global network of predominantly compromised residential machines acting as proxies, typically via IMAP. Account login timing and frequency suggests that credentials are being fed into automated collection systems. We have released our data for these accesses with lightly reduced granularity, which we discuss in more detail in Secion 7.4.

6.4.1 Login frequency. Table 3 lists the email accounts accessed, the type of password used by the account, the total number of accesses, and the number of days between account registration and first remote access, number of days since last access (as of Feb. 1, 2017), and the number of days between the first and last accesses.

	Type	# Logins	Until	Since	Frozen	Days Accessed
a1	hard	1	175	569	N	0
a2	easy	1	141	569	N	0
b1	easy	83	153	28	Y	518
c1	easy	27	167	45	N	496
d1	hard	10	195	53	N	452
d2	easy	4	193	177	N	328
d3	hard	85	35	15	N	366
e1	easy	22	214	26	N	459
f1	easy	119	214	54	N	430
g1	hard	181	235	10	N	458
g2	easy	62	311	2	Y	385
h1	hard	42	3	132	Y	296
h2	easy	48	38	133	Y	88
i1	easy	58	345	5	N	358
i2	hard	94	353	133	Y	228
j1	easy	3	374	299	N	26
j2	hard	8	378	78	N	245
k1	easy	3	381	301	Y	16
k2	hard	1	383	318	N	0
l1	easy	9	387	14	N	298
m1	hard	207	392	2	Y	306
m2	easy	363	390	65	Y	244
n1	easy	23	439	22	N	237
o1	easy	1	447	252	N	0
p1	easy	3	533	162	N	13
q1	easy	9	548	43	N	108
q2	hard	18	553	37	N	110
r1	hard	38	571	12	N	118
r2	hard	39	250	8	N	121
s1	easy	6	639	68	N	1

Table 3: Number and date range of login activity for compromised accounts. "Until" indicates the number of days between registration and first access. "Since" indicates the number of days since the most recent login (as of last check).

'Frozen' indicates whether the account has been frozen by our email provider due to suspicious activity. The account aliases encode the sites at which they were registered (e.g., we registered account a1 at site A).

Two of the sites (E and F) show periodic, temporally aligned logins. Manual inspection revealed that these two sites were owned and operated by the same entity, and appear to use the same registration backend. Otherwise, we found no discernible pattern across accounts regarding access timing. The data shows both recurring and non-recurring logins for sites: at the most popular site A, both accounts were only accessed once, while account m1 has been accessed 392 times. Accounts from several of the sites exhibit behavior consistent with ongoing observation or scraping rather than simply verifying credentials.

6.4.2 Bursty logins. Although no overall pattern emerges, eleven of the accounts have bursty login behavior where multiple logins occur to the same account from different IP addresses in rapid succession of each other. In the peak case, g1 experiences 46 distinct IPs accessing the account over 10 minutes. This behavior suggests that the systems used to login to accounts are very loosely coupled and failure is common. Nine of the accounts (b1, e1, f1, g1, k1, k2, m1, m2, r2) experience bursts of logins wherein a single IP accesses the same account dozens or hundreds of times within a few seconds. In the extreme cases, this can make up more than 75% of the logins seen for an account.

6.4.3 Login IPs. The IP addresses originating the account logins are consistent with large-scale botnets of leased proxies. As of our final check, a total of 1316 distinct IPs logged into the our accounts across approximately 1792 login attempts. Only 181 IPs appeared more than once in the logs, with one IP appearing 58 times (to account r2).

Based on WHOIS data, the most popular countries represented are Russia (194 IPs), China (144), USA (135), and Vietnam (89), with a total of 92 countries represented. Combining manual analysis of WHOIS with DNS, the majority of these IPs appear to be residential/consumer IPs, though several higher-volume IPs map to datacenter IPs with hosts serving legitimate content, suggesting compromised servers.[6]

6.4.4 Account activity. Since one of the goals of site compromise is to steal accounts, it is somewhat surprising that many of the stolen accounts have been relatively idle. No email account that has been accessed has received any unexpected email messages beyond a few generic spam messages.

Eight of the 27 accounts do show suspect behavior, though. The email provider forced a password reset on one of our accounts, m1, after recognizing account compromise. Accounts b1, g2, h1, h2, i2, k1 and m2 were all deactivated by the email provider for sending spam. Prior to being shut down, account g2 had had the password changed and our forwarding address removed by the attacker. For the accounts where passwords have not been changed, one possibility is that attackers are stockpiling the compromised accounts for later use or sale. Another possibility is that attackers watch these accounts for messages from sites such as banks that can be leveraged for direct monetization.

7 DISCUSSION

Though just a means to an end for our study, automated account registration is also potentially useful for others. We lead this section with more details on our registration results, lessons learned, and what would be required to further scale such a system.

Since a system like Tripwire must be robust against circumvention, we follow with a discussion on what would be required of an attacker to evade detection when compromising a site under Tripwire-like surveillance. Finally, we end with a brief discussion of what data and source we are making available.

7.1 Site eligibility

To evaluate what proportion of sites are even eligible for a Tripwire-like system, we manually visited three sets of 100 sites from the Alexa rankings, starting with Alexa ranks 1, 1,000, and 10,000, and Table 4 shows the results. On average, 6.7% of the pages failed to load, and 44.3% of pages rendered by default in a language other than English. Nearly 13% of them did not support any web registration,

[6]While we did not check extensively for spoofed reverse DNS, several spot checks suggested that reverse DNS either matched forward DNS or contained domains owned by the owner information present in WHOIS.

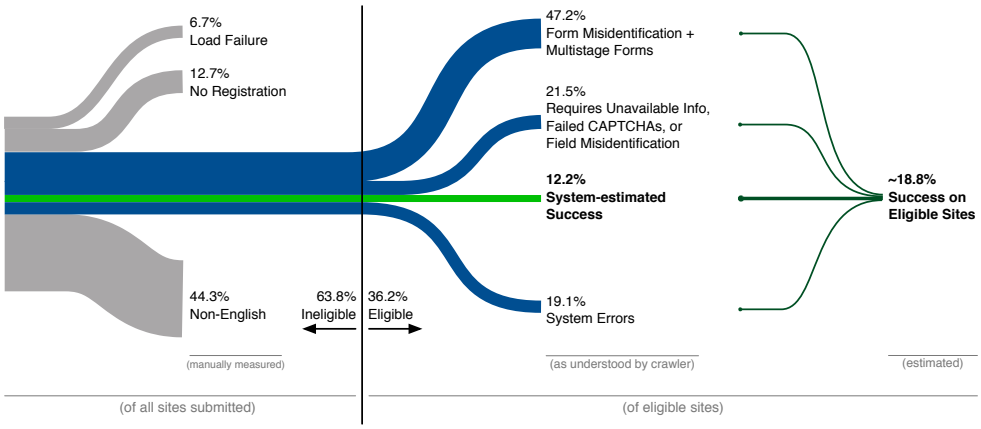

Figure 3: The various outcomes of Tripwire's registration attempts. The left and right thirds of the funnel are estimates, while the middle corresponds to crawler-measured outputs.

Start Rank	Load Failure	Not English	No Registration	Ineligible	Rest
1	3%	43%	7%	4%	43%
1,000	9%	37%	15%	6%	33%
10,000	8%	53%	16%	5%	18%
Average	6.7%	44.3%	12.7%	5.0%	31.3%
100,000	8%	43%	29%	3%	17%

Table 4: Registration eligibility of sites as determined by 100-site manual sample.

while 5% required a credit card or other information that Tripwire is unable to provide. In the end, fewer than a third of the sites were even plausible candidate sites for automated account registration.

One notable trend is the precipitous decline in the fraction of sites with viable registration pages (from 43% in the top-100 to 18% at top-10,000).[7] This trend does not affect the percentage of load failures and non-English sites, indicating that sites become decreasingly useful for registrations as one proceeds down the Alexa ranking. Although we did not use them, search engines may be an alternate source of sites to monitor.

Systematically, although we visited tens of thousands of sites across the Alexa rankings, only a fraction of them were compatible with our automated registration system. Figure 3 depicts the funnel of website registration attempts starting from the full set of URLs supplied to our automated account registration system on the left to the resulting set of successfully registered accounts on the right.

We input sites to the crawler without any additional knowledge about the sites other than URL and Alexa rank. The crawler ignores non-English or otherwise ineligible sites. The first third of the figure breaks down the reasons that our crawler is unable to register for an account, which we estimate to be about 64% of cases (see Table 4). Our crawler fails to find a registration page in about 69.2% of cases. In a manual inspection of 181 of sites where it failed, we only found valid registration pages on eight of them. This finding is consistent with an estimated false negative rate of around 5%, suggesting that if a site is completely ineligible for the current version of the crawler, the crawler is unlikely to identify a registration page on that site.

Any study that relies upon registering accounts across many sites likely has a notion of "high-value" sites, such as very popular sites. Although we originally intended to solely use automated means for registering accounts, in the end we augmented that process with manual registrations for top-ranked Alexa sites (Section 5.1). We consider the additional manual effort for high-value sites to be well worth the cost since registrations need only occur once.

7.2 Extending the crawler

The middle third of Figure 3 visually depicts the outcomes from the crawler (omitting the proportion in which no registration was possible). The final third shows the success outcome after accounting for email verification and discounting the various categories according to our success estimation methodology (Section 5.2.3). With the present system, the automated success rate is roughly 20% even when considering only eligible sites. What steps are necessary to improve the success rate?

Non-English sites alone make up more than forty percent of all sites, none of which are presently evaluated. Supporting multiple languages would be the single greatest improvement to the crawler's coverage. More tuning could also go into the heuristics used by the Tripwire crawler. Even with this and other improvements, however, automated registration on arbitrary sites is a sufficiently ill-formed problem that additional steps would be necessary.

Bot detection. In our manual study above, 19% of sites (37% of the top-100) with registration forms used some kind of test to ensure the registration form was being filled out by a human actor. If our crawler recognizes that a field is asking for human validation, it defers to third-party CAPTCHA-solving services (or, if available, a human operator). Such solving services have non-trivial error rates [37], and the crawler has no ability to handle interactive CAPTCHA services like modern reCAPTCHA [19] or KeyCAPTCHA [13].

Multi-stage forms. Around 10% of sites with registration forms that we tested have multi-step forms, in which a user completes a portion of the form before being able to advance and complete the

[7] For added scope, we also manually visited another 100 pages starting at Alexa rank 100,000 with similar results as the top-10,000.

remainder. Our crawler makes no attempt at handling these multi-step forms, resulting in both failures to recognize the first page of some registration forms (a 'no form found' result), and to fill out subsequent pages (a 'bad heuristics'/'field missing' result).

Form and field misidentification. A common failure mode for the crawler is to misidentify the meaning of individual form fields or to not recognize a given form as a registration form. Machine learning techniques would likely more reliably identify such forms and fields instead of heuristics.

Invalid identity assumptions. We chose usernames and passwords based on common policies at sites, but a small number of sites have password policies that have uncommon requirements (e.g., require special characters). Our crawler makes no attempt at inferring these policies, and since our usernames and passwords are created ahead of time, we currently have little ability to correct for these cases.

7.3 Evading Tripwire

The results presented in this paper have the advantage that no system like Tripwire (involving coordination between unrelated services to detect compromise) has previously existed, and attackers are thus unlikely to try to evade our detection. Future implementations of a similar system will not have that luxury, thus it is worth a brief discussion about what an informed attacker could do to evade Tripwire's detection. In this subsection, we assume that, at a minimum, an attacker knows that Tripwire exists, and generally how it works.

Avoiding Tripwire detection amounts to avoiding logging on to an observed email account in Tripwire. An attacker may be able to avoid this detection in a variety of ways, but each requires trade-offs. Firstly, an attacker could compromise the user database of a site not under our measurement. This is not so much an attack on Tripwire, so much as it is an acknowledgement that a system like Tripwire cannot have perfect coverage. Knowing what sites to attack requires having compromised the Tripwire operator, and thus evasion otherwise amounts to taking calculated risks on sites Tripwire was unlikely to cover. An attacker could also avoid detection by not attempting logins with the email provider, or by attempting to pick and choose which accounts to check. The odds of detection are inversely proportional to the percentage of email accounts tested. If all the attacker cares about is what approximate proportion of accounts re-use their password for the corresponding email accounts (if, for instance, the attacker was preparing the accounts for resale), then perhaps testing only a small sample may be sufficient. Alternatively, an attacker can also avoid detection by testing other accounts in lieu of testing the email account (e.g., at an OSN). As mentioned earlier, however, avoiding testing the credentials with the email provider closes off a substantial opportunity for monetization.

If the email provider for Tripwire were known, an attacker could choose to avoid checking accounts with that email provider. While effective, we chose a prominent email provider in part because a significant fraction of organic accounts on any service are likely to use this email provider, and thus this strategy is not without cost. As a happy side effect of our not disclosing our email provider partner, attackers are also left to wonder whether they must avoid checking all accounts from the largest email providers. Accounts with the largest providers, however, likely account for a significant majority of the accounts found in the breach.

An attacker may also attempt to determine specifically which accounts belong to Tripwire-like systems. Were the attacker able to determine the entire list of Tripwire accounts (for instance by compromising the Tripwire operator or the email provider), they would be able to form a complete blacklist of accounts to avoid, and completely evade Tripwire's detection. Provided that neither Tripwire's operators nor the email provider are compromised, an attacker must attempt to infer this information from signals associated with the accounts.

As discussed in Section 4.1, usernames, passwords and other identity information were chosen to look plausible, and thus hard to identify as part of the Tripwire system. If an attacker has access to information regarding initial registration, the attacker may be able to deduce which accounts are ours based on registration IP address. For our study, for ethical reasons and transparency, we registered for accounts with IP addresses that were clearly owned by our institution, but an operational deployment of Tripwire should use plausible user IPs to avoid this technique as a detection mechanism.

7.4 Data and source availability

Tripwire uses a variety of heuristics to find and fill registration forms, as well as to handle incoming email. All of these heuristics are detailed in the source code for the crawler, which is available at https://github.com/ccied/tripwire. In addition, we have provided an anonymized version of the login data at the same URL. This data consists of an entry for each login event. This record provides the account alias (e.g. 'a1'), a timestamp (rounded to the day), `/24` of the accessing IP, and login method (e.g. 'IMAP'). This anonymization was chosen to balance the desires of transparency and protecting the accounts in the Tripwire sample.

8 CONCLUSIONS

Website security is a critical problem whose personal and financial impacts are continuing to grow. While preventing and containing account compromise and disclosure are clearly of utmost importance, experience suggests that there will always be a risk that one website compromise will lead to further exploits. We have shown that this inevitable reality can be leveraged to passively monitor for compromise at a wide range of sites and detect compromises of which site operators are either unaware or unwilling to publicly disclose. A major open question, however, is how much (probative, but not particularly illustrative) evidence produced by an external monitoring system like Tripwire is needed to convince operators to act, such as notifying their users and forcing a password reset.

ACKNOWLEDGEMENTS

We thank our shepherd Theophilus Benson for his valuable guidance, and the anonymous reviewers for their helpful feedback and suggestions. We are also extremely grateful to our email provider for their very generous assistance, and to Cindy Moore and Brian Kantor for managing software and systems used in this project. This work was supported in part by National Science Foundation grants 1237264 and 1629973, the Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, and by generous research, operational and/or in-kind support via the UCSD Center for Networked Systems (CNS).

REFERENCES

[1] Alexa Top 500 global sites. http://www.alexa.com/topsites.

[2] CA Civil Code Section 1798.80-1798.84. http://www.leginfo.ca.gov/cgi-bin/displaycode?section=civ&group=01001-02000&file=1798.80-1798.84.

[3] Carbonite Accounts Targeted In Password Reuse Attack. https://www.carbonite.com/en/cloud-backup/business/resources/carbonite-blog/carbonite-password-attack/.

[4] Cluster of 'megabreaches' compromises a whopping 642 million passwords. http://arstechnica.com/security/2016/05/cluster-of-megabreaches-compromise-a-whopping-642-million-passwords/.

[5] DeCaptcher — CAPTCHA solving service, math CAPTCHA bypass, hard CAPTCHA recognition. http://de-captcher.com.

[6] Defending Against Hackers Took a Back Seat at Yahoo, Insiders Say. http://www.nytimes.com/2016/09/29/technology/yahoo-data-breach-hacking.html.

[7] Fake Name Generator. http://fakenamegenerator.com.

[8] GitHub Security Update: Reused password attack. https://github.com/blog/2190-github-security-update-reused- password-attack.

[9] GoToMyPC Password Issues: Incident Report for GoToMyPC System Status. http://status.gotomypc.com/incidents/s2k8h1xhzn4k.

[10] Hackers selling 117 million LinkedIn passwords. http://money.cnn.com/2016/05/19/technology/linkedin-hack/.

[11] Have I Been Pwned — Pwned Websites. https://haveibeenpwned.com/PwnedWebsites.

[12] [INFOGRAPHIC] Online Overload — It's Worse Than You Thought. https://blog.dashlane.com/infographic-online-overload-its-worse- than-you-thought/.

[13] KeyCAPTCHA — Innovative Anti-Spam Solution. https://www.keycaptcha.com/.

[14] No Simple Fix for Password Reuse. https://threatpost.com/no-simple-fix-for-password-reuse/118536/.

[15] No wonder hackers have it easy: Most of us now have 26 different online accounts - but only five passwords. http://www.dailymail.co.uk/sciencetech/article-2174274/No-wonder-hackers-easy-Most-26- different-online-accounts-passwords.html.

[16] NSA disguised itself as Google to spy, say reports. https://www.cnet.com/news/nsa-disguised-itself-as-google-to- spy-say-reports/.

[17] Online passwords: keep it complicated. https://www.theguardian.com/technology/2012/oct/05/online-security- passwords-tricks-hacking.

[18] Quantcast — Top Ranking International Websites. https://www.quantcast.com/top-sites.

[19] reCAPTCHA: Easy on Humans, Hard on Bots. https://www.google.com/recaptcha.

[20] WebKit. http://webkit.org.

[21] Yahoo Says Hackers Stole Data on 500 Million Users in 2014. http://www.nytimes.com/2016/09/23/technology/yahoo-hackers.html.

[22] P. Bisht and V. Venkatakrishnan. XSS-GUARD: precise dynamic prevention of cross-site scripting attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 23–43. Springer, 2008.

[23] S. W. Boyd and A. D. Keromytis. SQLrand: Preventing SQL injection attacks. In *International Conference on Applied Cryptography and Network Security*, pages 292–302. Springer, 2004.

[24] D. Canali, D. Balzarotti, and A. Francillon. The Role of Web Hosting Providers in Detecting Compromised Websites. In *Proceedings of the 22nd International World Wide Web Conference*, pages 177–188, 2013.

[25] B. Cheswick. An Evening with Berferd. In *Proc. Winter USENIX Conference, San Francisco*, pages 20–24, 1992.

[26] L. Coleman. Hacked BitcoinTalk.org User Data Goes Up For Sale On Dark Web. https://www.cryptocoinsnews.com/hacked-bitcointalk-org-user-data-goes-up-for-sale-on-dark-web/, June 2016.

[27] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The Tangled Web of Password Reuse. In *Proc. Network and Distributed System Security Symposium*, Feb. 2014.

[28] A. Doupé, L. Cavedon, C. Kruegel, and G. Vigna. Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner. In *Proc. of the 21st USENIX Security Symposium*, pages 523–538, 2012.

[29] D. Florencio and C. Herley. A Large-scale Study of Web Password Habits. In *Proceedings of the 16th International World Wide Web Conference*, pages 657–666, 2007.

[30] J. Fonseca, M. Vieira, and H. Madeira. Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection. *IEEE Transactions on Dependable and Secure Computing*, 11(5):440–453, 2014.

[31] S. Gaw and E. W. Felten. Password Management Strategies for Online Accounts. In *Proceedings of the Second Symposium on Usable Privacy and Security*, pages 44–55, 2006.

[32] C. Herley and D. Florêncio. Protecting Financial Institutions from Brute-Force Attacks. In S. Jajodia, P. Samarati, and S. Cimato, editors, *Proceedings of the 23rd International Information Security Conference*, pages 681–685, 2008.

[33] A. Hidayat. PhantomJS. http://phantomjs.org.

[34] B. Ives, K. R. Walsh, and H. Schneider. The Domino Effect of Password Reuse. *Commun. ACM*, 47(4):75–78, Apr. 2004.

[35] A. Juels and R. L. Rivest. Honeywords: Making Password-cracking Detectable. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pages 145–160, 2013.

[36] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovic. SecuBat: A Web Vulnerability Scanner. In *Proceedings of the 15th International World Wide Web Conference*, pages 247–256, 2006.

[37] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage. Re: Captchas-understanding captcha-solving services in an economic context. In *USENIX Security Symposium*, volume 10, page 3, 2010.

[38] J. Onaolapo, E. Mariconti, and G. Stringhini. What Happens After You Are Pwnd: Understanding the Use of Leaked Webmail Credentials in the Wild. In *Proceedings of the Internet Measurement Conference*, 2016.

[39] N. Perlroth and D. Gelles. Russian Hackers Amass Over a Billion Internet Passwords. http://www.nytimes.com/2014/08/06/technology/russian-gang-said-to-amass-more- than-a-billion-stolen-internet- credentials.html, August 2014.

[40] K. Soska and N. Christin. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *23rd USENIX Security Symposium*, pages 625–640, Aug. 2014.

[41] K. Thomas, D. Huang, D. Wang, E. Bursztein, C. Grier, T. J. Holt, C. Kruegel, D. McCoy, S. Savage, and G. Vigna. Framing Dependencies Introduced by Underground Commoditization. In *Workshop on the Economics of Information Security*, 2015.

[42] M. Vasek and T. Moore. Identifying risk factors for webserver compromise. In *International Conference on Financial Cryptography and Data Security*, pages 326–345, 2014.