USER AUTHENTICATION

GRAD SEC

SEP 26 2017

TODAY'S PAPERS

The Tangled Web of Password Reuse

Anupam Das[*], Joseph Bonneau[†], Matthew Caesar[*], Nikita Borisov[*] and XiaoFeng Wang[‡]
[*]University of Illinois at Urbana-Champaign
{das17, caesar, nikita}@illinois.edu
[†]Princeton University
jbonneau@princeton.edu
[‡]Indiana University at Bloomington
xw7@indiana.edu

Abstract—**Today's Internet services rely heavily on text-based passwords for user authentication. The pervasiveness of these services coupled with the difficulty of remembering large numbers of secure passwords tempts users to reuse passwords at multiple sites. In this paper, we investigate for the first time how an attacker can leverage a known password from one site to more easily guess that user's password at other sites. We study several hundred thousand leaked passwords from eleven web sites and conduct a user survey on password reuse; we estimate that 43-51% of users reuse the same password across multiple sites. We further identify a few simple tricks users often employ to transform a basic password between sites which can be used by an attacker to make password guessing vastly easier. We develop the first cross-site password-guessing algorithm, which is able to guess 30% of transformed passwords within 100 attempts compared to just 14% for a standard password-guessing algorithm without cross-site password knowledge.**

I. Introduction

Text passwords are the most common mechanism for authenticating human users of computing systems, particularly on the Internet. Password security has become a key research interest due to the pervasiveness of modern web services and their increasingly critical nature. Passwords form the foundation of security policy for a broad spectrum of online services, protecting users' financial transactions, health records and personal communications, as well as blocking intrusions into corporate, power grid, and military networks.

Security can be undermined if passwords are easy to guess and research has consistently shown that users tend to choose simple passwords that are easy to remember [20], [33]. To counter this, online services often make use of password composition policies (e.g., "the password must contain a mix of upper- and lower-case letters and at least one number"), or password meters to help users understand the strength of their passwords. Studies have shown that password composition policies along with password meters (or verbal notifications) do help users to choose stronger passwords [35], [44], [46]. However, they also increase user fatigue.

Unfortunately, the number of passwords a user must remember continues to increase, with typical Internet user estimated to have 25 distinct online accounts [10], [11], [32]. Because of this, users often *reuse* passwords across accounts on different online services. Password reuse introduces a security vulnerability as an attacker who is able to compromise one service can compromise other services protected by the same password, reducing overall security to that of the weakest site. Password reuse cannot be prevented by traditional composition policies or meters, as these tools only see passwords at a single site. Recent high-profile leaks of large numbers of passwords (including 55 000 accounts from Twitter [1], [12], [16], 450 000 accounts from Yahoo [17]–[19], and 6.5 million accounts from LinkedIn [2], [6], [8]) demonstrate that attackers can potentially gain huge lists of valid credentials for use in cross-site password attacks.

Beyond attacks exploiting exact password reuse, it is an open question if an attacker can use knowledge of a user's password at one site to more easily guess a *different* password chosen by the same user at another site. In this work, we study this question. We examine several leaked password data sets to measure password reuse across Internet sites and find that exact reuse of passwords is often mitigated by the fact that different sites have different complexity policies. However, we also find that users often use simple tricks to work around these different policies, for example making small edits to a common passphrase (e.g., adding a number *1* to the end of a password used at another site). We find that users typically use a very small set of simple rules to make these edits which can vastly improve an attacker's ability to guess passwords at other sites. For example, we were able to guess 30% of non-identical leaked password pairs within 100 attempts (10% password pairs required less than 10 attempts) while existing password cracking libraries (like John the Ripper [4]) were able to crack 14% of the passwords.

In this work, we make the following key contributions:

Permission to freely reproduce all or part of this paper for noncommercial purposes is granted provided that copies bear this notice and the full citation on the first page. Reproduction for commercial purposes is strictly prohibited without the prior written consent of the Internet Society, the first-named author (for reproduction of an entire paper only), and the author's employer if the paper was prepared within the scope of employment.
NDSS '14, 23-26 February 2014, San Diego, CA, USA
Copyright 2014 Internet Society, ISBN 1-891562-35-5
http://dx.doi.org/doi-info-to-be-provided-later

Detecting Credential Spearphishing Attacks in Enterprise Settings

Grant Ho[†] Aashish Sharma[◇] Mobin Javed[†] Vern Paxson[†*] David Wagner[†]
[†]UC Berkeley [◇]Lawrence Berkeley National Laboratory [*]International Computer Science Institute

Abstract

We present a new approach for detecting credential spearphishing attacks in enterprise settings. Our method uses features derived from an analysis of fundamental characteristics of spearphishing attacks, combined with a new non-parametric anomaly scoring technique for ranking alerts. We evaluate our technique on a multi-year dataset of over 370 million emails from a large enterprise with thousands of employees. Our system successfully detects 6 known spearphishing campaigns that succeeded (missing one instance); an additional 9 that failed; plus 2 successful spearphishing attacks that were previously unknown, thus demonstrating the value of our approach. We also establish that our detector's false positive rate is low enough to be practical: on average, a single analyst can investigate an entire month's worth of alerts in under 15 minutes. Comparing our anomaly scoring method against standard anomaly detection techniques, we find that standard techniques using the same features would need to generate at least 9 times as many alerts as our method to detect the same number of attacks.

1 Introduction

Over the past several years, a litany of high-profile breaches has highlighted the growing prevalence and potency of spearphishing attacks. Leveraging these attacks, adversaries have successfully compromised a wide range of government systems (e.g., the US State Department and the White House [1]), prominent companies (e.g., Google and RSA [3]), and recently, political figures and organizations (e.g., John Podesta and the DNC [21]).

Unlike exploits that target technical vulnerabilities in software and protocols, spearphishing is a type of social engineering attack where the attacker sends a targeted, deceptive email that tricks the recipient into performing some kind of dangerous action for the adversary. From an attacker's perspective, spearphishing requires little technical sophistication, does not rely upon any specific vulnerability, eludes technical defenses, and often succeeds. From a defender's perspective, spearphishing is difficult to counter due to email's susceptibility to spoofing and because attackers thoughtfully handcraft their attack emails to appear legitimate. For these reasons, there are currently no generally effective tools for detecting or preventing spearphishing, making it the predominant attack for breaching valuable targets [17].

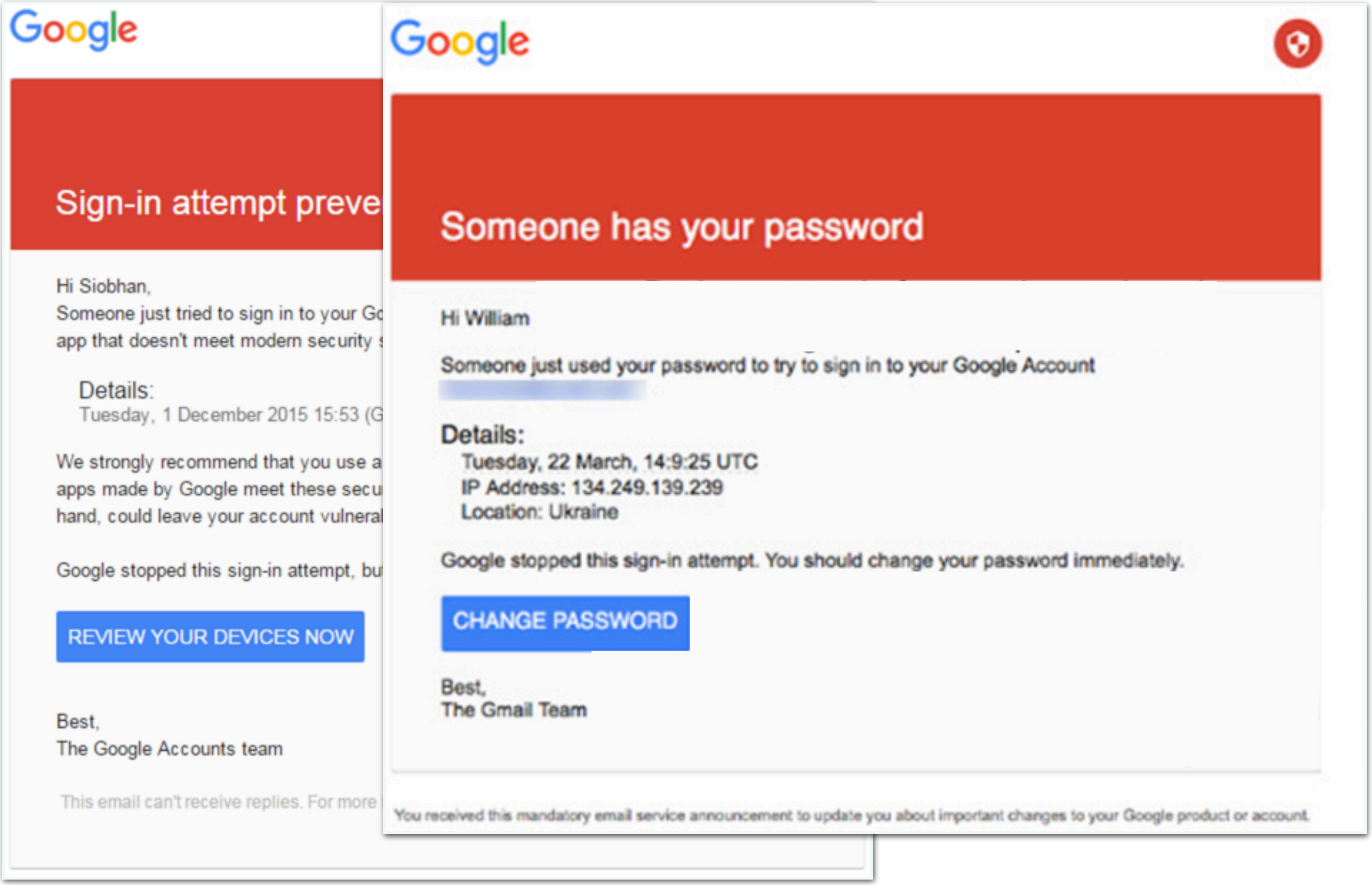
Spearphishing attacks take several forms. One of the most well-known involves an email that tries to fool the recipient into opening a malicious attachment. However, in our work, which draws upon several years worth of data from the Lawrence Berkeley National Lab (LBNL), a large national lab supported by the US Department of Energy, none of the successful spearphishing attacks involved a malicious attachment. Instead, the predominant form of spearphishing that LBNL encounters is *credential spearphishing*, where a malicious email convinces the recipient to click on a link and then enter their credentials on the resulting webpage. For an attachment-driven spearphish to succeed against a site like LBNL, which aggressively scans emails for malware, maintains frequently updated machines, and has a team of several full-time security staff members, an attacker will often need to resort to an expensive zero-day exploit. In contrast, credential spearphishing has an incredibly low barrier to entry: an attacker only needs to host a website and craft a deceptive email for the attack to succeed. Moreover, with widespread usage of remote desktops, VPN applications, and cloud-based email providers, stolen credentials often provide attackers with rich information and capabilities. Thus, although other forms of spearphishing constitute an important threat, credential spearphishing poses a major and unsolved threat in-and-of itself.

Our work presents a new approach for detecting credential spearphishing attacks in enterprise settings. This domain proves highly challenging due to base-rate issues. For example, our enterprise dataset contains 370 million emails, but fewer than 10 known instances of spearphishing. Consequently, many natural methods fail, because their false positive rates are too high: even a false positive rate as low as 0.1% would lead to 370,000 false alarms. Additionally, with such a small number of known spearphishing instances, standard machine learning approaches seem unlikely to succeed: the training set is too small and the class imbalance too extreme.

To overcome these challenges, we introduce two key contributions. First, we present an analysis of character-

USENIX Association 26th USENIX Security Symposium 469

SPEARPHISHING ATTACKS



Google

Sign-in attempt preve

Hi Siobhan,

Someone just tried to sign in to your Go

app that doesn't meet modern security s

Details:

Tuesday, 1 December 2015 15:53 (G

We strongly recommend that you use a

apps made by Google meet these secu

hand, could leave your account vulneral

Google stopped this sign-in attempt, bu

REVIEW YOUR DEVICES NOW

Best,

The Google Accounts team

This email can't receive replies. For more

Google

Someone has your password

Hi William

Someone just used your password to try to sign in to your Google Account

[redacted]

Details:

Tuesday, 22 March, 14:9:25 UTC

IP Address: 134.249.139.239

Location: Ukraine

Google stopped this sign-in attempt. You should change your password immediately.

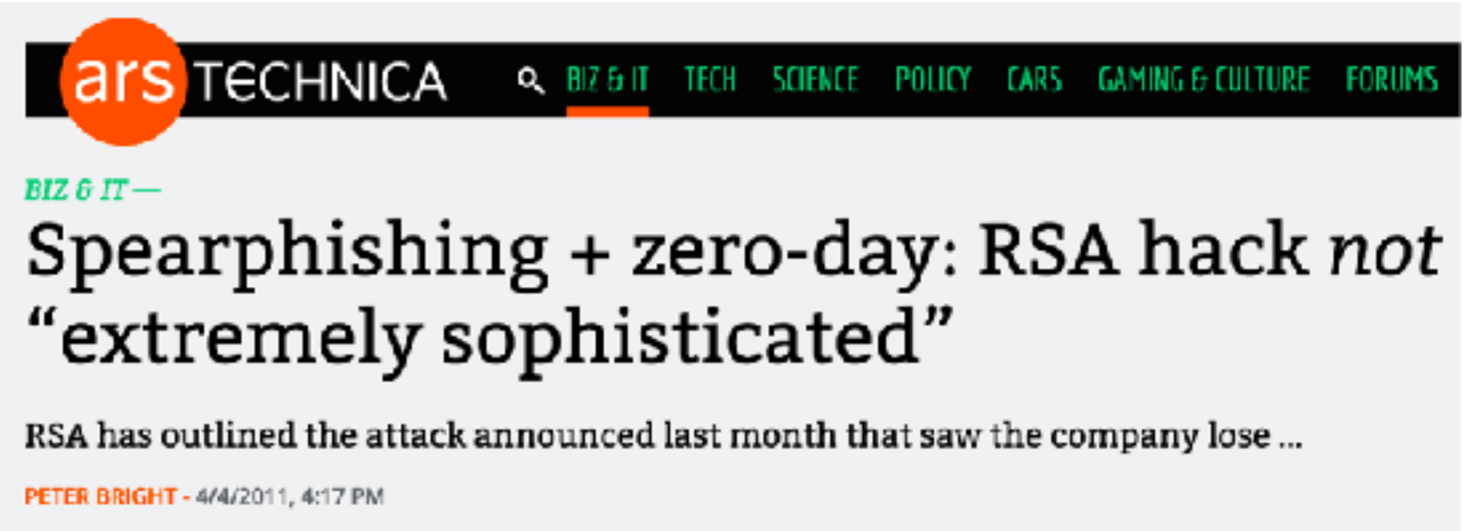
CHANGE PASSWORD

Best,

The Gmail Team

You received this mandatory email service announcement to update you about important changes to your Google product or account.

SPEARPHISHING ATTACKS

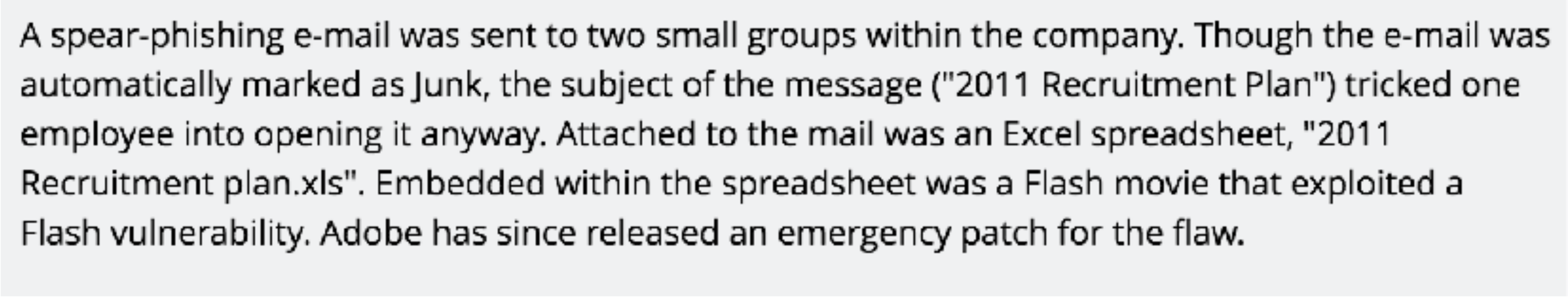


BIZ & IT —

Spearphishing + zero-day: RSA hack *not* "extremely sophisticated"

RSA has outlined the attack announced last month that saw the company lose ...

PETER BRIGHT - 4/4/2011, 4:17 PM



A spear-phishing e-mail was sent to two small groups within the company. Though the e-mail was automatically marked as Junk, the subject of the message ("2011 Recruitment Plan") tricked one employee into opening it anyway. Attached to the mail was an Excel spreadsheet, "2011 Recruitment plan.xls". Embedded within the spreadsheet was a Flash movie that exploited a Flash vulnerability. Adobe has since released an emergency patch for the flaw.

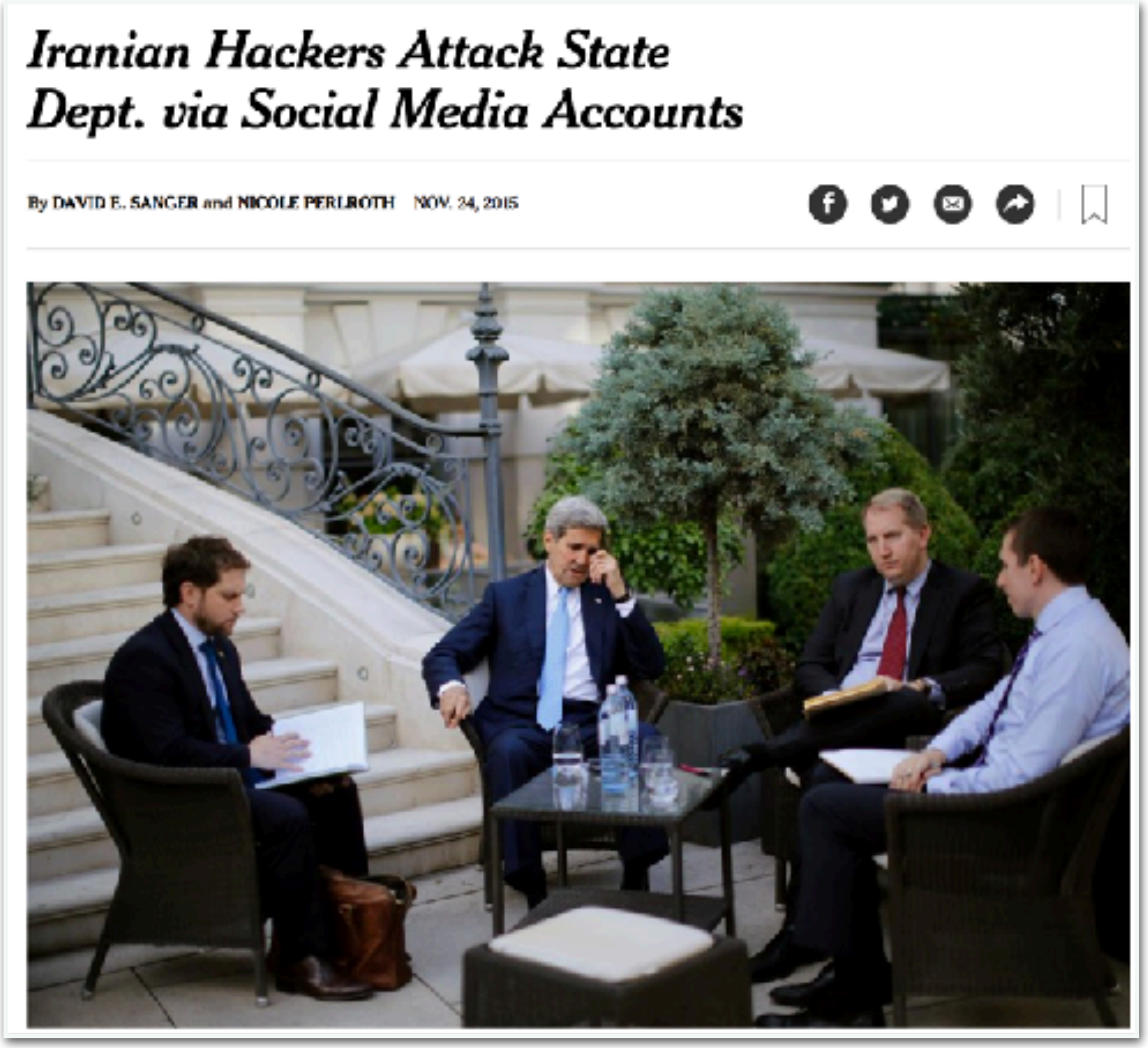
LURE

Imbue the email with a sense of trust or authority

Spoof (or log in as) the source / name

Make the topic such that they'll act quickly

SPEARPHISHING ATTACKS



Iranian Hackers Attack State Dept. via Social Media Accounts

By DAVID E. SANGER and NICOLE PERLROTH NOV. 24, 2015

For the most part, researchers said, the attacks were basic "spear phishing" attempts, in which attackers tried to **lure** their victims into clicking on a malicious link, in this case by impersonating members of the news media. Iranian hackers were successful in more than a quarter of their attempts.

LURE

Imbue the email with a sense of trust or authority

Spoof (or log in as) the source / name

Often: make the topic such that they'll act quickly

SPEARPHISHING ATTACKS

LURE

Imbue the email with a sense of trust or authority

Spoof (or log in as) the source / name

Often: make the topic such that they'll act quickly

EXPLOIT

Malicious attachment

URLs that get users to reveal more info

Out-of-band attacks (e.g., wiring money)

THREAT MODEL

Attacker can send arbitrary emails

Can convince the recipient to click on URLs

Security goal: Detect and stop with low false positives

IDEA: FLAG NEW 'FROM' ADDRESSES

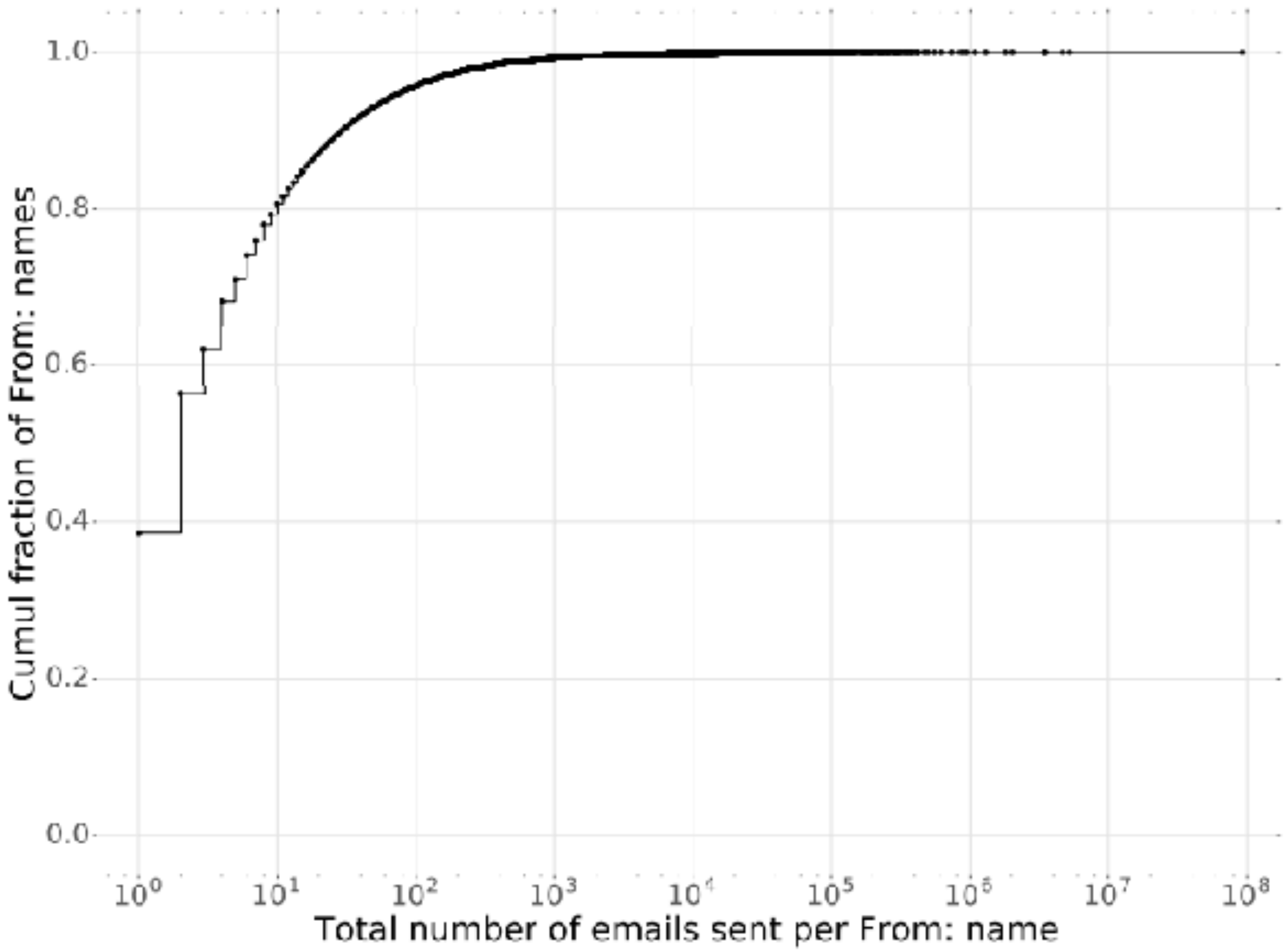


Figure 2: Distribution of the number of emails sent per `From` name. Nearly 40% of all `From` names appear in only one email and over 60% of all `From` names appear in three or fewer emails.

Most From names are new!

Too many false positives ⟹ too many admin checks ⟹ fatigue/failure

Benign behavior is diverse

IDEA: FLAG ADDRESSES WITH MANY 'FROM' NAMES

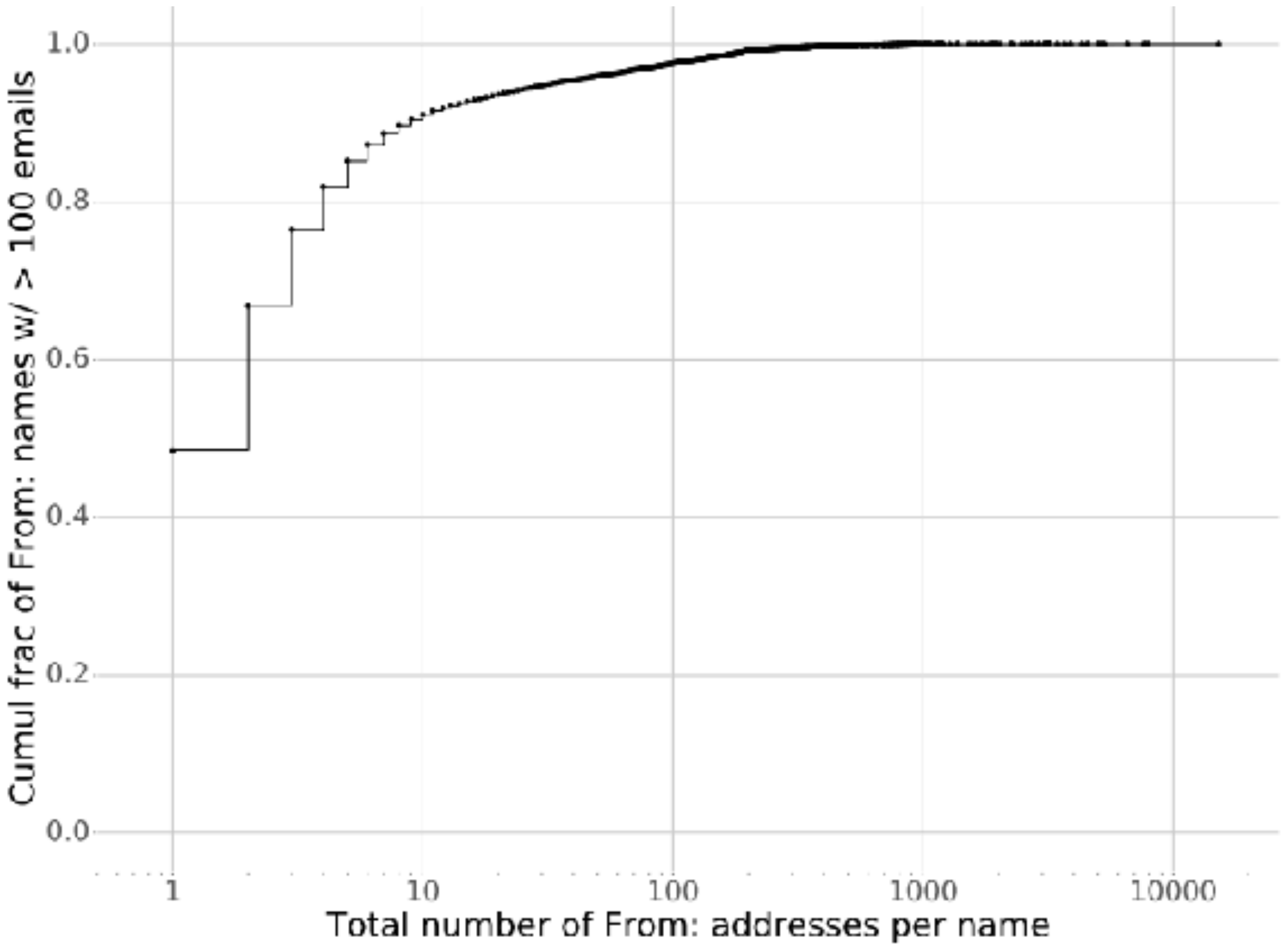
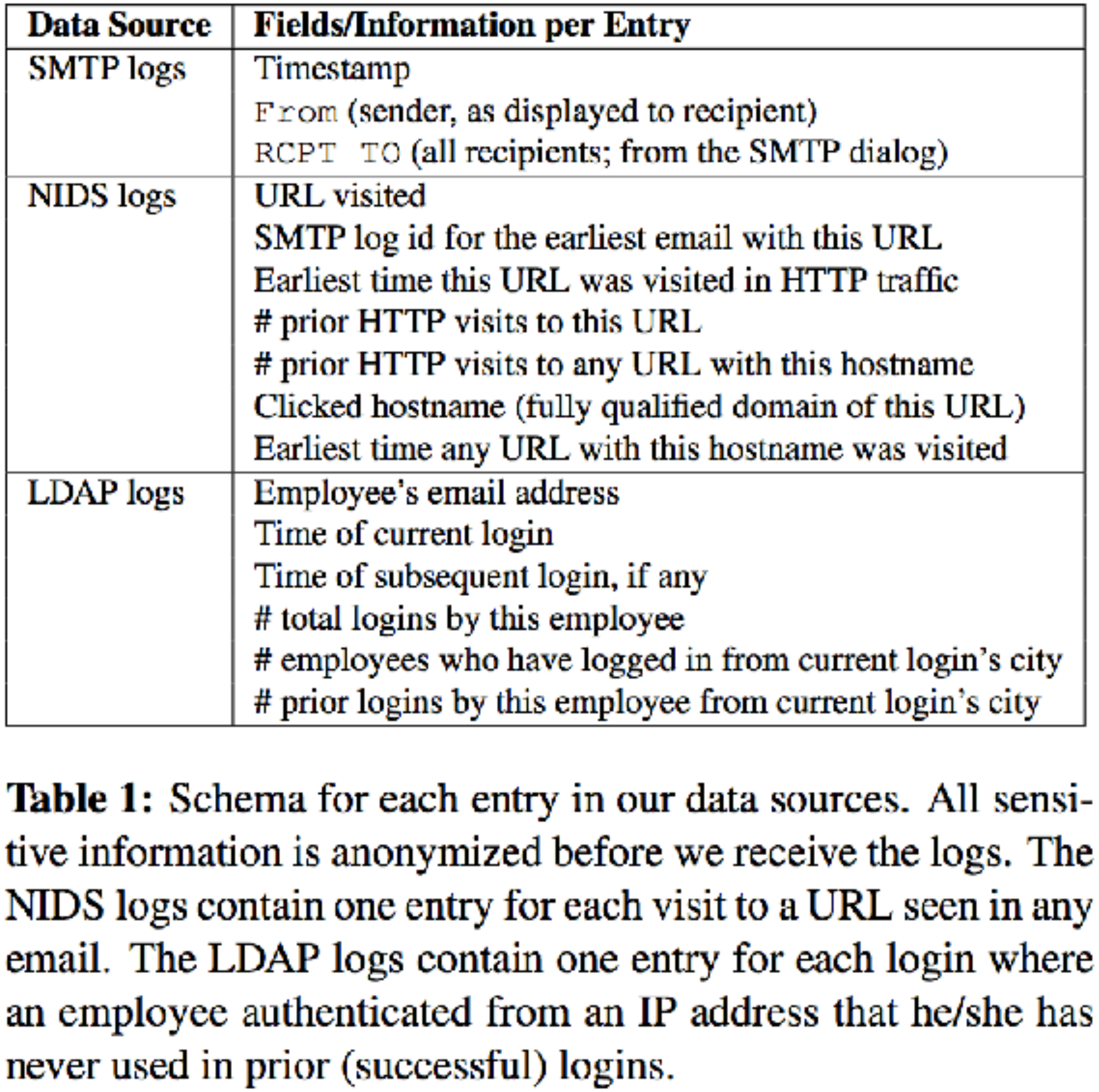


Figure 3: Distribution of the total number of From addresses per From name (who send over 100 emails) across all emails sent by the From name. Over half (52%) of these From names sent email from two or more From addresses (i.e., have at least one new From address).

Most addresses have ≥2 From names

Benign behavior is diverse

DATASETS



Data Source	Fields/Information per Entry
SMTP logs	Timestamp From (sender, as displayed to recipient) RCPT TO (all recipients; from the SMTP dialog)
NIDS logs	URL visited SMTP log id for the earliest email with this URL Earliest time this URL was visited in HTTP traffic # prior HTTP visits to this URL # prior HTTP visits to any URL with this hostname Clicked hostname (fully qualified domain of this URL) Earliest time any URL with this hostname was visited
LDAP logs	Employee's email address Time of current login Time of subsequent login, if any # total logins by this employee # employees who have logged in from current login's city # prior logins by this employee from current login's city

Table 1: Schema for each entry in our data sources. All sensitive information is anonymized before we receive the logs. The NIDS logs contain one entry for each visit to a URL seen in any email. The LDAP logs contain one entry for each login where an employee authenticated from an IP address that he/she has never used in prior (successful) logins.

Email server logs

Network Intrusion Detection System logs

User accounts & login attempt logs

373M+ emails

APPROACH

Analyze every email that contains a link that a user clicked on

***Features** for Lure vs.*
***Features** for Exploit*

Domain reputation vs.
Sender reputation

***Intuition:** if few employees from the enterprise have visited URLs from the link's domain, then we would like to treat a visit to the email's link as suspicious*

FEATURES

Domain reputation [NIDS logs]

- # prior visits to any URL with the same FQDN as the clicked URL (global count across all employees' visits)
- # days between the first visit by any employee to a URL on the clicked link's FQDN and the time when the clicked link's email initially arrived

Sender reputation - name spoofer [SMTP logs]

- # previous days where we saw an email whose From header contains the same name *and* address as the email being scored
- trustworthiness of the name in its From header
  # weeks where this name sent at least one email for every weekday of the week

FEATURES

Sender reputation - previously unseen attacker [SMTP logs]

Assumption: attacker will seek to avoid detection
and will therefore re-use the same address

- # prior days that the From **name** has sent email
- # prior days that the From **address** has sent email

Sender reputation - lateral attacker [LDAP logs]

Whether the email was sent during a login session where the sender-employee logged in using an IP address that the sender-employee has never used before. If so get the login country ***C***

- # distinct employees logged in from ***C***
- # previous logins where this sender-employees logged in from ***C***

ALERT BUDGET

THREAT MODEL

Attacker can send arbitrary emails

Can convince the recipient to click on URLs

Security goal: Detect and stop with ***low false positives***

Human limitations of the administrator

Human limitations of the user

So as not to overload administrators,
set thresholds to limit the number of total alerts per day

ALERT BUDGET

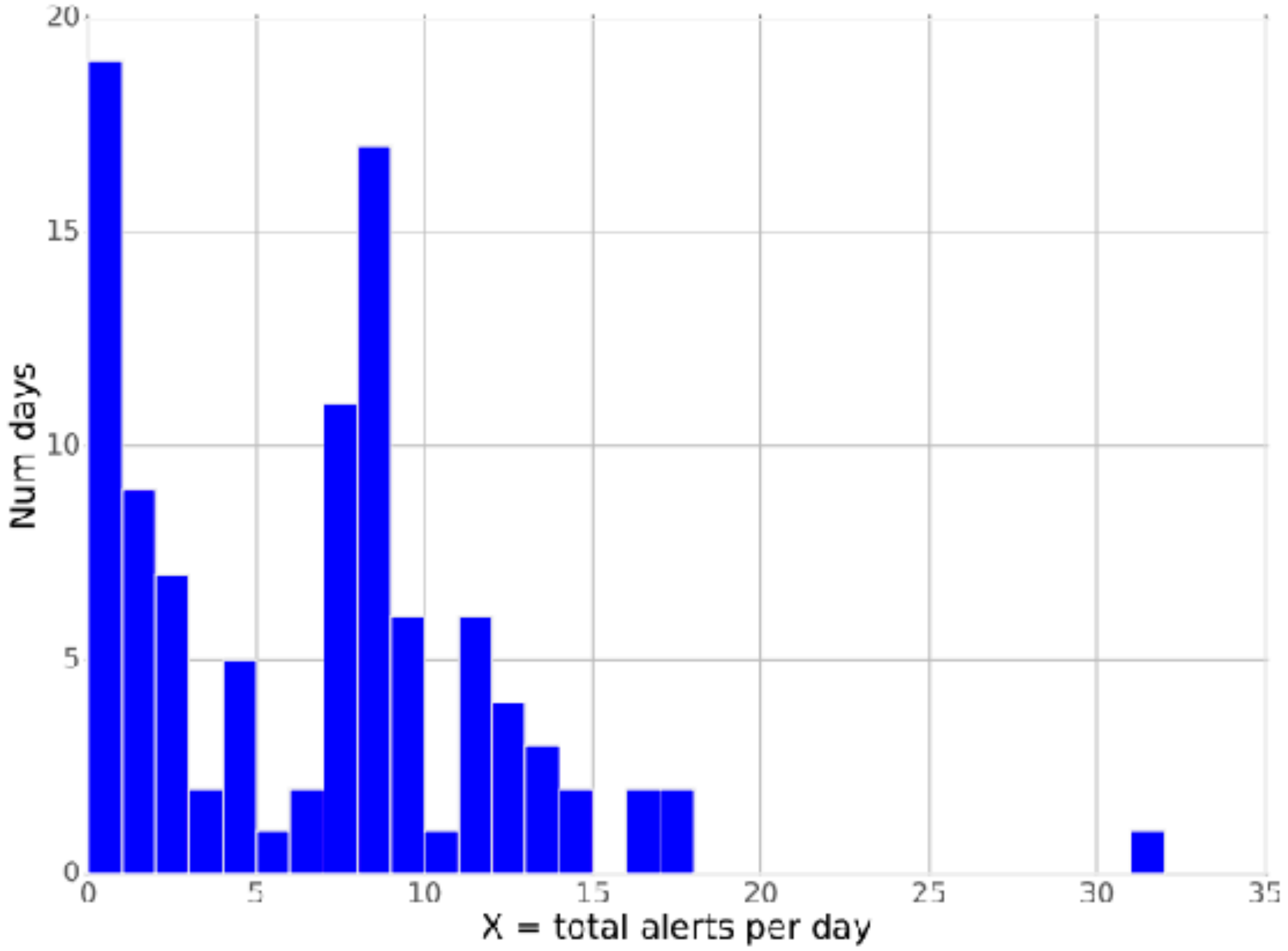


Figure 6: Histogram of the total number of daily alerts generated by our real-time detector (cumulative across all three sub-detectors) on 100 randomly sampled days. The median is 7 alerts/day.

Daily budget = 10

Take the N most anomalous

But when do you collect that N?

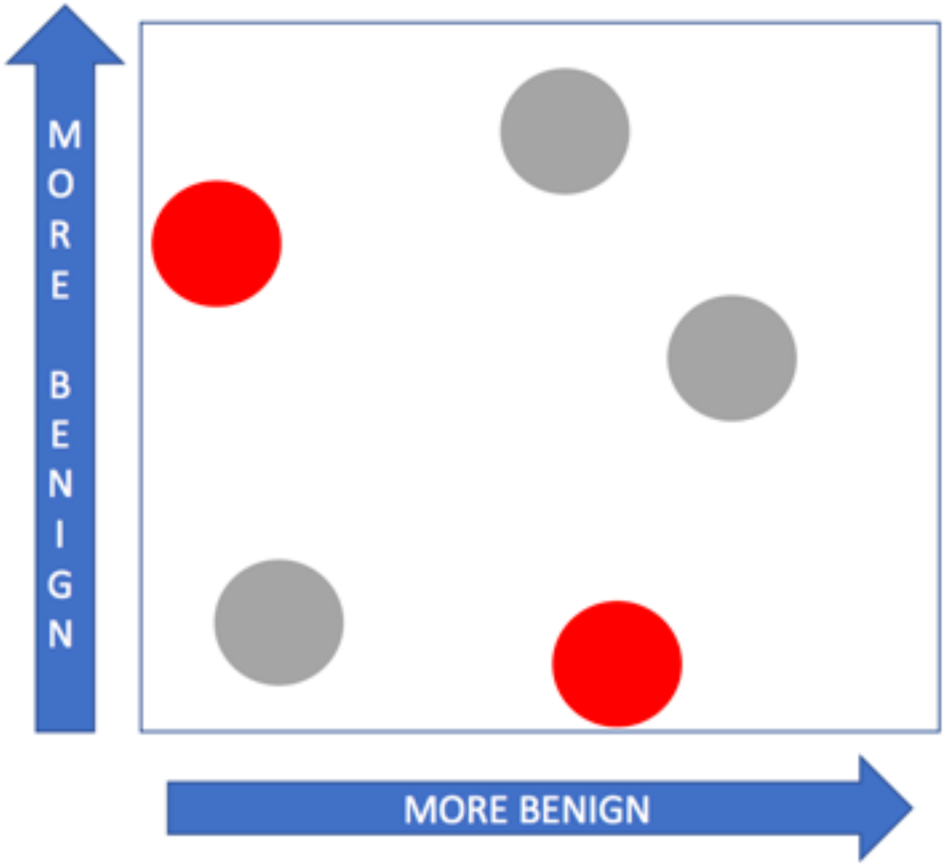
***Real-time**: Flag it if it is in the top 30N of the past month*

Sometimes it will go over/under the daily budget

DIRECTED ANOMALY SCORING (DAS)

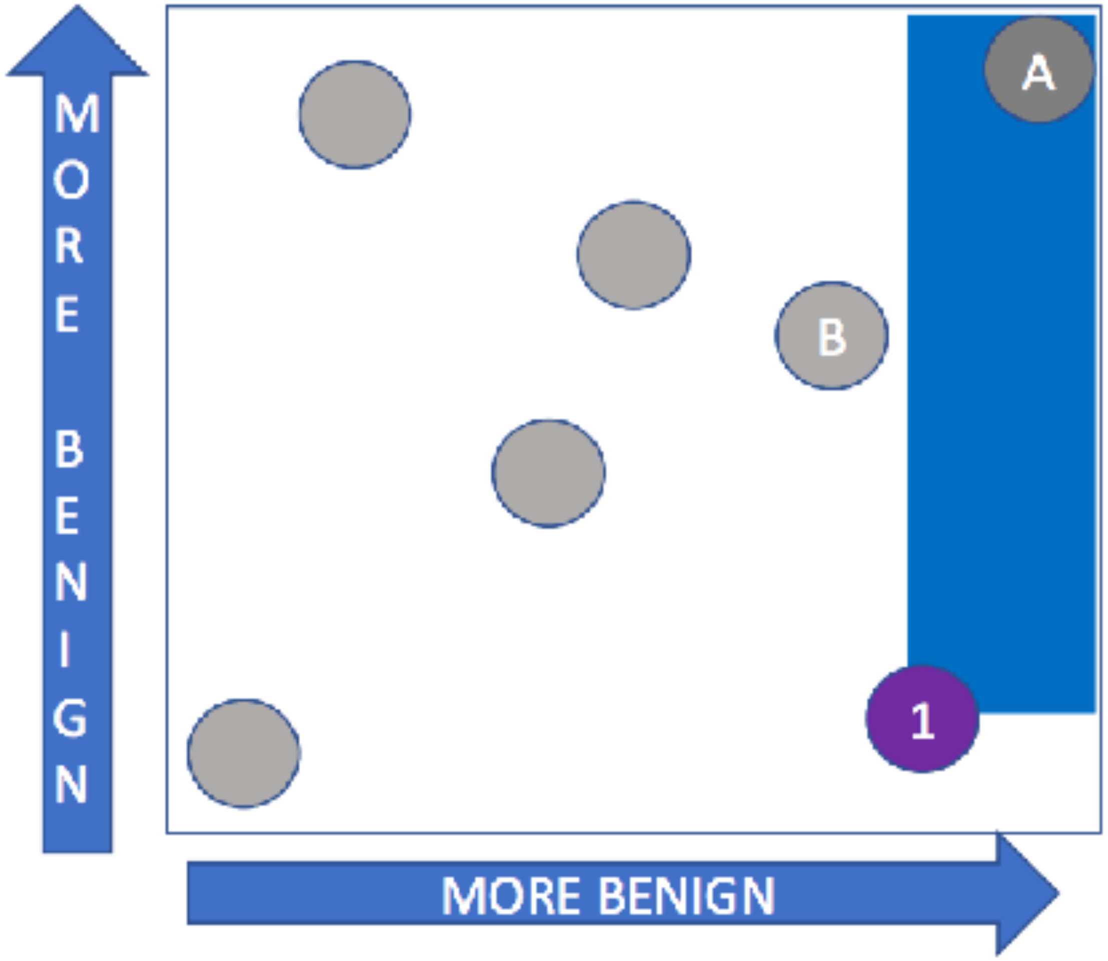
Limitations of traditional detection techniques

1. *Require hyperparameter tuning*
2. *Direction agnostic (+3std ⇔ –3std)*
3. *Alert if anomalous in only one dimension*



DIRECTED ANOMALY SCORING (DAS)

Score(Event X) = # of other events that are as benign as X in **every** dimension



FALSE NEGATIVES

Attackers leveraged the high reputation of a hosting provider

“The missed attack used a now-deprecated feature from Dropbox [7] that allowed users to host static HTML pages under one of Dropbox’s primary hostnames, which is both outside of LBNL’s NIDS visibility because of HTTPS and inherits Dropbox’s high reputation.”

SOME OF YOUR THOUGHTS ON SPEARPHISHING

- Reactive, not preventative: only captures the attack after it’s happened
- Organizations must keep detailed logs [many already do!]
  - Picked too narrow of a spearphishing attack for this system to be widely useful (doesn’t take the content into account)
- What’s the extent to which it can be applied in non-enterprise systems?
  - Requires prior data; this prior data can’t come from other enterprises [broad problem: sharing training without divulging private data]
- While I do believe their claim that DAS probably would be better in practice, I’m not sure they did enough to prove it.
- The system was able to detect 2 previously unknown attacks which shows how unreliable the known attack base is.
- Why did you show us this paper? Is this defense method the most commonly used?

PASSWORD REUSE

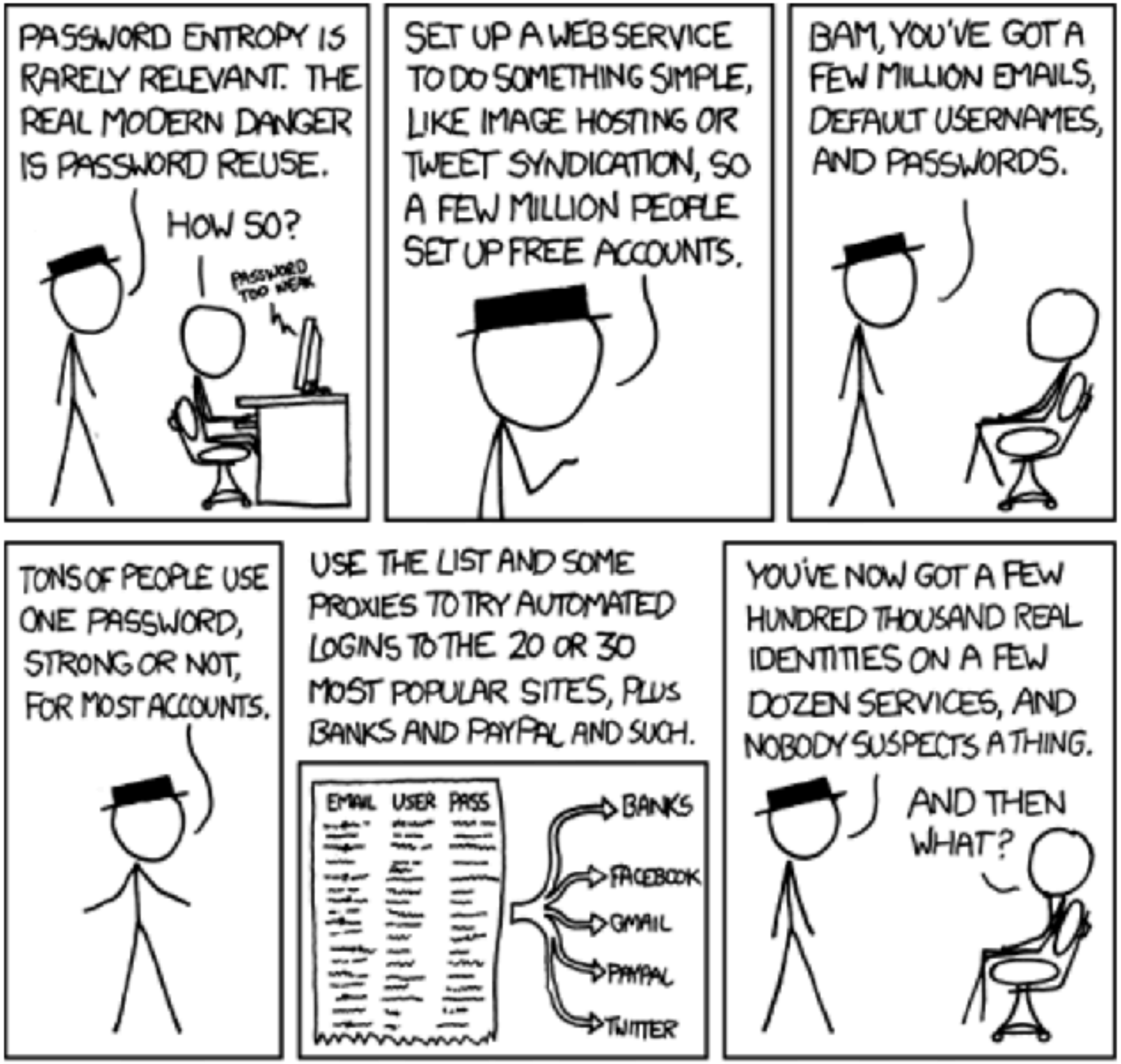
Admit it – you do this

But how would you go about measuring it?

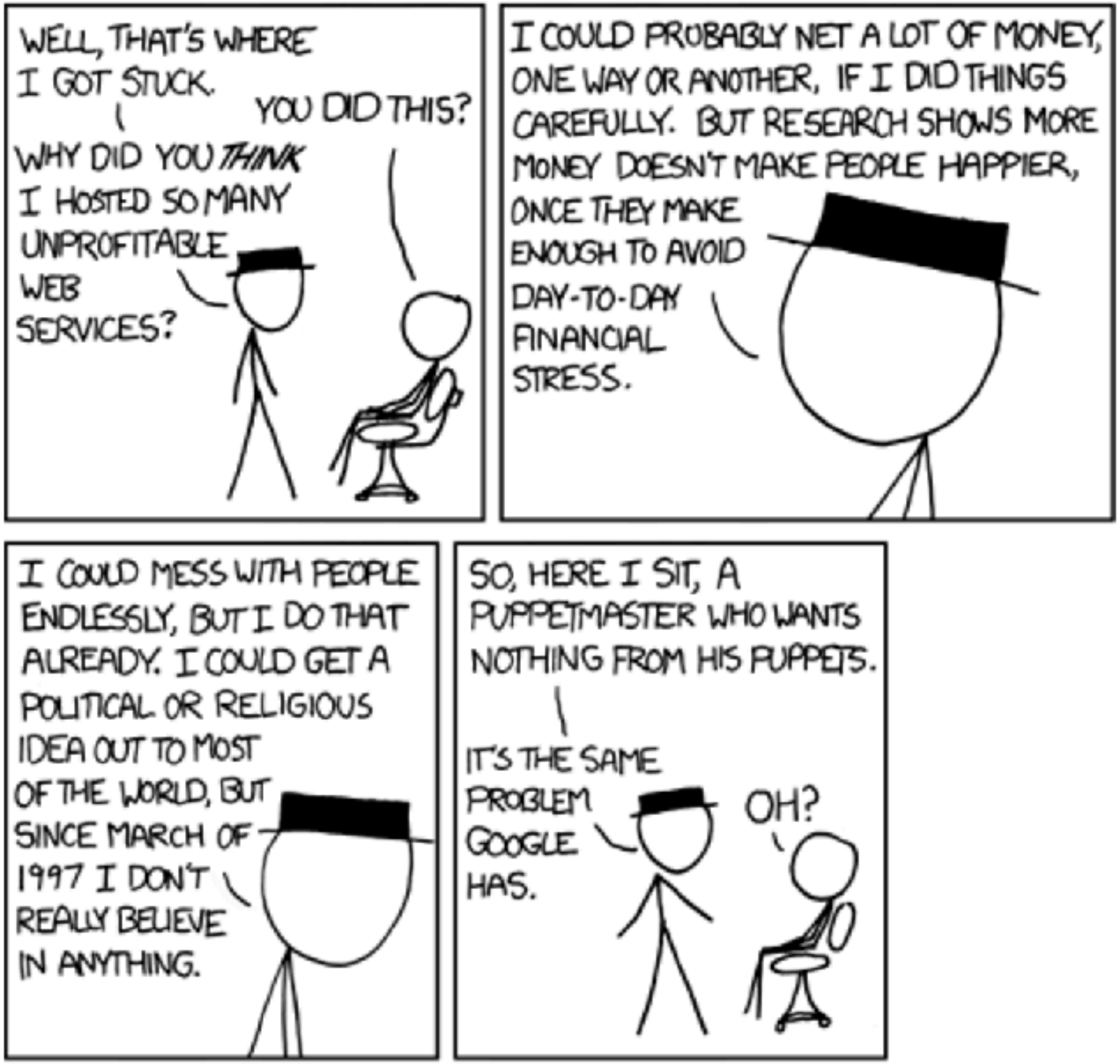
SOME OF YOUR THOUGHTS ON PASSWORD REUSE

- Disappointing to see they didn't have any great ideas for countermeasure
- I wonder how relevant this problem still is, though, given the widespread adoption nowadays of two-factor authentication schemes
- I wonder how the dangers of password similarities could be conveyed to users in a way that captures the same immediacy but for cross-site use cases
- This subject has always been something I thought of but never actually looked into. I love how people add emoticons to their passwords
- Should we all use password managers?
    - I think I will start to use a password manager

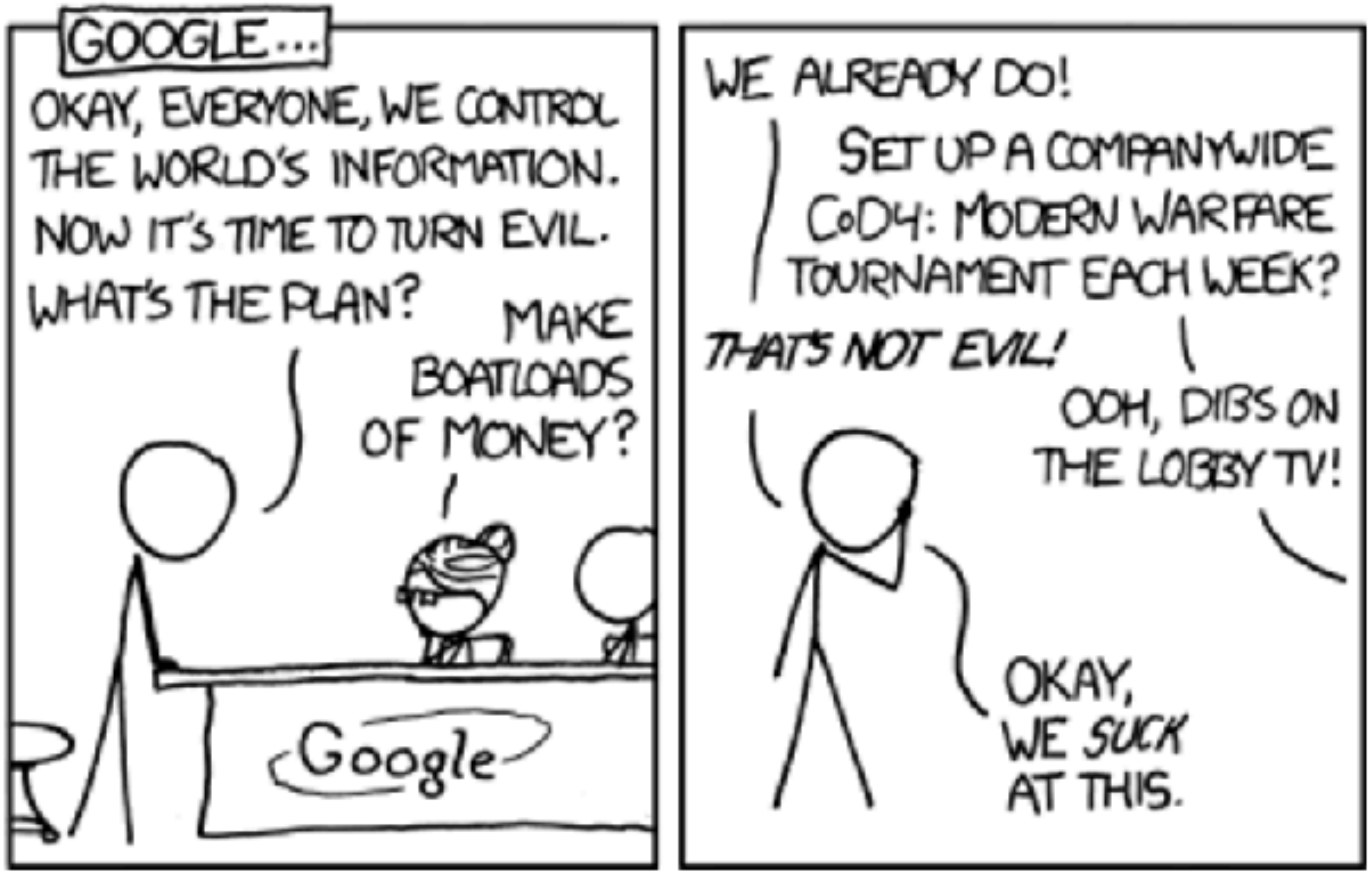
SOME OF YOUR THOUGHTS ON PASSWORD REUSE


PASSWORD ENTROPY IS RARELY RELEVANT. THE REAL MODERN DANGER IS PASSWORD REUSE.
HOW SO?
PASSWORD TOO WEAK
SET UP A WEB SERVICE TO DO SOMETHING SIMPLE, LIKE IMAGE HOSTING OR TWEET SYNDICATION, SO A FEW MILLION PEOPLE SET UP FREE ACCOUNTS.
BAM, YOU'VE GOT A FEW MILLION EMAILS, DEFAULT USERNAMES, AND PASSWORDS.
TONS OF PEOPLE USE ONE PASSWORD, STRONG OR NOT, FOR MOST ACCOUNTS.
USE THE LIST AND SOME PROXIES TO TRY AUTOMATED LOGINS TO THE 20 OR 30 MOST POPULAR SITES, PLUS BANKS AND PAYPAL AND SUCH.
EMAIL
USER
PASS
BANKS
FACEBOOK
GMAIL
PAYPAL
TWITTER
YOU'VE NOW GOT A FEW HUNDRED THOUSAND REAL IDENTITIES ON A FEW DOZEN SERVICES, AND NOBODY SUSPECTS A THING.
AND THEN WHAT?

SOME OF YOUR THOUGHTS ON PASSWORD REUSE


WELL, THAT'S WHERE I GOT STUCK.
YOU DID THIS?
WHY DID YOU *THINK* I HOSTED SO MANY UNPROFITABLE WEB SERVICES?
I COULD PROBABLY NET A LOT OF MONEY, ONE WAY OR ANOTHER, IF I DID THINGS CAREFULLY. BUT RESEARCH SHOWS MORE MONEY DOESN'T MAKE PEOPLE HAPPIER, ONCE THEY MAKE ENOUGH TO AVOID DAY-TO-DAY FINANCIAL STRESS.
I COULD MESS WITH PEOPLE ENDLESSLY, BUT I DO THAT ALREADY. I COULD GET A POLITICAL OR RELIGIOUS IDEA OUT TO MOST OF THE WORLD, BUT SINCE MARCH OF 1997 I DON'T REALLY BELIEVE IN ANYTHING.
SO, HERE I SIT, A PUPPETMASTER WHO WANTS NOTHING FROM HIS PUPPETS.
IT'S THE SAME PROBLEM GOOGLE HAS.
OH?

SOME OF YOUR THOUGHTS ON PASSWORD REUSE



GOOGLE...
OKAY, EVERYONE, WE CONTROL THE WORLD'S INFORMATION. NOW IT'S TIME TO TURN EVIL. WHAT'S THE PLAN?
MAKE BOATLOADS OF MONEY?
Google
WE ALREADY DO!
SET UP A COMPANYWIDE CoD4: MODERN WARFARE TOURNAMENT EACH WEEK?
THAT'S NOT EVIL!
OOH, DIBS ON THE LOBBY TV!
OKAY, WE SUCK AT THIS.

YOUR BOTNET IS MY BOTNET

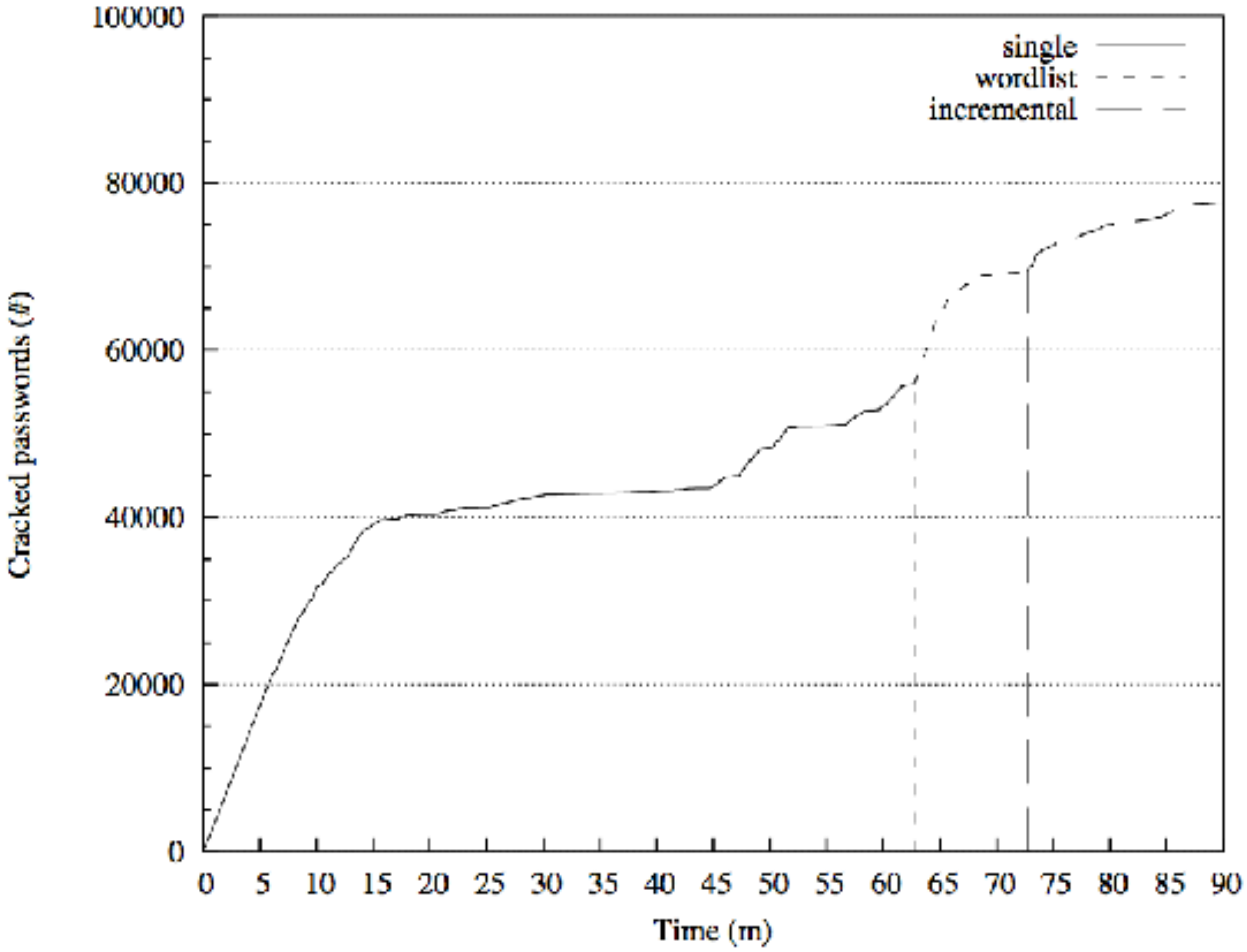
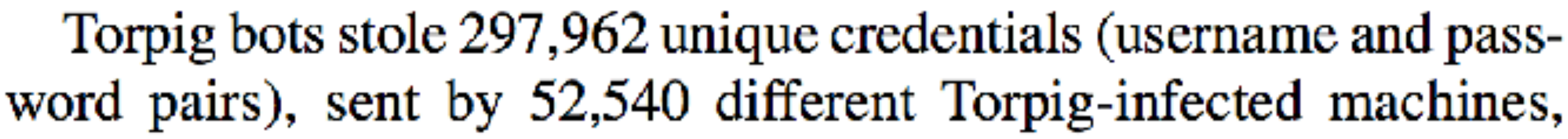


Figure 13: Number of passwords cracked in 90 minutes by the John the Ripper password cracker tool. Vertical lines indicate when John switches cracking mode. The first vertical line represents the switching from simple transformation techniques (“single” mode) to wordlist cracking, the second from wordlist to brute-force (“incremental”).

YOUR BOTNET IS MY BOTNET



Torpig bots stole 297,962 unique credentials (username and password pairs), sent by 52,540 different Torpig-infected machines,