Gone in Six Characters: Short URLs Considered Harmful for Cloud Services

Martin Georgiev
independent *

Vitaly Shmatikov
Cornell Tech

Abstract

Modern cloud services are designed to encourage and support collaboration. To help users share links to online documents, maps, etc., several services, including cloud storage providers such as Microsoft OneDrive[1] and mapping services such as Google Maps, directly integrate URL shorteners that convert long, unwieldy URLs into short URLs, consisting of a domain such as `1drv.ms` or `goo.gl` and a short token.

In this paper, we demonstrate that the space of 5- and 6-character tokens included in short URLs is so small that it can be scanned using brute-force search. Therefore, all online resources that were intended to be shared with a few trusted friends or collaborators are effectively public and can be accessed by anyone. This leads to serious security and privacy vulnerabilities.

In the case of cloud storage, we focus on Microsoft OneDrive. We show how to use short-URL enumeration to discover and read shared content stored in the OneDrive cloud, including even files for which the user did not generate a short URL. 7% of the OneDrive accounts exposed in this fashion allow anyone to *write* into them. Since cloud-stored files are automatically copied into users' personal computers and devices, this is a vector for large-scale, automated malware injection.

In the case of online maps, we show how short-URL enumeration reveals the directions that users shared with each other. For many individual users, this enables inference of their residential addresses, true identities, and extremely sensitive locations they visited that, if publicly revealed, would violate medical and financial privacy.

1 Introduction

Modern cloud services are designed to facilitate collaboration and sharing of information. To help users share links to online resources, several popular services directly integrate URL shortening services that convert long, unwieldy URLs into short URLs that are easy to send via email, instant messages, etc. For example, Microsoft OneDrive cloud storage service uses the `1drv.ms` domain[2] for its short URLs, Google Maps uses `goo.gl`, Bing Maps uses `binged.it`, etc. In this paper, we investigate the security and privacy consequences of this design decision.

First, we observe that the URLs created by many URL shortening services are so short that the entire space of possible URLs can be scanned or at least sampled on a large scale. We then experimentally demonstrate that such scanning is feasible. Users who generate short URLs to their online documents and maps may believe that this is safe because the URLs are "random-looking" and not shared publicly. Our analysis and experiments show that these two conditions cannot prevent an adversary from automatically discovering the true URLs of the cloud resources shared by users. Each resource shared via a short URL is thus effectively *public* and can be accessed by anyone anywhere in the world.

Second, we analyze the consequences of sharing for the users of cloud storage services, using Microsoft OneDrive as our case study. Like many similar services, OneDrive (1) provides Web interfaces and APIs for easy online access to cloud-stored files, and (2) automatically synchronizes files between users' personal devices and cloud storage. We demonstrate that the discovery of a short URL for a single file in the user's OneDrive account can expose *all* other files and folders owned by the same user and shared under the same capability key or without a capability key—even files and folders that cannot be reached directly through short URLs.

Because of ethical concerns, we did not download and analyze the content of personal files exposed in this manner, but we argue that OneDrive accounts are vulnera-

*This research was done while the author was visiting Cornell Tech.

[1] OneDrive was known as SkyDrive prior to January 27, 2014.

[2] When OneDrive was SkyDrive, the domain for short URLs was `sdrv.ms`

ble to automated, large-scale privacy breaches by less scrupulous adversaries who are not constrained by ethics and law. Recent compromises of Apple's cloud services[3] demonstrated that users store very sensitive personal information in their cloud storage accounts, sometimes intentionally and sometimes accidentally due to automatic synchronization with their mobile phones.

More than 7% of OneDrive and Google Drive accounts we discovered by scanning short URLs contain world-writable folders. This means that an adversary can automatically inject malicious content into these accounts. Since the types of all shared files in an exposed folder are visible, the malicious content can be format-specific, for example, macro viruses for Word and Excel files, scripts for images, etc. Furthermore, the adversary can simply add executable files to these folders. Because storage accounts are automatically synchronized between the cloud and the user's devices, this vulnerability becomes a vector for automated, large-scale malware injection into the local systems of cloud-storage users.

Third, we analyze the consequences of public sharing for the users of online mapping services such as Google Maps, MapQuest, Bing Maps, and Yahoo! Maps. Short-URL enumeration reveals not only the locations that users shared with each other, but also *directions* between locations. In many cases, these directions start from or terminate at single-family residential addresses and allow inference of users' identities via cross-correlation with public directories such as White Pages. In addition, residential-to-residential directions could reveal the existence of personal relationships, including those intended to remain discreet. Even worse, many of the destinations mapped by users are highly sensitive, including hospitals, clinics, and physicians associated with specific diseases (e.g., mental illnesses and cancer) or procedures (e.g., abortion); correctional and juvenile detention facilities; places of worship; pawnbrokers, payday and car-title loan stores, etc. Analytics APIs can also be invoked on individual maps to reveal the exact time when the directions were obtained and how often the map was referred to, thus providing further context.

In summary, our analysis shows that automatically generated short URLs are a terrible idea for cloud services. When a service generates a URL based on a 5- or 6-character token for an online resource that one user wants to share with another, this resource effectively becomes public and universally accessible. Combined with other design decisions, such as Web APIs for accessing cloud-stored files and retrieving user- or resource-specific metadata, as well as automatic synchronization of files and folders between personal devices and cloud storage, universal public access to online resources leads to significant security and privacy vulnerabilities.

[3] https://en.wikipedia.org/wiki/ICloud_leaks_of_celebrity_photos

2 Background

2.1 URL Shorteners

Uniform Resource Locators (URLs) are the standard method for addressing Web content. URLs often encode session management and/or document structure information and can grow to hundreds of characters in length. The HTTP standard [35] does not specify an a priori limit on the length of a URL, but implementations impose various restrictions, limiting URLs to 2048 characters in practice [38].

Long URLs are difficult to distribute and remember. When printed on paper media, they are difficult to read and type into the browser. Even when shared via electronic means such as email and blog posts, long URLs are not elegant because they are often broken into multiple lines. The problem is exacerbated when the URL contains (URL-encoded) special characters, which may be accidentally modified or filtered out by sanitization code aiming to block cross-site scripting and injection attacks. Another motivation for URL shortening comes from services like Twitter that impose a 140-character limit on the messages users post online and from mobile SMS that are limited to 160 characters, making it impossible to share long URLs.

URL shortening services (URL shorteners) map long URLs to short ones. The first URL shorteners were patented in 2000 [29]. Hundreds of URL shorteners have emerged on the market since then [25]. Many services offer additional features such as page-view counting, analytics for tracking page visitors' OS, browser, location, and referrer page, URL-to-QR encoding, etc.

A URL shortener accepts a URL as input and generates a short URL. The service maintains an internal database mapping each short URL to its corresponding original URL so that any online access using a short URL can be resolved appropriately (see Figure 1).

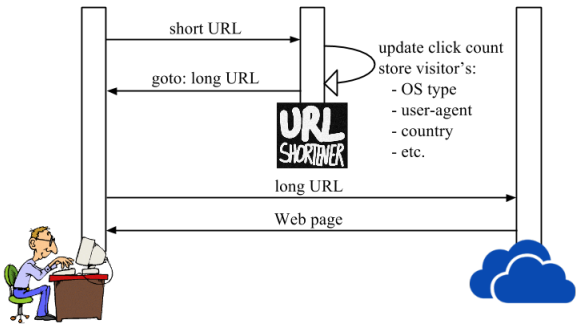

Figure 1: Resolving short URLs.

To generate short URLs, URL shorteners first define the alphabet (most commonly, [a-z,A-Z,0-9]) and the length of the output *token*. The token, sometimes referred to as the *key*, is the last part of the short URL, differentiating individual links in the shortener's internal database. For example, if the alphabet is [a-z,A-Z,0-9] and the token is 6 characters long, the shortener can generate $62^6 \approx 5.7 \cdot 10^{10}$ possible short URLs.

Short URLs can be generated sequentially, randomly, using a combination of the two (as in the case of `bit.ly` [31]), or by hashing the original URL. Sequential generation reveals the service's usage patterns and introduces concurrency issues.

`bit.ly` is a popular URL shortener. According to the counter on the front page of `bitly.com`, the company claims to have shortened over 26 billion URLs at the time of this writing. The tokens in `bit.ly` URLs are between 4 and 7 characters long, but currently the first character in 7-character tokens is almost always 1, thus the effective space of 7-character `bit.ly` URLs is 62^6 as described above. Therefore, the overall space of `bit.ly` URLs is $62^4 + 62^5 + 2 \cdot 62^6 \approx 1.2 \cdot 10^{11}$.

Some cloud services integrate URL shortening into their products to help users share links. For example, Microsoft OneDrive uses `1drv.ms` for this purpose. Reverse DNS lookup shows that `1drv.ms` is a branded short domain [10] operated by `bit.ly`. Therefore, OneDrive short URLs are in effect `bit.ly` short URLs. This fact has two implications: (1) `bit.ly` and `1drv.ms` share the same token space; (2) `1drv.ms` URLs can be resolved by the `bit.ly` resolver. Note that `bit.ly` URLs cannot be resolved using the `1drv.ms` resolver unless they point to OneDrive documents.

Other branded domains operated by `bit.ly` include `binged.it` for Bing Maps, `yhoo.it` for Yahoo! Maps, and `mapq.st` for MapQuest. All of them currently use 7-character tokens with the first character set to 1.

Google Maps uses the `goo.gl/maps` domain and, prior to the changes made in response to this paper (see Section 9), 5-character tokens. Thus, the entire token space of `goo.gl/maps` was $62^5 \approx 9.2 \cdot 10^8$.

2.2 Cloud Storage Services

Cloud storage services are gaining popularity because they enable users to access their files from anywhere and automatically synchronize files and folders between the user's devices and his or her cloud storage.

2.2.1 OneDrive

OneDrive is an online cloud storage service operated by Microsoft. The first 5 GB of storage are free; larger quotas are available for a small monthly fee.

OneDrive currently allows Word, Excel, PowerPoint, PDF, OneNote, and plain-text files to be viewed and edited through the service's Web interface. OneDrive also supports online viewing of many image and video file formats, such as JPEG, PNG, MPEG etc. Users may share OneDrive files and folders with view-only, edit, and public-access capabilities.

OneDrive provides client applications for Mac, PC, Android, iOS, Windows Phone, and Xbox to facilitate automatic file and folder synchronization between user's devices and his cloud storage account.

To facilitate application development and programmatic access to OneDrive accounts, Microsoft distributes two different, independent SDKs: Live SDK [2] and OneDrive pickers and savers SDK [33]. Live SDK is built using open standards like OAuth 2.0, REST, and JSON. It supports full-fledged access to files, folders, albums, photos, videos, audio files, tags, and comments. The lightweight OneDrive pickers and savers SDK supports limited functionality such as opening and storing OneDrive files and creating links to shared files.

2.2.2 Google Drive

Google Drive is Google's cloud storage product. New users get 15GB of storage for free; larger quotas, similar to OneDrive, are available for a small fee.

Google Drive has built-in support for Docs, Sheets, Slides, Forms, Drawings, and Maps. Users can thus view and edit popular file types like DOC, DOCX, PPT, PPTX, XLS, XLSX, etc. Users can also install applications from Google's Web Store that extend Google Drive's functionality to specialized file formats such as PhotoShop's PSD and AutoCAD's DWG.

Google Drive provides client applications for Mac, PC, Android, and iOS which automatically synchronize files and folders between the user's devices and his or her cloud storage account.

To facilitate programmatic access to files and folders stored on Google Drive, Google provides Google Drive SDKs [15] for Android, iOS, and the Web. Additionally, Google Drive API v.2 is available [14].

2.3 Online Mapping Services

Online maps are among the most popular and essential cloud-based services. MapQuest offered Web-based maps in 1996, followed by Yahoo! Maps in 2002, Google Maps in 2005, and Bing Maps in 2010. In addition to driving directions, modern online maps provide traffic details, road conditions, satellite, bird's-eye, and street views, 3D imagery of notable locations, etc.

All online mapping services let users share locations, as well as driving directions between two or more loca-

tions. The corresponding URLs are very long, thus mapping services directly integrate URL shorteners into their user interfaces, helping users share maps via text messages, social media, and email.

Mapping services provide APIs and SDKs to application developers. Google Maps [21] distributes SDKs for Android, iOS, and the Web. Bing Maps provides an SDK for Windows Store apps [8] and AJAX and REST APIs for Web and mobile [7]. There is also an unofficial, community-supported Bing Maps Android SDK [6]. MapQuest supports Web Services, JavaScript, and Flash APIs [28]. Yahoo! discontinued their Yahoo! Maps Web Services in 2011, but previously they had provided Flash, AJAX, and Map Image APIs [40].

3 Scanning Short URLs

Scanning rates. `bit.ly` provides an API [9] for querying its database. Access to this API is currently rate-limited to five concurrent connections from a single client, with additional "per-month, per-hour, per-minute, per-user, and per-IP rate limits for each API method" [34]. The limits are not publicly disclosed. When a limit is reached, the API method stops processing further requests from the client and replies with HTTP status code 403. In our experiments, a simple, unoptimized client can query the `bit.ly` database at a sustained rate of 2.6 queries/second over long periods of time. Further optimizations may push the effective query rate closer to the stated 5 queries/second rate limit and sustain it over a long time. We also observed that much higher rates, up to 227 queries/second, are possible for brief periods before the client's IP address is temporarily blocked by `bit.ly`.

`goo.gl/maps` also provides an API [18] for querying its database. The free usage quota is 1,000,000 queries per day [19]. At the time of our experiments, there was also an option to request a higher quota.

Sampling. To generate random tokens for the 6-character and 7-character token space of `bit.ly` and the 5-character token space of `goo.gl/maps`, we first defined the alphabet: [a-z,A-Z,0-9]. We then calculated the maximum number that a token can represent when interpreted as a Java BigInteger [5] and generated a random number within this space, interpreting it as a token. Random tokens in our samples were generated without replacement. The process of token generation ran until the desired number of unique random tokens was obtained for each target service (e.g., `bit.ly`) and target token space (e.g., 6-character token space.)

To sample the space of `bit.ly` URLs, we generated 100,000,000 random 6-character tokens and queried `bit.ly` from 189 machines. Our sample constitutes 0.176% of the 6-character token space. We found 42,229,055 URL mappings. Since the query tokens were chosen randomly, this implies that the space of 6-character `bit.ly` URLs has approximately 42% density. Because not all characters in `bit.ly` URLs appear to be random [31], there exist areas of higher density that would yield valid URLs at an even higher rate.

We also randomly sampled the 7-character token space on `bit.ly`. At the time of our experiments, `bit.ly` set the first character in all[4] 7-character tokens to 1. Thus, in practice, the search space of 7-character tokens has the same size as the space of 6-character tokens. Similarly to the 6-character scan, we generated 100,000,000 random tokens by setting the first character to 1 and appending a randomly generated 6-character token. The resulting sample constituted 0.176% of the 7-character token space and produced 29,331,099 URL mappings. Thus, the space of 7-character `bit.ly` URLs has approximately 29% density.

A careful reader will notice that if our density estimates are correct, `bit.ly` must have shortened more than $0.42 \cdot 62^6 + 0.29 \cdot 62^6 \approx 40$ billion URLs. Yet, the counter on the front page of `bitly.com` says that they shortened 26 billion URLs. We conjecture that this discrepancy is due to some URLs (e.g., those under branded domains) not being counted towards the reported total.

`goo.gl/maps` has a much smaller token space: $9.2 \cdot 10^8$ vs. $1.2 \cdot 10^{11}$. Prior to changes made by Google in response to our report (see Section 9), we scanned 63,970,000 tokens $\approx 7\%$ of the entire token space. Our scan produced 23,965,718 URL mappings, implying that the density on `goo.gl/maps` is 37.5%.

Exhaustive enumeration. At the current effective rate of querying `bit.ly`, enumerating the entire `bit.ly` database would take approximately 12.2 million compute hours, roughly equivalent to 510,000 client-days. Amazon EC2 Spot Instances [36] may be a cost-effective resource for automated URL scanning. Spot Instances allow bidding on spare Amazon EC2 instances, but without guaranteed timeslots. The lack of reserved timeslots matters little for scanning tasks. At the time we were conducting our scanning experiments, Amazon EC2 Spot Instances cost $0.003 per hour [37], thus scanning the entire `bit.ly` URL space would have cost approximately $36,700. This price will drop in the future as computing resources are constantly becoming cheaper. Moreover, Amazon AWS offers a free tier [3] service to new users with 750 free micro-instance hours of Linux plus 750 micro-instance hours of Windows per month for 12 months. Therefore, a stealthy attacker who is able to reg-

[4] With a few hard linked exceptions like `http://bit.ly/BUBVDAY`

ister hundreds of new AWS accounts can enumerate the entire bit.ly database for free.

Prior to changes described in Section 9, enumerating the entire goo.gl/maps database would have required 916 client-days. Google Cloud Platform offers a $300 credit [20] to be used over 60 days. Therefore, a stealthy attacker capable of registering a few hundred Google accounts could have enumerated the entire goo.gl/maps database for free in a matter of hours.

4 Short URLs in Cloud Storage Services

Cloud storage services create a unique URL for each file and folder stored in the user's account. These URLs allow users to view and edit individual files via the Web interface, change the metadata associated with files and folders, and share files and folders with other users.

Sharing actual URLs is often inconvenient: email agents may wrap long URLs, rendering them unclickable, text messages and Twitter have a limit on message size, etc. URL shortening helps users share URLs over email, text or instant messages, and social media.

4.1 Microsoft OneDrive

The experiments in this section used short-URL scanning to discover publicly accessible OneDrive files and folders. Our scanner accessed only public URLs and did not circumvent any access-control protections. Information was collected solely for measurement purposes.

Our scanner considered only the *metadata*, such as files and directory names. We did not analyze the contents of OneDrive files found by scanning because they may contain sensitive personal data. Note that these contents remain exposed through public URLs and are thus vulnerable to a less scrupulous adversary.

4.1.1 Discovering OneDrive Accounts

Of the 42,229,055 URLs we discovered from the 6-character token space of bit.ly, 3,003 URLs (0.003% of the sample space) reference files or folders under the onedrive.live.com domain. Additionally, 16,521 URLs (0.016% of the sample space) reference files or folders under the skydrive.live.com domain. If this density holds over the entire space, the full scan would produce $62^6 \cdot 0.003\% \approx 1,700,000$ (respectively, $62^6 \cdot 0.016\% \approx 9,000,000$) URLs pointing to OneDrive (respectively, SkyDrive) documents. In our sample scan, each client found, on average, 43 OneDrive/SkyDrive URLs per day. At this rate, it would take approximately 245,000 client-days to enumerate all OneDrive/SkyDrive URLs mapped to 6-character tokens. A botnet can easily achieve this goal in a single day or even much faster if the operator is willing to have bots' IP addresses blocked by bit.ly.

Of the 29,331,099 URLs we discovered from the 7-character token space of bit.ly, 25,594 (0.025% of the sample space) point to OneDrive files or folders, and 21,487 (0.021% of the sample space) point to SkyDrive files or folders. Thus, the projected URL counts of OneDrive/SkyDrive links in the 7-character token space of bit.ly are $62^6 \cdot 0.025\% \approx 14,200,000$, and $62^6 \cdot 0.021\% \approx 11,900,000$, respectively.

For each OneDrive/SkyDrive URL found by our sample scan, the scanner issued a GET request. If the landing page did not redirect to a page outside the user's account, we considered the link "live." The number of live links is generally greater than the number of OneDrive/SkyDrive accounts because different links may lead to different files in the same account.

Of the 3,003 OneDrive URLs (respectively, 16,521 SkyDrive URLs) sampled from the 6-character token space, 2,130 (respectively, 9,694) were live. Of the 25,594 OneDrive URLs (respectively, 21,487 SkyDrive URLs) sampled from the 7-character token space, 22,069 (respectively, 13,472) were live.

All URLs in our sample lead to distinct OneDrive accounts. Due to the small sample size, we cannot draw any conclusions about the total number of OneDrive accounts that would be discovered by a full scan.

4.1.2 Traversing OneDrive Accounts

OneDrive supports all URL formats shown in Table 1. Each account is uniquely identified by the value of the *cid* parameter. The *id* and *resid* parameters have the "*cid!sequence_number*" format. Thus, given *id* or *resid*, it is trivial to recover *cid*, but given *cid*, there is no easy way to construct a valid *id* or *resid*. However, these sequence numbers can be brute-forced. Possible values for the *app* parameter are *Word*, *Excel*, *PowerPoint*, *OneNote*, and *WordPdf*. We observed only the value of *3* for the *v* parameter. The *ithint* parameter denotes a folder and encodes the type of content therein, such as JPEG PNG, or PDF. The *authkey* parameter is a capability key that grants access rights (view-only, edit, etc.)

It is not necessary to guess URL parameter values to gain access to OneDrive files. Having obtained the URL of a single document, one can exploit the predictable structure of OneDrive URLs to traverse the account's directory tree and enumerate other shared files and folders. The account traversal methodology described in the rest of this section worked reliably between October 2014 and February 2016. As of March 2016, direct access to the account's root URL (see below) no longer reveals the URLs of files and folders shared under the same capability in that account.

File type	prefix	path	cid	id	resid	app	v	ithint	authkey
Word	https://onedrive.live.com	/view*	✓	✗	✓	✓	✗	✗	optional
Excel		/view or /edit	✓	✗	✓	✓	✗	✗	
PowerPoint		/view*	✓	✗	✓	✓	✗	✗	
OneNote		/view or /edit	✓	✗	✓	✓	✗	✗	
PDF		/view	✓	✗	✓	✓	✗	✗	
Surveys		/survey	✗	✗	✓	✗	✗	✗	
Media files		/	✓	✓	✗	✗	✓	✗	
Downloads		/download.aspx	✓	✗	✓	✗	✗	✗	
Folders		/ +	✓	✓	✗	✗	✗	✓	

Table 1: OneDrive URL formats.
* Word and PowerPoint files shared with "edit" capability can be edited online, despite the absence of "/edit" path.
+ Folders shared with "edit" allow anyone to write into them.

Suppose a scan found a short URL such as `http://1drv.ms/1xNOWV7` which resolves to `https://onedrive.live.com/?cid=485bef1a805 39148&id=485BEF1A80539148!115&ithint=folder,xlsx&authkey=!AOOp2TqTTSMT5q4`. Parse this URL and extract the *cid* and *authkey* parameters, then construct the root URL for the account by replacing *XXX* and *YYY* in `https://onedrive.live.com/?cid=XXX&authkey=YYY` with the *cid* and *authkey* values.

Prior to March 2016, access to the root URL made it easy to automatically discover URLs of shared files and folders in the account. For example, to find URLs of individual files, parse the HTML code of the page and look for "*a*" elements with "*href*" attributes containing "*&app=*", "*&v=*", "*/download.aspx?*", or "*/survey?*". Such links point to individual documents. Links that start with `https://onedrive.live.com/` and contain the account's *cid* may lead to other folders.

Starting from each of the 2,130 OneDrive URLs discovered by our sample scan of the 6-character token space of `bit.ly` and navigating through the directory trees of the corresponding OneDrive accounts, we found a total of 227,276 publicly accessible files. Similarly, navigating from each of the 22,069 OneDrive URLs discovered in the 7-character token space yielded a total of 1,105,146 publicly accessible files (see Table 2).

Figure 2 shows the distribution of files per account in our sample of the 6-character token space. The average number of files per account is 106, the maximum is 23,240, the minimum is 0 (i.e., an empty folder). The distribution of files per account in the 7-character token space is shown in Figure 3. The average is 50, the maximum is 30,779, the minimum is 0.

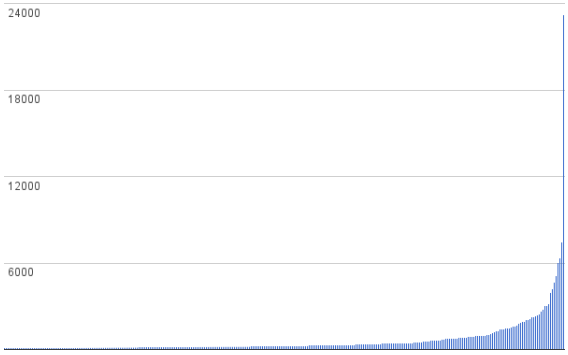

Figure 2: Distribution of files per OneDrive account discovered by scanning the 6-character token space of bit.ly

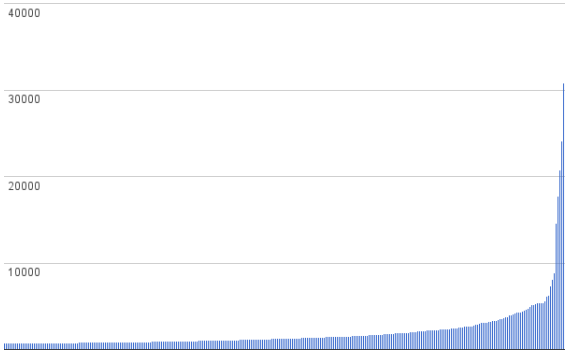

Figure 3: Distribution of files per OneDrive account discovered by scanning the 7-character token space of bit.ly

4.1.3 Exploiting Unlocked OneDrive Folders

Among the 2,130 live OneDrive accounts discovered in our sample of the 6-character token space of `bit.ly`, 150 have at least one folder shared with edit functionality. To find such accounts, our scanner searched the HTML code of the page for a "*span*" element with "*class*" attribute equal to "*navLinkText*" and text attribute equal to "*Upload*". Of the 22,069 OneDrive accounts found in our sample of the 7-character token space, 1,561 have at least one folder shared with edit functionality. We estimate that approximately **7%** of discoverable OneDrive accounts have world-writable folders.

File type	# of files found in 6-char sample space	# of files found in 7-char sample space
Word	2,116	21,077
Excel	921	6,050
PowerPoint	688	5,068
OneNote	51	6
PDF	10,080	41,465
Surveys	22	226
Media files	204,735	862,641
Downloads*	8,663	168,613

Table 2: Publicly accessible files on OneDrive discovered by sampling from the 6- and 7-character token space of bit.ly.
* "Downloads" refers to file types not natively supported for viewing or editing via the OneDrive Web interface.

We call these folders "unlocked" because *anyone* who knows their URL—which, as we demonstrated, is easily discoverable—can use the edit feature to overwrite existing files and/or add new files, potentially planting malware into users' OneDrive accounts. Microsoft appears to perform some rudimentary anti-virus scanning on OneDrive accounts, but it is trivial to evade. For example, this scanning fails to discover even the test EICAR virus[5] compressed in the `.xz` format.

Automatic synchronization between the OneDrive cloud and users' personal machines and devices, which is normally a very convenient feature, turns this vulnerability into a major security hole. For example, if the attacker infects a user's existing file (e.g., inserts a macro virus into a Word or Excel file), all of the victim's devices linked to his or her OneDrive account will automatically download the infected file. When the victim opens the file, the malware will execute with the victim's privileges on the machine where the file was opened.

The attacker can also add new files to unlocked folders, for example, executable malware files with names designed to trick the user into clicking on them when these files automatically appear on the user's personal computer or device. This attack vector can also be leveraged to exploit any number of known bugs in parsers and renderers of common file formats such as JPEG [23], PDF [32], and DOCX [13].

4.2 Google Drive

Unlike OneDrive, Google Drive does not directly integrate a URL shortener, thus users need to manually invoke a shortener if they want to generate a short URL.

Our sample scan of 6-character `bit.ly` tokens yielded 44 links to Google Drive folders: 30 are view-only, 3 are writable, 7 have already been taken down, and 4 are permission-protected. Our sample of 7-character tokens yielded 414 links to Google Drive folders: 277 are view-only, 40 are writable, 49 have been taken down, and 48 are permission-protected. As with OneDrive, anyone who discovers the URL of a writable Google Drive folder can upload arbitrary content into it, which will be automatically synced with the user's devices.

Unlike OneDrive, Google Drive allows access to documents that were removed from the folders but not permanently deleted from the trash.

[5]The EICAR Standard Anti-Virus Test file is a special 'dummy' file used to check and confirm the correct operation of security products.

5 Short URLs in Mapping Services

In this section, we first show that by scanning short URLs, one can discover driving directions shared by users of online mapping services. We then explain how these directions compromise users' privacy by revealing sensitive locations they visited, their social ties, etc.

Our analysis focuses on Google Maps because the token space of `goo.gl/maps` URLs was so small prior to the changes described in Section 9. We believe that similar results can be obtained for any mapping service that integrates URL shorteners, including MapQuest, Bing Maps, and Yahoo! Maps.

Google Maps. Of the 23,965,718 URLs in our Google Maps sample, 2,357,844 or about 9.8% are for directions; the rest are for individual locations.

Our sample includes directions to and from many sensitive locations: clinics for specific diseases (including cancer and mental illnesses), addiction treatment centers, abortion providers, correctional and juvenile-detention facilities, payday and car-title lenders, gentlemen's clubs, etc. In particular, the sample contains 3,913 map directions that start at a hospital and end at a residential address, and 12,668 directions that start at a residential address and end at a hospital. Figure 6 shows an example. We could have constructed similar

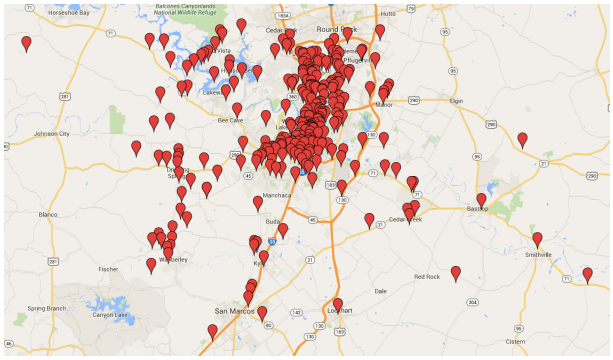

Figure 4: Locations associated with a single user in Austin, TX.

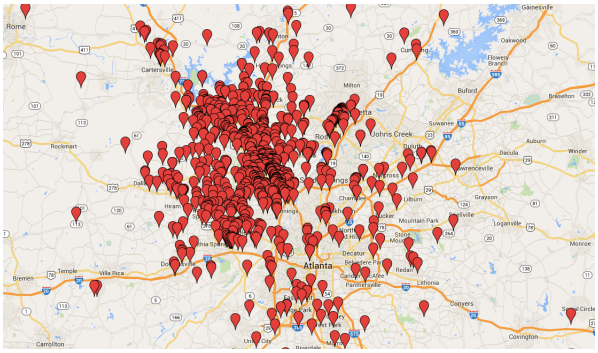

Figure 5: Locations associated with D & D Autows Inc.

maps for any sensitive location. More importantly, *anyone* could have constructed them simply by scanning all `goo.gl/maps` URLs.

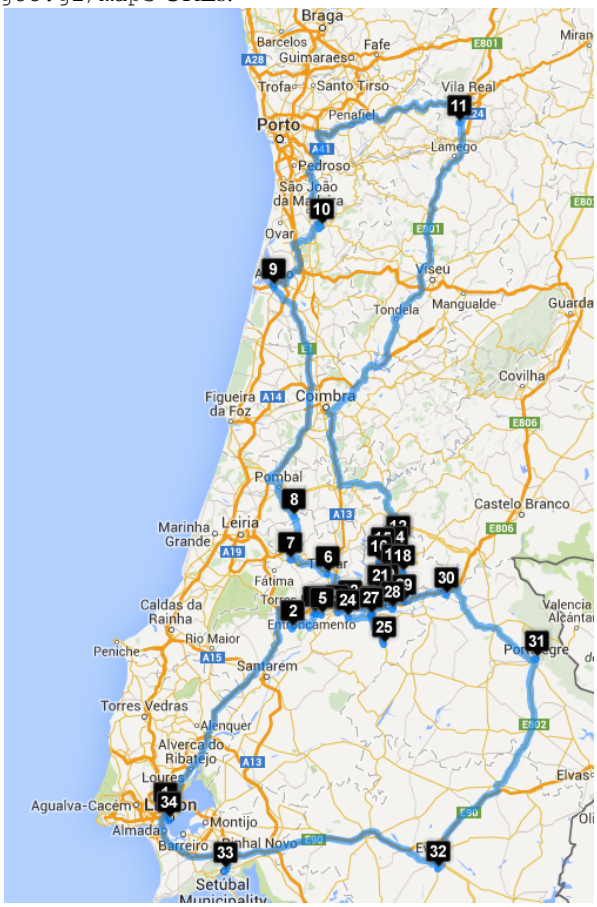

Figure 6: Residential addresses associated with Hospital Doutor Manoel Constâncio.

The endpoints of driving directions shared via short URLs often contain enough information to uniquely identify the individuals who requested the directions. For instance, when analyzing one such endpoint, we uncovered the address, full name, and age of a young woman who shared directions to a planned parenthood facility.

Conversely, by starting from a residential address and mapping all addresses appearing as the endpoints of the directions to and from the initial address, one can create a map of who visited whom.

Fine-grained data associated with individual residential addresses can be used to infer interesting information about the residents. For instance, we conjecture that one of the most frequently occurring residential addresses in our sample (see Figure 4) is the residence of a geocaching enthusiast. He or she shared directions to hundreds of locations around Austin, TX, many of them specified as GPS coordinates. We have been able to find some of these coordinates in a geocaching database [16].

Similarly, we can take a business (e.g., D & D Autows Inc.) and extract all map directions created to or from its location—see Figure 5. If a person had their vehicle towed to or from their house, then their identity can be easily inferred by cross-correlating the addresses with public directories such as White Pages.

Additionally, Google API for short URLs reveals the exact time when the URL was created, as well as the approximate time of recent URL visits [39]. This information can be used to create fine-grained activity profiles for users who share Google Maps directions.

MapQuest. MapQuest uses the branded `mapq.st` domain operated by `bit.ly`. Our 6-character sample contains 151,334 short MapQuest URLs: 6,737 for directions and 144,597 for point locations. Our 7-character sample contains 141,721 URLs: 66,929 for directions and 74,792 for point locations. All new short MapQuest URLs use 7-character `bit.ly` tokens.

Bing Maps. Bing Maps uses the branded `binged.it` domain operated by `bit.ly`. Our 6-character sample contains 28,271 Bing Maps URLs: 10,020 for directions and 18,251 for point locations. Our 7-character sample contains 34,363 URLs: 13,116 for directions and 21,247 for point locations. All new short Bing Maps URLs use 7-character `bit.ly` tokens.

Yahoo! Maps. Yahoo! Maps uses the branded `yhoo.it` domain operated by `bit.ly`. Our 6-character sample contains 1,331 Yahoo! Maps URLs: 771 for directions and 560 for point locations. Our 7-character sample contains 8,400 URLs: 6,207 for directions and 2,193 for point locations. All new short Yahoo! Maps URLs use 7-character `bit.ly` tokens.

6 Mitigation

We suggest five approaches to mitigate the vulnerabilities identified in this paper: (1) make short URLs longer, (2) inform users about the risks of URL shorteners, (3) do not rely on universal URL shorteners, (4) employ CAPTCHAs or other methods to separate human users from automated scanners, and (5) design better APIs for the cloud services that use short URLs.

First, URL shorteners should use longer URLs. There is an obvious tension between maintaining the benefits of short URLs and preventing scanning attacks. It might be instructive to compare the size of token space to the "bits of security" metric sometimes used in cryptography. Most token alphabets consist of 62 characters, which is close to 2^6. Each character can thus be thought of as providing roughly 6 bits of search space. We estimate that tokens of 10 characters or more would make it difficult to scan the entire token space. From a usability perspective, 10-character tokens will slightly increase the difficulty of hand-copying short URLs and also make them less "friendly-looking." From an attack mitigation perspective, longer tokens would be highly effective. For example, at the current density of 42% in the 6-character token space of `bit.ly`, the attacker needs about 2 queries to obtain a single valid URL. Should the token size be increased to 10 characters, the attacker would have to send 35 million queries to obtain a single valid URL. Future decisions on the size of token space should involve careful analysis of attackers' capabilities.

Second, URL shorteners should warn users that creating a short URL may expose the content behind the URL to third parties. For integrated applications, the warnings can be more specific and tailored to the application (e.g., maps). This approach has limitations. A typical user may not be able to properly assess whether using a shortener is dangerous. Furthermore, the person who is asking for a URL to be shortened could be different from the person who is impacted by its disclosure. For example, a towing company may not care about disclosing the residences it serviced even if the individuals being serviced do.

Third, cloud services should consider using internal, company-owned URL shorteners, as opposed to universal shorteners such as `bit.ly`. This change would enable companies to (1) significantly decrease the density of the token space, (2) closely monitor automated scans of the short-URL space, and (3) take appropriate actions as soon as a scan is detected. Furthermore, it will increase the burden on the attackers since they will need to scan different token spaces for different services.

Fourth, URL shorteners must take a more aggressive approach against scanning. Instead of fixed monthly/daily/hourly limits, they should identify large-scale scanners and block their IP addresses. Alternatively, they could ask users to solve CAPTCHAs every few hundred requests to verify that the requester is human.

Fifth, cloud services that use URL shorteners need better API design. Lengthening short URLs does not prevent an attacker who discovers a short URL to a single file from enumerating all files and folders shared under the same capability key. In particular, Microsoft OneDrive should change the format and structure of long URLs so that, given the URL of one document, it is no longer possible to discover the URLs of other documents in the same account (Microsoft appears to have made this change some time between February and March of 2016). A similar approach is already taken by Google Drive when individual files are shared.

7 Related Work

Antoniades et al. [1] explored the popularity, temporal activity, and performance impact of short URLs.

Neumann et al. [31] studied the security and privacy implications of URL shorteners on Twitter, focusing on the use of short URLs for spam and user tracking, as well as leakage of private information via URL-encoded parameters and HTTP referer headers. As part of their analysis, they discovered and manually examined 71 documents hosted on Google Docs.

In contrast to Neumann et al., we demonstrate that it is practically feasible to automatically discover a large number of cloud-stored files by randomly scanning short URLs. We show how automated traversal of OneDrive accounts reveals even files and folders that do not have short URLs associated with them. We also identify large-scale malware injection as a serious security risk for cloud storage accounts discovered via short-URL scanning. Unlike Neumann et al., we did not analyze the content of private documents we found due to ethical and legal considerations and the logistical difficulties of requesting permission from the affected users.

Klien and Strohmaier [24] investigated the use of short URLs for fraud, deceit, and spam. Based on the logs of the URL shortener their group operated over several months, they concluded that about 80% of the URLs they shortened are spam-related. This analysis does not apply to short URLs integrated into cloud services.

Maggi et al. [27] built a service for crowdsourced users to preview the landing pages of short URLs resolved by

622 URL shorteners and found that short URLs are often used to hide the true URLs of drive-by download and phishing pages. They also explored the countermeasures that can be deployed by URL shorteners. They did not discover the problem of cloud storage accounts or mapping directions exposed by short URLs.

There is a rich literature on inferring information about individuals from location data. Becker et al. [4] used anonymized call detail records from a large US communications service provider to identify large groups of people who collectively share the same usage patterns. Crandall et al. [12] inferred social ties between people based on their co-occurrence in a geographic location. Isaacman et al. [22] inferred important places in people's lives from location traces. Montjoye et al. [30] observed that 95% of individuals can be uniquely identified given only 4 points in a high-resolution dataset such as a cell phone carrier's service records. Golle and Partridge [17] showed the feasibility of re-identifying anonymized location traces; futility of anonymizing location traces was also demonstrated in [26, 41].

Between 2013 and 2015, information about many Uber rides, including customers' exact addresses, was accidentally made public after Google indexed "share your ETA" links posted by Uber's customers [11]. Uber fixed the problem by expiring the links after 48 hours.

To the best of our knowledge, this paper is the first to observe that the sharing of maps between users can lead to significant privacy violations because short URLs integrated into popular mapping services effectively make all shared locations and directions public.

8 Conclusions

URL shortening, which looks like a relatively minor feature, turns out to have serious consequences for the security and privacy of modern cloud services. In this paper, we demonstrate that the token space of short URLs is so small as to be efficiently enumerable and scannable. Therefore, any short link to an online document or map shared by a user of a cloud service is effectively public.

In the case of cloud-storage services such as Microsoft OneDrive, this not only leads to leakage of sensitive documents, but also enables anyone to inject arbitrary malicious content into unlocked accounts, which is then automatically copied into all of the account owner's devices.

In the case of mapping services, short URLs reveal addresses and—via easy cross-correlation with public directories—identities of users who shared directions to medical facilities (including abortion, mental-health, and addiction-treatment clinics), prisons and juvenile detention centers, places of worship, and other sensitive locations; enable inference of social ties between people; and leak other sensitive private information.

Solving the problem identified in this paper will not be easy since short URLs are an integral part of many cloud services and previously shared information remains publicly accessible (unless URL shorteners take the drastic step of revoking all previously issued short URLs). We present several recommendations which could mitigate the damage caused by short URLs.

9 Disclosure

Microsoft. We notified Microsoft about the security and privacy risks of short OneDrive URLs on May 28, 2015. In particular, any user who shares a short OneDrive URL with a collaborator may unintentionally expose the shared files and folders to *everyone*. Furthermore, if the shared documents and folders allow writing, anyone can inject malicious content into them that will be automatically downloaded to the user's computers and devices.

After an email exchange involving several messages, "Brian" from Microsoft's Security Response Center (MSRC) informed us on August 1, 2015, that the ability to share documents via short URLs "appears by design," and thus "does not currently warrant an MSRC case."

In March of 2016, Microsoft removed the "shorten link" option from OneDrive, causing a number of user complaints.[6] We asked MSRC whether this change was made in response to our previous report. MSRC informed us that our analysis played no role in their decision to remove this option and reiterated that they do not consider our report a security vulnerability. At approximately the same time, Microsoft changed the API so that the account traversal methodology described in Section 4.1.2 no longer appears to work.

As of this writing, all previously generated short OneDrive URLs remain vulnerable to scanning and malware injection.

Google. We notified Google about the privacy risks of short Google Maps URLs on September 15, 2015. Google promptly responded to our report. As of September 21, 2015, newly created short URLs to Google Maps have 11 or 12-character tokens and are thus not vulnerable to brute-force scanning.

[6] http://answers.microsoft.com/en-us/onedrive/forum/odwork-odshare/shorten-link-option-no-longer-available-on/bc3dc4eb-cb54-43e0-bcff-a072e8dba3ad?auth=1

References

[1] ANTONIADES, D., POLAKIS, I., KONTAXIS, G., ATHANASOPOULOS, E., IOANNIDIS, S., MARKATOS, E., AND KARAGIANNIS, T. we.b: The Web of Short URLs. In *WWW* (2011).

[2] MSDN: The Live SDK. https://msdn.microsoft.com/en-us/library/dn631819.aspx, 2015.

[3] AWS Free Tier. http://aws.amazon.com/free/, 2015.

[4] BECKER, R., CÁCERES, R., HANSON, K., LOH, J., URBANEK, S., VARSHAVSKY, A., AND VOLINSKY, C. Clustering Anonymized Mobile Call Detail Records to Find Usage Groups. In *In Proceedings of the 1st Workshop on Pervasive Urban Applications* (2011).

[5] java.math.BigInteger. http://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html, 2015.

[6] Bing Maps Android SDK. http://bingmapsandroidsdk.codeplex.com, 2015.

[7] Bing Maps API. http://www.microsoft.com/maps/choose-your-bing-maps-API.aspx, 2015.

[8] Bing Maps SDK for Windows Store apps. https://visualstudiogallery.msdn.microsoft.com/ebc98390-5320-4088-a2b5-8f276e4530f9, 2015.

[9] Bit.ly: API Documentation and Resources. http://dev.bitly.com, 2015.

[10] How Do I Set Up a Branded Short Domain? http://support.bitly.com/knowledgebase/articles/76741-how-do-i-set-up-a-branded-short-domain, 2015.

[11] CARSON, B. Uber fixes link problem after trip information appears in Google search results. http://www.businessinsider.com/uber-fixes-flaw-with-trip-info-appearing-in-google-2015-9, 2005.

[12] CRANDALL, D., BACKSTROM, L., COSLEY, D., SURI, S., HUTTENLOCHER, D., AND KLEINBERG, J. Inferring Social Ties from Geographic Coincidences. In *Proceedings of the National Academy of Sciences, 107(52)* (2010).

[13] National Vulnerability Database. https://web.nvd.nist.gov/view/vuln/search-results?query=docx+microsoft, 2015.

[14] Google Developers: Drive API v2. http://developers.google.com/apis-explorer/?hl=en_US#p/drive/v2/, 2015.

[15] Google Developers: Google Drive SDK. https://developers.google.com/drive, 2015.

[16] Geocaching. https://www.geocaching.com/play, 2015.

[17] GOLLE, P., AND PARTRIDGE, K. On the Anonymity of Home/Work Location Pairs. In *Pervasive* (2009).

[18] Google Developers: URL Shortener API. https://developers.google.com/url-shortener/, 2015.

[19] Google Developers: URL Shortener: Getting Started. https://developers.google.com/url-shortener/v1/getting_started#quota, 2016.

[20] Google Cloud Platform: Start Your 60 Day Free Trial. https://cloud.google.com/free-trial/, 2015.

[21] Google Maps API. https://developers.google.com/maps/, 2015.

[22] ISAACMAN, S., BECKER, R., CÁCERES, R., KOBOUROV, S., MARTONOSI, M., ROWLAND, J., AND VARSHAVSKY, A. Identifying Important Places in People's Lives from Cellular Network Data. In *Pervasive* (2011).

[23] National Vulnerability Database. https://web.nvd.nist.gov/view/vuln/search-results?query=jpeg+microsoft, 2015.

[24] KLIEN, F., AND STROHMAIER, M. Short Links Under Attack: Geographical Analysis of Spam in a URL Shortener Network. In *HT* (2012).

[25] URL Shortening Services. http://vanityurlshorteners.com, 2015.

[26] MA, C., YAU, D., YIP, N., AND RAO, N. Privacy Vulnerability of Published Anonymous Mobility Traces. In *MobiCom* (2010).

[27] MAGGI, F., FROSSI, A., ZANERO, S., STRINGHINI, G., STONE-GROSS, B., KRUEGEL, C., AND VIGNA, G. Two Years of Short URLs Internet Measurement: Security Threats and Countermeasures. In *WWW* (2013).

[28] MapQuest + Developer: Your Geospatial Toolkit. https://developer.mapquest.com/products/, 2016.

[29] MEGIDDO, N., AND MCCURLEY, K. Efficient retrieval of uniform resource locators. http://pimg-fpiw.uspto.gov/fdd/24/572/069/0.pdf, 2000.

[30] MONTJOYE, Y., HIDALGO, C., VERLEYSEN, M., AND BLONDEL, V. Unique in the Crowd: The Privacy Bounds of Human Mobility. *In Nature Scientific Reports 3, 1376* (2013).

[31] NEUMANN, A., BARNICKEL, J., AND MEYER, U. Security and Privacy Implications of URL Shortening Services. In *W2SP* (2011).

[32] National Vulnerability Database. https://web.nvd.nist.gov/view/vuln/search-results?query=pdf+microsoft, 2015.

[33] MSDN: OneDrive pickers and savers. https://msdn.microsoft.com/en-us/library/office/dn782252.aspx, 2015.

[34] Bit.ly: Rate Limiting. http://dev.bitly.com/rate_limiting.html, 2015.

[35] Hypertext Transfer Protocol – HTTP/1.1. https://www.ietf.org/rfc/rfc2616.txt, 1999.

[36] Amazon EC2 Spot Instances. http://aws.amazon.com/ec2/spot/, 2016.

[37] EC2 Spot Prices. http://ec2price.com/, 2015.

[38] What is the maximum length of a URL in different browsers? http://stackoverflow.com/questions/417142/what-is-the-maximum-length-of-a-url-in-different-browsers, 2012.

[39] Google Developers: URL Shortener API: Look up a short URL's analytics. https://developers.google.com/url-shortener/v1/getting_started#url_analytics, 2015.

[40] Yahoo Maps API. https://developer.yahoo.com/maps/simple/V1/, 2015.

[41] ZANG, H., AND BOLOT, J. Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study. In *MobiCom* (2011).