Automated Malicious Advertisement Detection using VirusTotal, URLVoid, and TrendMicro

Conference Paper · April 2017

DOI: 10.1109/IACS.2017.7921994

Monther Aldwairi
Zayed University

Automated Malicious Advertisement Detection using VirusTotal, URLVoid, and TrendMicro

Rima Masri* and Monther Aldwairi*†
*College of Technological Innovation
Zayed University, Abu Dhabi, UAE 144534
Email: {m80006800, Monther.Aldwairi}@zu.ac.ae
†College of Information Technology
Jordan University of Science and Technology, Irbid, Jordan
Email: munzer@just.edu.jo

Abstract—**The Internet economy is based on free access to content in exchange of viewing advertisements that might lead to online purchases. Advertisements represent an important source of revenue to Advertising companies. Those companies employ every possible technique and trick to maximize clicks and visits to advertisers' websites. Modern websites exchange advertisement contents from ads' providers (such as Google AdSense), which means they do not control the contents of those advertisements. Although large providers such as Google and Yahoo! are supposed to be trustworthy, ad arbitration allows them to auction of those ad slots to other providers. Therefore, web administrators cannot guarantee the source of the ads on their delegated website areas. Those advertisements contain Javascript and may redirect to malicious websites which might lead to malicious code being executed or malware being installed. This paper proposes and implements a system for automatically detecting malicious advertisements. It employs three different online malware domain detections systems (VirusTotal, URLVoid, and TrendMicro) for malicious advertisements detection purposes and reports the number of detected malicious advertisements using each system. In addition, we study the efficiency of each system by calculating the confusion matrix and accuracy. We find that URLVoid is the best in terms of accuracy (73%) because it uses a combination of well known website scanners and domain blacklists.**

Keywords*—*Malvertisements; malicious advertisements; malicious websites; VirusTotal

I. Introduction

Online advertising combines the traditional form of advertising with new technologies to promote its own communication strategies [1]. The industry of online advertising is growing dramatically. According to the IAB report, the revenues of online advertisement increased by 19% in the first half of 2015 compared to the first half of 2014 [2]. Online advertisements along with Ecommerce are the main sources of revenue for many tech giants such as Google, Yahoo!, and Facebook. In addition to those big players, cybercriminals are now advertisers, publishers and running ad exchanges. This resulted in the emergence of new attacks such as click fraud, clickbaiting, drive-by-download [3], and URL redirection [4]. Those attacks are delivered through online advertisements that are referred to as malicious advertisements or Malvertisements for short.

Previous researches were conducted to detect, analyze, and filter out malicious advertisements. Some papers used existing tools in the detection and analysis process without comparing between those tools [5], while others built their own detection systems [6] [7] [8] [9] [10] [11]. In this paper, we propose and build an automatic detection system that extracts advertisements, employs three malicious domain detection tools: VirusTotal, URLVoid, and TrendMicro to classify and detect malicious advertisements. Then we evaluated the efficiency of using each tool separately by calculating their confusion matrices.

This project aims to detect malicious online advertisements using different detection and analysis tools: VirusTotal, URLVoid and TrendMicro. It also aims to find out which tool is the most effective, by building a confusion matrix for each tool and calculating its accuracy. The rest of this paper is organized as follows. Section II presents real examples of Malvertisements and briefly explains the three detection systems used. Section III briefly discusses related work and their shorting comings while Section IV presents the methodology, data collection and three detection algorithms. Section V lays out the experimental setup and presents a thorough analysis of the results.

II. Background

In this section we present an example of real benign and malicious advertisements and explain the three systems employed in our detection algorithms.

A. Online Advertisement

Online advertisements are delivered through web-based infrastructure to achieve a number of objectives: informing, persuading, reminding, or building brand awareness or brand loyalty. Moreover, online advertisements are useful to increase sales, revenues, and profits [12]. The infrastructure of online advertisements consists of four main parties: publishers, advertisement networks, advertisers, and audiences. Publishers are those who display advertisements on their web pages to make money. Advertisement networks are entities that manage publishers and advertisers by buying and selling advertisements traffic. Advertisers are those who create advertisements and pay advertisements' networks to display their advertisements on publishers' websites [13]. Audiences are users who access web pages and watch or interact with advertisements [6].

Fig. 1 shows an example of an online advertisement. A typical online advertisement usually consists of three parts: image, destination URL, and ID. Fig. 2 shows the HTML source code of the advertisement in Fig. 1. We can find the three elements of the advertisement defined in the HTML code. In addition, we can see that the destination URL of the advertisement goes through ”Propeller Ads Media”, which is a major advertisement network [14]. This is an example of a benign advertisement.

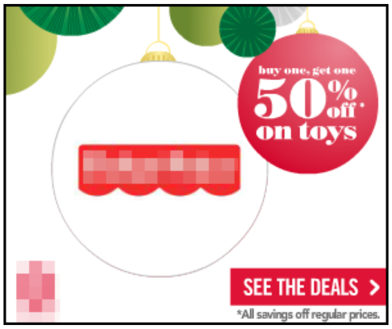
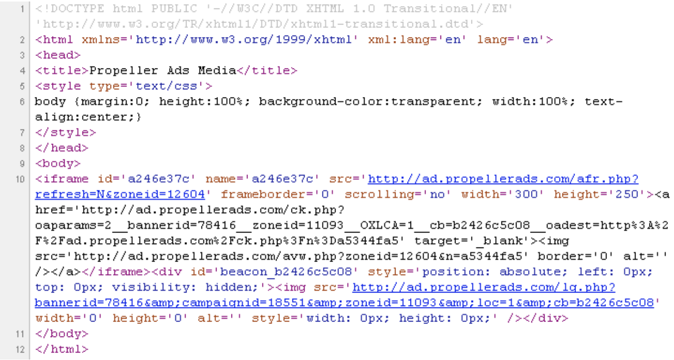


Fig. 1. Example of an Online Advertisement[14]



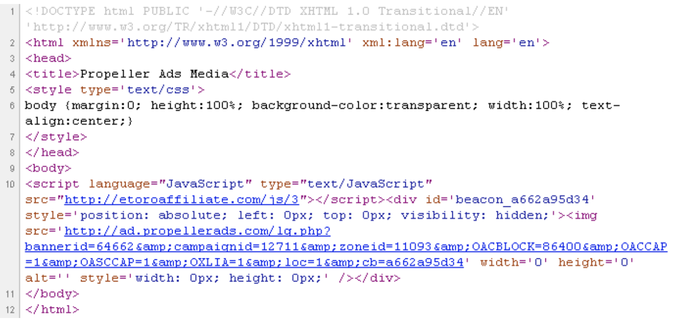
```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en' lang='en'>
<head>
<title>Propeller Ads Media</title>
<style type='text/css'>
body {margin:0; height:100%; background-color:transparent; width:100%; text-
align:center;}
</style>
</head>
<body>
<iframe id='a246e37c' name='a246e37c' src='http://ad.propellerads.com/afr.php?
refresh=N&zoneid=12604' frameborder='0' scrolling='no' width='300' height='250'><a
href='http://ad.propellerads.com/ck.php?
oaparams=2__bannerid=78416__zoneid=11093__OXLCA=1__cb=b2426c5c08__oadest=http%3A%2
F%2Fad.propellerads.com%2Fck.php%3Fn%3Da5344fa5' target='_blank'><img
src='http://ad.propellerads.com/avw.php?zoneid=12604&n=a5344fa5' border='0' alt=''
/></a></iframe><div id='beacon_b2426c5c08' style='position: absolute; left: 0px;
top: 0px; visibility: hidden;'><img src='http://ad.propellerads.com/lg.php?
bannerid=78416&amp;campaignid=18551&amp;zoneid=11093&amp;loc=1&amp;cb=b2426c5c08'
width='0' height='0' alt='' style='width: 0px; height: 0px;' /></div>
</body>
</html>
```

Fig. 2. Code of a Benign Online Advertisement [14]

Now let’s take a look into the HTML source code of the malicious advertisement shown by Fig. 3. Form this code, we can see that the destination URL is directly linked to ”etoroaffiliate.com”, which is a third party website, instead of going through ”Propeller Ads Media” or other major advertisement network. Therefore, ”Propeller Ads Media” is no longer in full control. Now let’s imagine that the URL is linked to a malicious website or tries to execute malicious JavaScript. This would simply install a malware or just result in unwanted behavior [14].

B. Detection System

There are plenty of detection systems that can take URLs, HTML code, and/or .EXE files to scan and detect malicious activities or malware. In our algorithm, we used three online detection systems described below to test URLs extracted from advertisements. These tools accept the redirection URLs of the online advertisements as input and analyze them. Based on the analysis results, our system classifies the advertisements as malicious or benign. The subsections below will discuss each system in full details.



```
<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>
<html xmlns='http://www.w3.org/1999/xhtml' xml:lang='en' lang='en'>
<head>
<title>Propeller Ads Media</title>
<style type='text/css'>
body {margin:0; height:100%; background-color:transparent; width:100%; text-
align:center;}
</style>
</head>
<body>
<script language="JavaScript" type="text/JavaScript"
src="http://etoroaffiliate.com/js/3"></script><div id='beacon_a662a95d34'
style='position: absolute; left: 0px; top: 0px; visibility: hidden;'><img
src='http://ad.propellerads.com/lg.php?
bannerid=64662&amp;campaignid=12711&amp;zoneid=11093&amp;OACBLOCK=86400&amp;OACCAP
=1&amp;OASCCAP=1&amp;OXLIA=1&amp;loc=1&amp;cb=a662a95d34' width='0' height='0'
alt='' style='width: 0px; height: 0px;' /></div>
</body>
</html>
```

Fig. 3. Code of a Suspicious Online Advertisement [14]

1) *VirusTotal:*

VirusTotal is a free tool that can be downloaded on desktop or accessed online to analyze suspicious files, hashes or URLs. It uses a number of Antivirus engines to facilitate the detection of different malwares including viruses, worms, and trojans [15]. In addition to Antivirus engines, VirusTotal uses website scanners to analyze and detect any malicious content available in URLs or files. It uses 58 Antivirus products such as Kaspersky Lab, Doctor Web, AVG Technologies, Cyren, up to 62 Website/domain [16] scanning engines and datasets such as AutoShun, CRDF, Sucuri SiteCheck, Quttera, and up to 18 file characterization tools and datasets such as ExifTool, Snort [17] and Wireshark.

2) *URLVoid:*

URLVoid is a free tool that can be accessed online to scan and analyze suspicious URLs. It uses a number of online security services to check the reputation of the submitted URL [18]. In addition to website reputation engines, URLVoid uses a number of domain blacklists to detect any malicious content available in URLs. URLVoid is developed by NoVirusThanks company, which is a dedicated for developing services and software in the field of computer and Internet security. URLVoid uses up to 42 website reputation engines and domain blacklists such as MyWOT, MalwareDomainList, GoogleSafeBrowsing, CRDF, Fortinet, etc. URLVoid has been given excellent reviews by a number of credible websites like PCWorld and IISoftware [19]. Besides the classification result, URLVoid provides details about the scanned website such as IP Address, IP Host, Country where the site servers are located and if the site is benign or not [18].

3) *TrendMicro:*

TrendMicro Antivirus company offers a domain-reputation database called Site Safety Center. It was developed as online free service to scan and analyze suspicious URLs. The service classifies the submitted URL as malicious or benign based on a number of factors such as website’s age, historical location changes, and indications of suspicious activities discovered through malware behavior analysis [20].

III. Related Work

This section will briefly discuss the notable and recent work in the area of malicious advertisements detection. Zarras et al. [5] studied the safety of advertisements on the web. This paper examined the ecosystem of advertisements in different aspects as well as the sources and reasons behind malicious advertisements. The paper started by crawling web pages and collecting the whole contents of 673,596 advertisements using Selenium software-testing framework. Then, it classified the collected advertisements to malicious or benign using three components: Wepawet, malware and phishing blacklists, and VirusTotal. After that, it analyzed the malicious advertisements that were discovered by examining any redirection or cloaking behaviors that are similar to known malicious behaviors. The paper found that only 1% of all the collected advertisements showed malicious behavior. Moreover, it found that bigger ad networks tend to perform a more accurate filtering of the advertisements than the smaller networks.

Cova et al. [8] implemented JavaScript Anomaly-based aNalysis and Detection (JSAND) tool for detecting and analyzing malicious web pages. The tool used ten features to characterize substantial events of a drive-by-download. These features were selected based on the steps that usually followed in carrying out an attack. In addition, the features were classified into necessary features that are required for a successful exploit such as the number of instantiated components (plugins) and useful features that are not strictly required to launch a successful attack such as the number of redirections to different URLs. In order for the system to decide whether a given feature value occurred, it assigned a probability score to each feature value based on several models such as Token Finder and Character Distribution. The tool was able to detect previously unseen drive-by download attacks on over 140,000 web pages. Finally, the proposed tool was evaluated against different datasets: known-good, known-bad, and uncategorized datasets. The found that JSAND achieved a good detection rate compared with other existing tools.

Li et al. [6] analyzed the collected advertisements crawled from 90,000 websites in a period of three months. Moreover, they recorded all network requests, responses, browser events, and the code retrieved. Based on the collected dataset, they analyzed malicious advertisements activities and their infrastructure features and came up with a system called MadTracer. MadTracer is a malvertising detection system built by combining the knowledge of malicious advertising nodes features and their related content delivery paths. It automatically generates detection rules and utilizes them to inspect advertisement delivery processes and detect malvertising activities. Finally, they found that MadTracer achieved a good detection rate in which it was able to detect new attacks that were not detected by other tools.

Xing et al. [9] studied the possibility of facilitating Malvertisements deployment through browser extensions. They developed Expector, which is a measurement framework used to identify browser extensions that inject ads in web pages. Expector was designed based on a number of things such as identifying the websites that may trigger the ad injection functionalities and triggering events that extension might be interested in for ad injection. They run Expector on almost 18,000 Google Chrome extensions and identified 292 extensions that perform ad injection. They found that 56 extensions lead to malware sites. Finally, the paper concluded that using extensions with ad injecting property might lead to Malvertising threats.

Ford et al. [11] designed and implemented a tool called OdoSwiff to analyze Flash content and detect malicious behaviors based on certain characteristics such as forceful web browser redirections. OdoSwiff used static analysis module to parse the tags of Flash files and dynamic analysis module to execute the Flash application and create an execution trace. The tool was evaluated on a big collection of Flash files that exhibit malicious behaviors. The results showed that OdoSwiff was able to detect malicious Flash advertisements in a better way compared with existing systems that scan Flash applications.

IV. Methodology

This section discusses the algorithms developed to collect advertisements from different types of websites, and how the tools of VirusTotal, URLVoid and TrendMicro are used to classify the collected advertisements as benign or malicious.

A. Data Collection

In the first step, we built a collection of web advertisements by crawling a total of 600 websites. We selected these websites from two different data feeds: a good and a bad data feeds, to make sure we have realistic number of web pages with benign and malicious ads. The good data feed was taken from Alexa's top million websites list while the bad data feed was taken from a blacklist, which contains web pages that have malicious behaviors such as spyware, reducing bandwidth use, and embedding in spam. This feed was managed by Dan Pollock who is a data director at a BC NDP political organization. We crawled 400 websites from Alexa's list feed: 200 were selected from the top of the feed and another 200 were selected from the middle of the feed. Moreover, we crawled 200 websites from the blacklisting feed: 100 were selected from the top of the feed and the other 100 were selected from the bottom of the feed.

Our crawler was based on Selenium, which is a web browser automation tool that allows a programmable control over the web browser. We programmed our crawler in Java to access different websites and extract the iframe's URL of each advertisement using Selenium functions and Google Chrome. Since we were using a blacklisting feed that contains websites that are blocked in the UAE web filter, we installed Browsec VPN to encrypt our traffic and access the blocked websites. Browsec VPN is a browser extension that can be installed from Chrome Web Store. The pseudo code for our crawler is depicted by Algorithm 1. This crawler is responsible for accessing a list of websites, one at a time, and extracting the URL of each ad iframe located in these websites. It takes the list of websites from a file called ”domain.txt”, and it saves the extracted URLs into ”URLs.txt” file located in the same directory.

In order for us to identify advertisements' URLs from other URLs, we used Websense Master Database to categorize our data and identify advertisements then we saved the updated list in a file called ”AdsURLs.txt”, where each URL occupies one line.

B. Malvertisements Detection Using VirusTotal

After accessing 600 websites and extracting advertisements' URLs, we ended up with a total of 230 ad URLs saved in ”AdsURLs.txt”. Next, we implemented a system that reads these URLs, one at a time and automatically submit them to VirusTotal website. For each URL, it opens VirusTotal website, searches for the URL text field, places the URL, and clicks on the ”Scan it!” button. After the page loads, our system

Algorithm 1 Collecting URLs

```
driver ← new ChromeDriver() WebDriver
$Open"domain.txt"ForInputAsInputFile$
$Open"URLs.txt"ForOutputAsOutputFile$
$ReadLinefromInputFile(String)$
loop1:
if $Line \neq null$ then
    $driver.browse(Line)$
    iframeElements ← driver.findElementsByTagName(”iframe”)List(WebElement)
    loop2:
    if $iframeElements.src \neq null$ then
        $WriteiframeElements.urlToOutputFile(String)$
        iframeElements ← driver.findElementsByTagName(”iframe”)List(WebElement)
        goto $loop2.$
        close;
    $ReadLinefromInputFile(String)$
    goto $loop1.$
    close;
$driver.close$
```

extracts the detection ratio and analyzes it. If the detection ratio is higher than one, then it will classify the advertisement as a malicious, saves the URL in ”VirusTotalResults.txt” file and increments malicious counter by one. Otherwise, it will increment the benign counter by one. At the end, we will get the number of malicious advertisements and the number of benign advertisements as classified by VirusTotal. Algorithm 2 shows the pseudo code of our Algorithm.

Algorithm 2 Detecting Malvertisements Using VirusTotal

```
benign ← 0 int
malicious ← 0 int
driver ← new ChromeDriver() WebDriver
$Open"AdsURLs.txt"ForInputAsInputFile$
$Open"VirusTotalResults.txt"ForOutputAsOutputFile$
$ReadLinefromInputFile(String)$
loop:
if $Line \neq null$ then
    $driver.browse("https://virustotal.com/")$
    $driver.findURLTabElement.click()$
    $driver.findURLTextFieldElement.write(Line)$
    $driver.findScanItButton.click()$
    if $DetectionRatioText \neq 0$ then
        malicious ← malicious+1
        $WriteLineToOutputFile(String)$
    else
        benign ← benign+1
    $ReadLinefromInputFile(String)$
    goto $loop.$
    close;
$Write"malicious = "maliciousToOutputFile(String)$
$Write"benign = "benignToOutputFile(String)$
$driver.close$
```

C. Malvertisements Detection Using URLVoid

Similar to VirusTotal, we implemented a system that reads the same URLs, one at a time, from ”AdsURLs.txt” file and automatically submit them to URLVoid website. For each URL, it opens URLVoid website, searches for the URL text field, places the URL, and clicks on the ”Submit Now” button. After the page loads, our system extracts the number that correspond to safety reputation. If the safety reputation number is higher than one, then it classifies the advertisement as a malicious, saves the URL in ”URLVoidResults.txt” file and increments malicious counter by one. Otherwise, it will increment the benign counter by one. At the end, we will get the number of malicious advertisements and the number of benign advertisements detected by URLVoid. Algorithm 3 shows the pseudo code of our Algorithm.

Algorithm 3 Detecting Malvertisements Using URLVoid

```
benign ← 0 int
malicious ← 0 int
driver ← new ChromeDriver() WebDriver
$Open"AdsURLs.txt"ForInputAsInputFile$
$Open"URLVoidResults.txt"ForOutputAsOutputFile$
$ReadLinefromInputFile(String)$
loop:
if $Line \neq null$ then
    $driver.browse("http://www.urlvoid.com/")$
    $driver.findURLTextFieldElement.write(Line)$
    $driver.findSubmitNowButton.click()$
    if $SafetyReputationText \neq 0$ then
        malicious ← malicious+1
        $WriteLineToOutputFile(String)$
    else
        benign ← benign+1
    $ReadLinefromInputFile(String)$
    goto $loop.$
    close;
$Write"malicious = "maliciousToOutputFile(String)$
$Write"benign = "benignToOutputFile(String)$
$driver.close$
```

D. Malvertisements Detection Using TrendMicro

In addition to VirusTotal and URLVoid, we implemented a system that reads the same URLs, one at a time, from ”AdsURLs.txt” file and automatically submit to TrendMicro website. For each URL, it opens SiteSafety Center website, searches for the URL text field, places the URL, and clicks on ”Check Now” button. After the page loads, our system extracts the assigned score. If the score is ”Dangerous”, then classifies the advertisement as malicious, saves the URL in ”URLVoidResults.txt” file and increments malicious counter by one. Otherwise, it will increment the benign counter. At the end, we will get the number of malicious advertisements and the number of benign advertisements detected by TrendMicro. Algorithm 4 shows the pseudo code of our procedure.

V. Results and Discussions

In this section, we discuss the experimental results, compute the confusion matrix and the accuracy for each detection algorithms. After crawling 600 websites from two different data feeds, we were able to extract 230 advertisements' URLs. Submitting the list of these URLs to the three detection systems

Algorithm 4 Detecting Malvertisements Using TrendMicro

```
benign ← 0 int
malicious ← 0 int
driver ← new ChromeDriver() WebDriver
$Open"AdsURLs.txt"ForInputAsInputFile$
$Open"SSCResults.txt"ForOutputAsOutputFile$
$ReadLinefromInputFile(String)$
loop:
if $Line \neq null$ then
    $driver.browse("https://global.sitesafety.trendmicro.com/index.php")$
    $driver.findURLTextFieldElement.write(Line)$
    $driver.findSubmitButton.click()$
    if $IsItSafeText == Dangerous$ then
        malicious ← malicious+1
        $WriteLineToOutputFile(String)$
    else
        benign ← benign+1
    $ReadLinefromInputFile(String)$
    goto loop.
    close;
$Write"malicious = "maliciousToOutputFile(String)$
$Write"benign = "benignToOutputFile(String)$
$driver.close$
```

separately, and examining the false positives and negatives produced the following results:

- VirusTotal, detected 28 malicious advertisements out of 230 submitted. The VirusTotal percentage of detected malicious advertisements was 12.2% of all extracted advertisements and 4.67% of the crawled URLs.
- URLVoid, detected 40 malicious advertisements out of 230 submitted. The URLVoid percentage of detected malicious advertisements was 17.4% of extracted advertisements and 6.67% of the crawled URLs.
- TrendMicro, detected only 1 malicious advertisement out of 230 submitted. The TrendMicro percentage of detected malicious advertisements was 0.4% of extracted advertisements and 0.167% of the crawled URLs.

TABLE I. VIRUSTOTAL CONFUSION MATRIX

		Prediction outcome		
		n	**p**	**total**
actual value	**n′**	True Negative 32	False Positive 3	N′ = 35
	p′	False Negative 28	True Positive 7	P′ = 35
	total	N = 60	P = 10	

The next step was to evaluate the efficiency of the three detection systems by constructing the confusion matrix for each one of them. We took a sample of 70 advertisements in which 35 advertisements were benign and the other 35 were malicious. Our classification decision was made after studying the historical reputation of the site and the lookup database of malicious reported URLs and blacklists. Then we submitted the URLs of these 70 advertisements to the three detection systems. Finally, we compared the results provided by the detection systems with the actual classification.

TABLE II. URLVOID CONFUSION MATRIX

		Prediction outcome		
		n	**p**	**total**
actual value	**n′**	True Negative 30	False Positive 5	N′ = 35
	p′	False Negative 14	True Positive 21	P′ = 35
	total	N = 44	P = 26	

TABLE III. TRENDMICRO CONFUSION MATRIX

		Prediction outcome		
		n	**p**	**total**
actual value	**n′**	True Negative 35	False Positive 0	N′ = 35
	p′	False Negative 34	True Positive 1	P′ = 35
	total	N = 69	P = 1	

Each confusion matrix consists of two rows and two columns that indicate the number of true positive (malicious and the system classified it as malicious), true negative (benign and the system classified it as benign), false negative (malicious but the system classified it as benign), and false positive (benign and the system classified it as malicious). Tables I, II, and III present the confusion matrices for VirusTotal, URLVoid, and TrendMicro, respectively.

From these confusion matrices, we observe that URLVoid has the highest true positive value, and the lowest false negative value. On the other hand, VirusTotal comes in the second place where the true positive value is lower than URLVoid and higher than TrendMicro, and its false negative value is higher than URLVoid and lower than TrendMicro. Furthermore, we calculated the accuracy of each system by adding the number of true positive and the number of true negative and dividing the total by the number of samples take (i.e.70). The accuracy measures the classifier correctness in finding the total number of predictions that were correct [21]. We found that URLVoid is the most accurate system with 73% accuracy, then comes

VirusTotal with 56% accuracy, and finally the lowest accuracy measured was for TrendMicro at 51%. Based on accuracy measurements URLVoid was is the best detection system that can help detect malicious advertisements. That is because URLVoid uses a combination of both website scanners and domain blacklists. While VirusTotal uses Antivirus engines and website scanners only without relying domain blacklists. TrendMicro, on the other hand, does not rely on other website reputation engines or scanners, it classifies URLs on its own using a number of factors such as website age and historical locations. This indicates that there is no best tool, the more tools and blacklists we combine the more accurate results we will get.

VI. Conclusions

This paper contributes three automatic algorithms to extract ad URLs and submit them to malicious domains detection tool. Wet built a system that was able to crawl 600 websites from two different data feeds, extract up to 230 advertisements' URLs, and submit those URLs to three online malicious websites detection systems: VirusTotal, URLVoid, and TrendMicro to classify them as malicious or benign. Furthermore, the paper evaluated the performance of each detection tool by calculating their confusion matrices and accuracy. We conclude that URLVoid is the most accurate detection system. However, we are working on run more malicious websites detection tools to increase the detection accuracy.

Acknowledgments

This work was supported by Zayed University Research Office, Research Cluster Award # R17079.

References

[1] G. Bakshi and S. K. Gupta, "Online advertising and its impact on consumer buying behavior," *International Journal of Research in Finance and Marketing*, vol. 3, no. 1, pp. 21–30, 2013.

[2] (2015, Oct.) Digital ad revenues surge 19%, climbing to $27.5 billion in first half of 2015. [Online]. Available: http://www.iab.com/news/digital-ad-revenues-surge-19-climbing-to-27-5-billion-in-first-half-of-2015- according-to- iab-internet-advertising-revenue-report/

[3] M. Aldwairi and R. Alsalman, "Malurls: Malicious urls classification system," in *Annual International Conference on Information Theory and Applications*, 2011.

[4] M. Aldwairi and Y. Flaifel, "Baeza-yates and navarro approximate string matching for spam filtering," in *The Second International Conference on Innovative Computing Technology (INTECH 2012)*, 2012.

[5] A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna, "The dark alleys of madison avenue: Understanding malicious advertisements," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014, pp. 373–380.

[6] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, "Knowing your enemy: Understanding and detecting malicious web advertising," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 674–686. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382267

[7] M. Aldwairi and N. Ekailan, "Hybrid pattern matching algorithm for intrusion detection systems," *Journal of Information Assurance and Security*, vol. 6, no. 6, pp. 512–521, 2011.

[8] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious javascript code," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 281–290.

[9] X. Xing, W. Meng, B. Lee, U. Weinsberg, A. Sheth, R. Perdisci, and W. Lee, "Understanding malvertising through ad-injecting browser extensions," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2015, pp. 1286–1295. [Online]. Available: https://doi.org/10.1145/2736277.2741630

[10] M. Aldwairi and K. Al-Khamaiseh, "Exhaust: Optimizing wu-manber pattern matching for intrusion detection using bloom filters," in *Web Applications and Networking (WSWAN), 2015 2nd World Symposium on*. IEEE, 2015, pp. 1–6.

[11] S. Ford, M. Cova, C. Kruegel, and G. Vigna, "Analyzing and detecting malicious flash advertisements." in *ACSAC*, 2009, pp. 363–372.

[12] D. L. Rubinfeld and J. D. Ratliff, "Online advertising: Defining relevant markets," *Journal of Competition Law and Economics*, vol. 6, pp. 653–686, 2010.

[13] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna, "Understanding fraudulent activities in online ad exchanges," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 279–294.

[14] J. Segura. (2013, Dec) Malvertising and the joys of online advertising. [Online]. Available: https://blog.malwarebytes.com/threat-analysis/2013/12/malvertising-and-the-joys-of-online-advertising/

[15] Virustotal scanner. [Online]. Available: http://securityxploded.com/virus-total-scanner.php

[16] M. Aldwairi, "Hardware efficient pattern matching algorithms and architectures for fast intrusion detection." Ph.D. dissertation, North Carolina State University, 2006.

[17] M. Kharbutli, M. Aldwairi, and A. Mughrabi, "Function and data parallelization of wu-manber pattern matching for intrusion detection systems," *Network Protocols and Algorithms*, vol. 4, no. 3, pp. 46–61, 2012.

[18] Vikram. Urlvoid online scanner to find if a website is safe to visit. [Online]. Available: http://www.technorms.com/1284/urlvoid-online-scanner-to-find-if-a-website-is-safe-to-visit

[19] What is urlvoid? [Online]. Available: http://www.urlvoid.com/about-us/

[20] Site safety center. [Online]. Available: https://global.sitesafety.trendmicro.com

[21] K. Markham. (2014, March) Simple guide to confusion matrix terminology. [Online]. Available: http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/