Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 124 (2017) 664–671

Procedia Computer Science

www.elsevier.com/locate/procedia

4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia

Typosquat Cyber Crime Attack Detection via Smartphone

Zakiah Zulkefli, Manmeet Mahinderjit Singh*, Azizul Rahman Mohd Shariff, Azman Samsudin

School of Computer Sciences,University Sains Malaysia 11800 Penang, Malaysia

Abstract

A Smartphone is a multi-purpose device that can act as both mediums of communications and entertainment due to the availability of various sensors and services, such as SMS, NFC and Bluetooth. Through these functionalities, Smartphone owner can exchange information to each other by sharing links or even files. However, an attacker see these as an advantage to perform an Advanced Persistent Threat (APT) attack. APT is an attack which incorporates both social engineering attack and malware. In this paper, the authors will shed light on how APT attack through spear phishing can occur in Smartphone and how to detect it. First, the authors will examine the tactics that can be used by the attacker to perform a successful social engineering attack. Then, based on the discussion that has been made, the authors have used a machine learning algorithm to classify whether a certain URL is a phish or not. Lastly, the authors have evaluated the propose technique using machine learning and obtained more than 90% accuracy. This proves , that the proposed technique would able to help mitigating APT attack through spear phishing in the Smartphone.

Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

Keywords: Advanced Persistent Threat (APT); Cyber Crime; Typosquat; Smart Phone; Decision Tree

1. Introduction

Advanced Persistent Threat (APT) attack is a carefully planned attack that involved both social engineering and malware, where spear phishing is the most popular method that has been used by the attacker [1]. The attacker will send email to the targeted victim by including link(s) to the targeted web or malicious attachment (e.g. Trojan horse). In order to achieve a high chance of success, the spelling of the link or URL will have high similarity with the link that the victim's frequently accessed to. This is known as typosquatting or url hijacking. Unlike computer, where the

* Corresponding author. Tel.: +604-6535346[2].; fax: +0-000-000-0000 .
E-mail address: manmeet@usm.my

1877-0509
Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017
10.1016/j.procs.2017.12.203

spear phish is likely to happen through email, there are various mediums that can be used to perform the social engineering attack in Smartphone. For example, in 2013, a Tibetan activist had become one of the targeted victims by a sophisticated attacker through Smartphone [2]. The attacker used the social engineering attack by sending email with Android application's file (.apk) attached. Once the victim has downloaded the file and executed it, the application will act as a backdoor that sends several information to the attacker. This event proof that Smartphone can be used as a medium to perform a sophisticated and well planned attack.

This era, Smartphone has been widely adopted as a personal device to be used in the working environment or known as Bring Your Own Device (BYOD), thus it is very important to protect the device from becoming the next target of an APT. However, there are still lack of research in this area. Most of the research on APT attacks are focused on malware features [8,7,21,22,23] and not on the URL features. Thus, this research will explore how APT attack can occur in Smartphone through url hijack and how to detect it. There are three main objectives of this research. First, to investigate the how APT attack through spear phishing could happen in Smartphone. Second, to propose an enhanced tool for spear phish detection by examine the URL and website features. This proposed technique is called LESSIE, LabEl baSed Spear phIsh dEtection. Third, to further employ the LESSIE and make evaluation by using machine learning technique. The main contribution of this research is a technique on detecting an APT attack through spear phishing on Smartphone by examining the URL that has been received.

This research can be divided into five sections. In Section 2, the research will highlight about how APT can occur in Smartphone and the related works on how to detect phishing website. Then, in Section 3, the research will discuss about LESSIE's components and the experimental setup. In Section 4, the research will discuss about the results that have been obtained and its evaluation. Lastly, in Section 5 will conclude the research's finding.

2. State-of-art

Adopting Smartphones as personal device to be used in a working environment or BYOD might bring benefit towards employee. However, not all of the organizations are aware of the needs to implement security policy on the BYOD in order to prevent information leakage that can happen due to the employee's behaviour and different level of awareness [3]. In this section, the research will discuss in details on how an APT attack can occur in Smartphone and methods that can be used in order to detect it.

2.1. Advanced Persistent Threat (APT) : Definition And Methods

Advanced Persistent Threat (APT) is a sophisticated attack that involves social engineering and malware. In general, the APT attack can be divided into three stages, where each of it are chained together and keep repeating until the attacker has gained the information that they want. They are information gathering, social engineering attack and malware. Previously, there are several real cases of APT attacks that have been reported [4, 5]. APT leave huge impacts to the affected party in term of monetary gain, security and trust relationship. Hence, it is very important to prevent the threat from happening

An APT attack can happen through spear phishing, exploiting the web infrastructure through SQL injection and exploiting the social network [6]. The likelihood and impact of spear phishing attack on BYOD is the highest [7] where, 91% spear phishing of APT attack occur through email [1]. Thus, APT through spear phishing has become the main focus of this research. In this section, details on how the spear phish attack occurs in Smartphone will be discussed.

2.1.1 Spear phish attack in Smartphone

In Smartphone, the medium of spear phishing are email, services and sensors. These sensors are used as a medium to transferring file from one device to another, share services (e.g. connect to headphone) and communication. However, it can be exploited by the attackers in order to perform a spear phish attack by sending malware in the form of .apk file to the targeted victim or sharing malicious links. For example, sending a SMS or MMS that contains URL with/or attachment with it by using content that the user familiar with, transferring malicious file with bluetooth, scanning NFC tag or QR code which later lead the user to a malicious website to download certain document or drive-by-download. Compared to malware, victims usually more prone to click on a URL as malware requires request access

to a certain permission(s) such as Internet in order to gain access to the information. Previously, APT attack called Adwind [4] has tricked its victim by using this method. Thus, in this research, the authors will focus on spear phish that occurred through the website link as it has high impact to the organization, where it can lead to both malware and information gathering by the attacker. In general, a success spear phish attack will manipulate the user's familiarity with certain URL [8]. This is known as typosquatting. The attacker is taking advantage of the lack of user judgment when they are meeting with the URL that is almost similar to the one that they frequently accessed [9]. This attack had led the associated organization to lost clients' trust due to the misleading content of the website [10, 11]. Thus, this research will focus on typosquatting as it will likely employ to perform APT attack due to its ambiguity. Next, the research will discuss the url hijacking or typosquatting attack in depth.

2.1.2 Spear phish attack through URL vulnerabilities

A study conducted by Agten et al. [12] on the number of active typosquatting domain of 500 most popular website have shown that each day there are more than 10,000 malicious typosquatting domain produced. The typo URL or known as typo-neighbourhood can be further categorized into [13]:- missing-dot typo , character omission typos, character permutation typos, character replacement typos and character insertion typos. However, giving a random URL without considering the victim's interest will lower their trust. Thus, it is important to use URL that they are familiar with. One of the way is by looking on the victim's browser history. In Android, it can be accessed by gaining permission to *android.browser.permission.read*. Previously, this method has been used by APT attack called CloudAtlas [5]. Since the attacker will likely use the URL that can be obtained from the browser's history, thus by examining the similarity between the URL in the browser's history with the URL obtained from the received message would help preventing the user from visiting the phish site. Next, this research will discuss on how to examine these URL.

2.1.3 Similarity Search

In general, a uniform resource locator or URL structure can be divided into several parts [14]:- protocol, domain name, port, path, parameters and anchor. A domain name is the most important part as it points the address of the associated webpage. The domain name can be furthered divided into several parts [15]:- label(s) and Top-Level Domain (TLD). For example, consider a victim receives a link of *abc.example.com*, then *abc.example.com* is the domain name, while *abc.example* is the label and *com* is the TLD. A label specifies the identity or brand of the webpage, meanwhile the TLD specifies more general details of the webpage, for example *.gov* specify that the webpage is a government webpage. Previously, Maurer and Höfer [16] and Kang and Lee [17] has performed a similarity check between malicious and legitimate domain name. Kang and Lee [17] use Ratcliff/Obershelp pattern matching algorithm, while Maurer and Höfer [16] use Levenshtein algorithm. The Levenshtein algorithm detects the number of words that need to be changed in order to become the words that have been queried. For example, 4 words need to be changed from *abcd* in order to become *zfjk*. Thus, it is out of this research. Meanwhile, the Ratcliff/Obershelp algorithm works by examining the longest common substring available [18]. For example, consider a legitimate website of *www.yahoo.com* and illegitimate website of *www.yahho.com*. At first, the longest common substring will be searched first, which is *www.yah*. The score will be given by multiplying the number of characters with two. Thus, making *www.yah* to have 14. Then, the other group would be the remaining string on the left and right side of the longest common substring. Here, the other group will be on the right side, which is *ho.com* and *oo.com*. The common substring of this group would be *o.com*, thus the score would be 10. This makes the total score of the common substring to be 10 + 14 = 24. Hence, the overall similarity between these two domain name would be 24/26*100 = 92.31%. This indicates that *www.yahho.com* is likely a phish website as it exceeds the value of 78%, which is the threshold for a phish website [17]. Thus, this research will employ Ratcliff/Obershelp pattern matching algorithm as the URL similarity search.

However, differ from Kang and Lee [17] that make comparison by using domain names (e.g. *www.google.com*), this research will only make comparison between labels only, without including the 'www'and Top Level Domain (TLD), so that the accuracy could be increased. For example, consider that the users frequently visit *www.example.jp* (URL A), but he receives a URL of *www.examples.com* (URL B). Without discarding the TLD and '*www*', we will have a similarity of 72%. However, if we discard the TLD and '*www*', the similarity score will be 92%. This indicates that the proposed method will give more accurate result. Imagine if the attacker is using a long generic TLD. This will decrease the similarity score. However, it is hard to determine where the position of TLD as it can exist in multiple

layer, for example '*co.uk*'. In order to solve this issue, Mozilla has created a list of TLD that updated daily [19] and has been implemented in PHP by Florian Sager[20]. However, by depending on the string similarity between the two domain name only is not enough, as not all domain name that has a high similarity score is bad. Hence, this shows that other detection method needs to be combined together. This will become the next discussions.

2.2. Available Solutions on Preventing Phishing Attacks.

Previously, there are several APT detections method introduced. These methods employs on malware behavior and content via malwares API call [21] and malware DNS activities & network IP addresses [8, 17]. Analysing phish email by contents [22] and Smartphone network traffic [23] are another two methods proposed. Hence, none of the research work focus on URL comparison in phishing detection on Smartphone and this show a clear gap to our proposed method. In general, the methods used for phishing detection are comparing hash image between legitimate and illegitimate websites [24], seeking the most used words in a webpage [25], heuristic [26, 27], blacklisting and reputation based [28] . However, comparing hash and calculating the frequency of certain words are not suitable to be used since current sophisticated attackers are able to mimic carefully the interface of a legitimate website [29]. Meanwhile, reputation based method could suffer in term of unfair rating [28]. Heuristic approach leads to high chance of false positive, whereas blacklist has a low positive rate, but most of it requires human verifications in order to obtain an accurate result and lack the ability to detect fresh phish website [30]. Among all of these methods, blacklisting is the most convincing method in order to detect phish websites. Thus, this research will use a combination of heuristic and blacklist method. The usage of machine learning techniques such as decision tree has been arise due to the learning patterns simplicity and high accuracy.

3. Material and Methods

LESSIE refers to the technique of detecting spear phish by making comparison between the link that have been received with the websites that have been accessed by the user through the label of the domain names and its features. In general, LESSIE has been implemented on a machine with Core i5-4210U Intel Processors and 8 GB RAM and Samsung Galaxy Note S3 with Android 5.0. Then, the result obtained has been evaluated using Decision Tree with 10 fold cross validation in RapidMiner. In order to get an accurate form of URL that is sent to the Smartphone, first the research has collected 30 URLs from SMSs, NFC tags and smart codes. Then, the remaining 1415 URLs are the general URLs from Alexa, typosquatting domain names that are generated using Kali Linux [33] and obtains through PhishTank's blacklist [34]. This research also has taken a browser's history log from 1 user. From the browser's history, only the domain names and the total access are taken, as the domain name symbolized the brand of the website, thus it will become the main target of the attacker. Items below are the component of LESSIE:-

- **Obtains the redirect URL (RU)** - Based on the data that has been collected, out of 1445, about half of them are redirected to another website, which could be in different domain name or path. However, this occurrence cannot be set as a pre-condition to make a decision tree rule. This is because, the length of SMS that is limited to 140 characters, the total space of NFC tag and design of the QR code, which is small in size has made senders to apply URL shortener, eventhough the URL is a trusted one. Thus, only the redirect location needs to be taken into consideration.
- **Obtains the results from search engines (SE)** - Obtained from the Google Custom Search API [32]. In this research, the Top 10 items return by the Google Custom Search will be used as an indication whether it is a phish or not. In order to make sure that the return result will give priority to the website that has been queried, this research has put the keyword "site:"along with the URL [35]. For example, *site:www.example.com*. As a result, only related site will be returned.
- **Check website if it is in the blacklist (B)** - Checked through Google Blacklist API [36]. This method has been used by Basnet et al. [26]. Mohammad et al. [27] stated that Google Blacklist API generally update every seven hours.

- **Obtain total request to another URL (RE)** - In order to check whether there is an existing link that will direct the user to another web page, a href call needs to be obtained through regex. Here, the research have taken the percentage of the number of href to the website which have di_erent domain name.
- **Obtain domain age (DA)** - Obtained by performing Whois lookup using JsonWhoIs API [37].
- **Obtain the domain rank (DR**) - Obtained by using Alexa API [38]. As the popularity of certain URL is highly correlated with country, thus it is unfair to evaluate its ranking globally. Hence, in this research, the ranking of a URL is taken based on its published country.
- **Obtain similarity scores (SS)** - Employed the Ratcliff/Obershelp algorithm.

First, the value of SS of domain name's label that is received by the user with the user's browser history is checked. If the SS is 100, then that mean their label is the same, thus second round of SS is made in order to check whether their TLD is similar. This time, by using domain name. If they are both 100% and the user has visited the site more or equal to 10 times, then the URL is classified as good. On the other hand, if the value of SS is less than 100% or equal to 100% but, the user never or has accessed the site less than 10 times, the value of B is checked. If it returns yes or exist in the blacklist, then it is immediately classified as bad. Otherwise, the value of RE is taken, where the URL is classified as bad if it contains more than or equal to 61% of reference to another website.

Next, if the RE is less than 61%, then the value of SE is checked first, beginning with the domain name. If the domain does not exist, then it means that the website has not been crawled. If it is still not crawled, the value of DA is checked, where if it is less than 6 months, then it is categorized as bad. Otherwise, the value of DR is checked. If it is bigger than 100,000, the URL is classified as bad. Otherwise, it is classified as suspicious. On the other hand, if the domain name exists in SE, then the existance of the URL is checked. After that, the value of DA is checked. If the DA is less than 6 months, then the URL is classified as suspicious, since bad URL can also exist in the search engine. Otherwise, the value of DR is checked, where if it is equal to or less than 100,000, then the URL is good. Otherwise, the URL is classified as bad. Lastly, the research has used Decision Tree algorithm in order to make evaluation of the proposed technique by using RapidMiner software. The evaluation is made through multi-class confusion metric, with 3 labels instances:- good, suspicious and bad. Next, the research will discuss more about the result of the experiment.

4. Results and discussions

In this section, the authors will provide the results of the experiment and discussions about the LESSIE. During the experiment, there are 24 unique domain names that have been accessed by the user. Based on the data obtained, the websites which are interested most by the user have a total accessed of 39. On the contrary, the websites which are less interested by the user have been accessed between 1 and 9. Thus, from here, an assumption of websites that are less interest by the user should have a total accessed of less than 10. On the other hand, 1445 of sample URLs have been collected. Among them, 476 URLs are collected from crazyurl, 500 urls are collected from PhishTank, 500 are collected from Alexa and 47 are collected from SMSs, NFC tags and smart codes. From the samples, 252 of them are suspicious website, 738 of them are bad website and 455 of them are good website. Among them, 432 of the URLs are dead and 1014 of the URLs are alive.In this research, the dead URLs are still considered as a data in order to understand its properties.

4.1. Results

In order to determine the effectiveness of the proposed technique of calculating the similarity score through label of the domain name, the experiment has been conducted using two different techniques. Both of them are using the same rules for feature extraction but, different technique to calculate the similarity score. The first one is using domain name (e.g. *www.example.com*), while the second one is through domain name's label (e.g. *example*) or LESSIE, as shown in Table 1. Based on Table 1, LESSIE achieves 97.51% of accuracy, while domain based achieves 99.72% of accuracy. LESSIE also have more false negative compared to the domain based technique. However, only LESSIE achieves the research objective. Details discussions regarding the results achieved are discussed in the next of subsection.

Table 1. Results of decision tree with 10 fold cross-validation of 1440 URLs. (Notes : G = Good, S = Suspicious and B = Bad)

Predicted vs. Actual		(a) Results of decision tree with 10 fold cross-validation of 1440 URLs using domain based technique				(b) Results of decision tree with 10 fold cross-validation of 1440 URLs using LESSIE			
		Actual			Class Precision	Actual			Class Precision
		G	S	B		G	S	B	
Predicted	G	739	0	2	99.73%	731	0	6	99.19%
	S	0	453	1	99.78%	0	448	16	96.55%
	B	1	0	241	99.59%	7	7	230	94.26%
	Class Recall	99.86%	100.00%	98.77%		99.05	98.46%	91.27	

4.2. Discussions

This section will provide a detail discussions about result achieves previously, starting with decision tree classifier, URL features and techniques. Lastly, overall research's results and discussion is summarized.

4.2.1. Decision Tree Classifier

The results obtained on the previous subsection (Section 4.1) proves that by using different technique for calculating the similarity score will affect the decision making rules. Due to space limitation, Fig. 1 is shown with structure of trees depending on how the similarity score is calculated. The research obtained that using the label based technique could assist the detection of spear phishing more compared to domain based technique, where it has been used in the second and fifth-tier of decision making process. The main factor that result to this finding is, label based technique has made the process of recognizing a typosquatting website become more significant as the score are calculating using meaningful value only, thus assisting the classification of decision tree.

```
urlRank > 0.500
|  urlTotalHref > 60.967: bad {bad=75, good=0, suspicious=0}
|  urlTotalHref ≤ 60.967
|  |  urlGoogleBlacklist = No
|  |  |  isDomainExistInGoogle = FALSE
|  |  |  |  similarityScore > 26.136: suspicious {bad=0, good=7,
suspicious=147}
|  |  |  |  similarityScore ≤ 26.136: good {bad=0, good=2, suspicious=0}
|  |  |  isDomainExistInGoogle = TRUE
|  |  |  |  urlRank > 95747: suspicious {bad=0, good=0, suspicious=6}
|  |  |  |  urlRank ≤ 95747: good {bad=0, good=443, suspicious=15}
|  |  urlGoogleBlacklist = yes: bad {bad=12, good=0, suspicious=0}
```

```
urlRank ≤ 0.500
|  similarityScore > 97.826: good {bad=0, good=3, suspicious=1}
|  similarityScore ≤ 97.826
|  |  isUrlExistInGoogle = FALSE
|  |  |  isDomainExistInGoogle = FALSE: bad {bad=600, good=0,suspicious=6}
|  |  |  isDomainExistInGoogle = TRUE
|  |  |  |  urlGoogleBlacklist = No: suspicious {bad=5, good=0, suspicious=28}
|  |  |  |  urlGoogleBlacklist = yes: bad {bad=37, good=0, suspicious=0}
|  |  isUrlExistInGoogle = TRUE
|  |  |  urlTotalHref > 60.258: bad {bad=7, good=0, suspicious=0}
|  |  |  urlTotalHref ≤ 60.258: suspicious {bad=2, good=0, suspicious=49}
```

Fig. 1. Decision tree rules using LESSIE for the detection of spear phish by using 1440 URLs

4.2.2. Comparison between Label Based & Domain Based Technique

In order to understand the proposed technique, the authors have conducted the experiment by using 476 typosquatting URLs (obtained from crazyurl) of websites that has been visited by the user for more than 10 times. Then, any duplicate data is removed, where 466 unique URLs have been obtained. Next, the average similarity score is obtained by dividing the summation of similarity score of typosquatting URLs with total unique URLs. On average, the total similarity score for the proposed technique (label based) is 84.14% while, domain based technique is 88.97%. This indicates that there is a high probability that the typoquatting URL has the same length of label and domain name as the legit URL, but with a small level of modification. Thus, achieving high value of longest common substring. However, the research proposed technique score lower than the domain based similarity score because the proposed technique only consume small amount of common substring after the TLD and '*www*' has been discarded from the URLs. Thus, making it more compact and comparisons is made using meaningful value. As a prove, consider the user receives a URL of *http://foogle.com*. By using the domain based technique, the result would be, 1) *google.com* (85.71%), 2) *trc.taboola.com* (53.85%), 3) *us.yahoo.com* (52.17%) and 4) *ca.yahoo.com* (52.17%). Meanwhile, using the research technique (label based), the result would be 1) google (83.33%), 2) *trc.taboola* (35.29%) , 3) *support.sonymobile* (33.33%) and 4) *ca.yahoo* (28.57%).

4.2.3. URL Feature

The similarity score of domain name's label and total access have improved the detection of spear phishing through typosquatting by enabling the identification of malicious URL. This can be obtained by looking at the total access to certain URL. In this experiment, based on the user's history, the less interested site has been accessed not more than 10 times. Thus, any URL received that has a 100% similarity score and have been accessed more than 10 times should be classified as good. Otherwise, if it has been accessed less than 10 times or have a similarity score of less than 100%, the URL can fall between good, bad and suspicious as not all URLs that are almost similar to legitimate one are bad.

By using LESSIE, 8.58% of the typosquatting URLs are classified as good, 68.32% are bad and 22.96% are classified as suspicious. The factors that contribute to the good classifications are 1) the legit domain has taken ownership of the URL where they are allowing it to redirect to the legit domain, 2) website features of lower total reference to another site, low rank and exist in the search engine. This kind of classification mostly occurs to the alive URLs. Meanwhile, for the dead URLs, they are classified as either bad or suspicious. The main factor that has caused the URL to be classified as suspicious is, the URL rank is depended on the number of queries made within a certain period of time. As the interface of a keyboard for Smartphone is small, thus it can lead to misspelling errors, that later contributes to a high number of queries with spelling mistakes. As a result, an attacker can take advantage on this issue. Thus, this proves that similarity score would help the prevent spear phishing through typosquatting. In this research, the authors have perform the checking by the URL's label instead of its domain name because most of the attackers are going to change the TLD instead of its domain name. Although it achieves more false negative compared to the domain name, but based on the experiment that have been done, only the LESSIE able to identify typosquatting website. Next, the research will provide the conclusion and future work.

5. Conclusion and Future Work

In this paper, the authors have shown how APT attack trough spear phishing by exploiting the use of sensors can occur through Smartphone. In this research, authors have proposed LESSIE; a technique of detecting whether the URL received will be linked to a phish website by comparing it with the URLs exist in the browser's history. Based on the preliminary results, LESSIE achieves 97.51% of accuracy. Although it is lower compared to the domain name based technique [17], but LESSIE provide a better coverage in term of detection and more thoroughly. The limitations of this research are, only a small data set is used and the browser's history is taken from one user only. Besides that, the threshold of total access to interested website might differ from one user to another. Thus, as for future work, the authors plan to test the proposed technique with bigger input data and real data set from multiple users. More work will be done next in order to reduce the false negative errors.

Acknowledgement

This work is supported by the Fundamental Research Grant Scheme (FRGS) No : 203/PKOMP/6711424 of under the Universiti Sains Malaysia and Government of Malaysia.

References

[1] APT Research Team TrendLab, Spear-phishing email: Most favored apt attack bait (research paper), Report (2012).
[2] K. Baumgartner, C. Raiu, D. Maslennikov, Android trojan found in targeted attack (2013). [Online]. Available: https://securelist.com/blog/incidents/35552/android-trojan-found-in-targeted-attack-58/
[3] Z. Zulkefli, M. Mahinderjit-Singh, N. Malim, Advanced Persistent Threat Mitigation Using Multi Level Security Access Control Framework, Vol. 9158 of Lecture Notes in Computer Science, Springer International Publishing, 2015, book section 7, pp. 90–105.
[4] Myonlinesecurity, Spoofed wupos agent portal upgrade for all agents delivers java adwind/jacksbot (2017). [Online]. Available: https://myon linesecurity.co.uk/spoofed-wupos-agent-portal-upgrade-for-all-agents-delivers- java-adwind-jacksbot/
[5] SonicWall, Sonicwall security center (2014). [Online]. Available: https://www.mysonicwall.com/sonicalert/searchresults.aspx?ev=article&id=768
[6] A. K. Sood, R. J. Enbody, Targeted cyberattacks: A superset of advanced persistent threats, Security & Privacy, IEEE 11 (1) (2013) 54 61.
[7] L. B. Lau, M. M. Singh, A. Samsudin, Trusted system modules for tackling apt via spear-phishing attack in byod environment, Universiti Sains Malaysia, Thesis (2015).

[8] G. Zhao, K. Xu, L. Xu, B. Wu, Detecting apt malware infections based on malicious dns and trac analysis, IEEE Access 3 (2015) 1132–1142.
[9] C. Harbison, New url malware will punish you for sloppy typing: Mac, windows computers vulnerable to latest '.om 'url hijacks (2016). [Online]. Available: http://www.idigitaltimes.com/new-url-malware-will-punish-you-sloppy-typing-mac-windows-computers-vulnerable-latest-519723
[10] M. A. M. Fusco, The cybersquatting countdown has begun: the moncler case (2016). [Online]. Available: http://www.lexology.com/library/ detail.aspx?g=9bb8c61b-3c95-4647-8087-fe69304acc20
[11] C. Roth, M. Dunham, J. Watson, Cybersquatting; typosquatting facebook's $2.8 million in damages and domain names (2013). [Online]. Available: http://www.lexology.com/library/detail.aspx?g=7088bf09-8a9e-4449-a179-d90bdfad3310
[12] P. Agten, W. Joosen, F. Piessens, N. Nikiforakis, Seven months' worth of mistakes: A longitudinal study of typosquatting abuse, in: Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015), Internet Society, 2015.
[13] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, B. Daniels, Strider typo-patrol: discovery and analysis of systematic typo-squatting, in: Proceedings of the 2nd conference on Steps to Reducing Unwanted Trac on the Internet - Volume 2, USENIX Association, 1251301, 2006, pp. 5–5.
[14] Mozilla, Learn web development : What is a url? (2016). [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Common questions/What is a URL
[15] Mozilla, Learn web development : What is a domain name? (2016). [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Common questions/What is a domain name
[16] M.-E. Maurer, L. Höfer, Sophisticated Phishers Make More Spelling Mistakes: Using URL Similarity against Phishing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 414–426.
[17] J. Kang, D. Lee, Advanced white list approach for preventing access to phishing sites, in: Convergence Information Technology, 2007. International Conference on, 2007, pp. 491–496.
[18] J. W. Ratcliff, D. E. Metzener, Pattern-matching-the gestalt approach, Dr Dobbs Journal 13 (7) (1988) 46–&.
[19] Mozilla, View the public sux list (2016). [Online]. Available: https://publicsuffix.org/list/
[20] F. Sager, Registered-domain-libs (2009). [Online]. Available: https://github.com/usrflo/registered-domain-libs/
[21] D. Moon, H. Im, I. Kim, J. H. Park, Dtb-ids: an intrusion detection system based on decision tree using behaviour analysis for preventing apt attacks, The Journal of Supercomputing (2015) 1–15doi:10.1007/s11227-015-1604-8.
[22] P. Dewan, A. Kashyap, P. Kumaraguru, Analyzing social and stylometric features to identify spear phishing emails, in: Electronic Crime Research (eCrime), 2014 APWG Symposium on, IEEE, 2014, pp. 1–13.
[23] C. Xenakis, C. Ntantogian, An advanced persistent threat in 3g networks: Attacking the home network from roaming networks, Computers & Security 40 (2014) 84–94. http://www.sciencedirect.com/science/article/pii/S0167404813001685
[24] J. S. White, J. N. Matthews, J. L. Stacy, A method for the automated detection phishing websites through both site characteristics and image analysis, Vol. 8408, 2012, pp. 84080B–84080B–11.
[25] Y. Pan, X. Ding, Anomaly based web phishing page detection, in: 2006 22nd Annual Computer Security Applications Conference (ACSAC'06), 2006, pp. 381–392.
[26] R. B. Basnet, A. H. Sung, Q. Liu, Rule-based phishing attack detection, in: Proc. International Conference on Security and Management (SAM11), 2011, pp. pp. 624–630.
[27] R. M. Mohammad, F. Thabtah, L. McCluskey, Intelligent rule-based phishing websites classification, IET Information Security 8 (3) (2014) 153–160. URL http://digital-library.theiet.org/content/journals/10.1049/iet-ifs.2013.0202
[28] A. Josang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decis. Support Syst. 43 (2) (2007) 618–644.
[29] R. Zhao, S. John, S. Karas, C. Bussell, J. Roberts, D. Six, B. Gavett, C. Yue, The highly insidious extreme phishing attacks, in: 2016 25th International Conference on Computer Communication and Networks (ICCCN), 2016, pp. 1–10.
[30] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, C. Zhang, An empirical analysis of phishing blacklists, in: Proceedings of Sixth Conference on Email and Anti-Spam (CEAS), 2009.
[31] J. Markey, A. Atlasis, Using decision tree analysis for intrusion detection: a how-to guide, SANS Institute.
[32] Google Developer, Google safe browsing : What is safe browsing ? (n.d.). [Online]. Available: https://developers.google.com/safe-browsing/
[33] Ports, Kali tools : Urlcrazy (2014). [Online]. Available: http://tools.kali.org/information-gathering/urlcrazy
[34] PhishTank, Phishtank: Join the fight against phishing (n.d.). [Online]. Available: https://www.phishtank.com/index.php
[35] Katie, Search help : Refine web searches (2017). [Online]. Available: https://support.google.com/websearch/answer/2466433?visit id=0-636224813165786320-3307756866&p=adv pages similar&hl=en&rd=1
[36] Google Developer, Google safe browsing : What is safe browsing ? (n.d.). [Online]. Available: https://developers.google.com/safe-browsing/
[37] JsonWhoIs, n.d. (n.d.). [Online]. Available: https://jsonwhois.com/
[38] Alexa, Alexa (2017). [Online]. Available: http://www.alexa.com/