ABSTRACT

Unlike a random, run-of-the-mill website infection, in a strategic web attack, the adversary carefully chooses the target frequently visited by an organization or a group of individuals to compromise, for the purpose of gaining a step closer to the organization or collecting information from the group. This type of attacks, called “watering hole”, have been increasingly utilized by APT actors to get into the internal networks of big companies and government agencies or monitor politically oriented groups. With its importance, little has been done so far to understand how the attack works, not to mention any concrete step to counter this threat.

In this paper, we report our first step toward better understanding this emerging threat, through systematically discovering and analyzing new watering hole instances and attack campaigns. This was made possible by a carefully designed methodology, which repeatedly monitors a large number of potentially watering hole targets to detect unusual changes that could be indicative of strategic compromises. Running this system on the HTTP traffic generated from visits to 61K websites for over 5 years, we are able to discover and confirm 17 watering holes and 6 campaigns never reported before. Given so far there are merely 29 watering holes reported by blogs and technical reports, the findings we made contribute to the research on this attack vector, by adding 59% more attack instances and information about how they work to the public knowledge.

Analyzing the new watering holes allows us to gain deeper understanding of these attacks, such as repeated compromises of political websites, their long lifetimes, unique evasion strategy (leveraging other compromised sites to serve attack payloads) and new exploit techniques (no malware delivery, web only information gathering). Also, our study brings to light interesting new observations, including the discovery of a recent JSONP attack on an NGO website that has been widely reported and apparently forced the attack to stop.

1. INTRODUCTION

Consider that you are viewing your favorite restaurant’s menu, as you always did in countless prior visits. This time, however, the menu stealthily drops a piece of malware on your office computer, which later silently collects your personal and business information and further propagates across your company’s internal network. It turns out that the malware was actually strategically planted there for the purpose of infiltrating your company, due to the popularity of the restaurant among the company’s employees. This is an example of targeted infiltrations, which aim at one specific organization, population group or industry of high value. They are the first step of an Advanced Persistent Threat (APT) [34, 45], a continuous, multi-stage and stealthy hacking process for such serious purposes as international espionage, sabotage, intellectual property theft and domestic surveillance, etc. In the past few years, APT attacks have led to the breach of critical national infrastructures [29] and the computing systems of leading companies [48], news agencies [53] and political dissidents [55]. According to various published reports (e.g., [34, 45]), such an attack involves several phases, among which the most important are the steps (reconnaissance and delivery) that get adversaries a foothold into the target system. This is often achieved through social engineering, such as spear phishing [39, 42] and increasingly strategic website compromising1, dubbed “watering hole attacks” [29], as described in the above example.

Strategic site targeting. Simply put, in a watering hole attack, the adversary carefully selects a set of websites frequently visited by his targets and by compromising these sites gains opportunities to penetrate the targets’ systems, in a way much like the predator lurking around a watering hole to wait for its prey to show up. Since the selected websites are typically trusted by the target, such an attack is often very effective, as pointed out by a Symantec report: “Any manufacturers who are in the defense supply chain need to be wary of attacks emanating from subsidiaries, business partners, and associated companies, as they may have been compromised and used as a stepping-stone to the truly intended target” [54]. Examples include the compromise of the Council on Foreign Relation website for attacking other agencies [37], the infection on forbes.com that targets the defense industry [53] and the use of a restaurant menu to get into an oil company’s network [59].

Defense against watering hole attacks is challenging. These attacks often involve zero-day exploits, malware or other unique techniques, which are hard to detect. Limiting access to all such frequently visited sites is not a viable solution, given the inconvenience it would bring in. Without scalable techniques, monitor-
ing these sites for malicious activities is difficult, due to their large number (e.g., 120K sites we found to be visited for at least 10 times within 8 months from a company’s traffic traces). Most importantly, given the fact that such targeted attacks only aim at a small set of carefully selected targets, and therefore are much less frequent [46, 39, 42] and stealthier than random compromises, so far only 29 watering hole instances have been reported and made public. In the absence of adequate real-world attack data, our understanding of this emerging threat is very limited, making it hard to come up with any effective response.

**Finding watering holes.** In this paper, we report the first systematic study on watering holes, making an important step toward better understanding such elusive but significant threat, through discovery of new strategic compromises and in-depth analysis of these cases. This was achieved using a new methodology which helped find new watering holes by continuously analyzing the network traffic triggered during the visit to a set of likely target websites. At the center of the methodology is Eyeson, a system that performs a lightweight persistent surveillance of the likely targets by inspecting only the headers of the HTTP requests to the site. From such thin information, Eyeson automatically builds a model for the rate of the change (based upon quantified features such as hostnames, content types, URL patterns, file names, etc.) observed across different visits to the same site. Once a sudden big change occurs (discovered from HTTP headers), which are supposed to be unlikely to happen according to the model, the site is flagged and goes through other analyses to determine if it is indeed compromised.

The simplicity and efficiency of the methodology enable continuous monitoring and analysis of a large number of potential targets of watering hole attacks. In our research, we ran Eyeson on 61K websites (selected from a company’s internal traffic logs), and 178 other likely targets. For each targeted website, we collected its HTTP traffic, leveraging archive.org, from 2010 to 2015 to analyze their change rates. Additionally, we performed continuous real time crawling of a smaller list of targets. As a result of the study, 30 possible strategic compromises were sent to our industry partner for validation and in 3 months, 17 sites among them and 6 campaigns never reported before have so far been confirmed. Note that confirming watering holes is extremely complicated and time-consuming, requiring resources, experience and cyber intelligence to infer the adversary’s intentions. As an example for the complexity of this task, iphonedevsdk.com is reported to be compromised twice: one is strategic and the other is not (Section 3.3). Given the difficulty in finding and validating this new type of targeted attacks, so far only 29 watering hole cases were made public (through blogs [48, 53, 22, 55], technical reports [30, 29], etc.). Our study brings to light over 59% more attack instances and 6 new campaigns, a significant contribution to the effort of understanding and mitigating this emerging threat.

**Our discoveries.** Indeed, our study has already led to new insights into how such elusive attacks work. As a prominent example, we found that RSF-chinese.org, a Chinese NGO, was strategically compromised and implanted with malicious JavaScript code that exploits the JSONP vulnerabilities within leading Chinese websites like baidu.com to collect identity information from the visitor, such as her email address, name, etc. The attack technique here has never been reported before and is important in a sense that it can defeat the protection of popular privacy enhancing technologies like TOR provided to the visitors of politically sensitive sites. This finding has been confirmed by an AV vendor and received intensive media coverage [23, 58, 51, 64]. Also found in our study are other high profile political sites such as boxun.com, which was compromised multiple times, the Carter Center, cartercenter.com and other politically oriented watering hole campaigns. For the attacks on enterprises, we observed new instances and interesting strategies. For example, an India shipyard’s website was found to be strategically compromised and a new watering hole running the ScanBox framework [22] was found to exhibit targeting behavior never reported before, e.g., delivering malicious payload to website visitors from a computing center in San Diego but not from a university.

These new findings help us better understand such strategic compromises. Particularly, our study presents strong evidence that the watering hole perpetrators not only aim at a specific organization but also at a group of politically oriented people. Further, the repeated compromises of political sites demonstrate that they are relatively soft targets compared with the corporate websites that are better protected and rarely exploited multiple times. Also, we found that watering holes are characterized by a long lifetime, in some cases, a few years, and unique evasion tricks, which hide their attack payloads, redirection scripts etc. on legitimate (but compromised) or legitimate looking third party domains, or even utilize these domains as command and control (C&C) centers. In terms of attack techniques, it is surprising to see that the adversary may not deliver any malware or compromise the victim’s system at all: sometimes, all they do is just collect information through the victim’s browser (e.g., the JSONP watering hole).

**Contributions.** The contributions of the paper are outlined below:

- **New understanding of the watering hole attacks.** We conducted the first systematic study on strategic compromises leading to the discovery of high impact watering hole attacks never reported before. Compared with the state of the art, these findings enable us to gain a better understanding about the motivations, targets and strategies of the APT actors. Now we know how politically oriented sites are targeted and exploited and how unique strategies were deployed to collect information and spread infections. This is critical for developing effective responses and will inspire follow up work on the emerging threat.

- **New methodology.** Such a new understanding was gained through analyzing a set of new attack instances and campaigns our research contributes to the APT research community. These instances were discovered by a lightweight methodology that only inspects the headers of HTTP requests, which is therefore capable of continuously monitoring a large number of potential targets to track down the changes occurring there. Although Eyeson, at its current state, was only used to find us more watering hole instances, the system has the potential to be deployed in a corporate environment as a pre-filtering mechanism, after proper improvement and evaluation.

## 2. BACKGROUND

Here, we present overview about the strategic attacks on websites, and outline the adversarial model used in our research.

**Strategically compromised websites.** For an APT actor attempting to infiltrate an organizational network, one effective approach is to compromise the site frequently visited by the employees and use it as an infection vector to disseminate malware. The details of this type of website compromise were first revealed by RSA FirstWatch in 2012, which reported a “VOHO” campaign aiming at business and government agencies in certain geographic areas [29]. In the campaign, the adversary infected carefully selected sites (e.g., Massachusetts Bank [10]) and planted a malicious JavaScript there. The compromised site checked whether a visiting system was running Windows and a specific version of Internet Explorer, and if so, redirected the browser to torontocurling.com, also a compromised site,
using an iframe to exploit the browser and install a remote access Trojan (RAT) called Gh0st [57]. This attack first used a zero-day vulnerability in Microsoft XML Core Services, and then switched to a known Java vulnerability.

More recently, a few more watering hole cases are reported [53, 38, 30, 31]. A prominent example is the attack on Forbes.com, in which malware infection was hiding inside the “Thought of the Day” Flash widget that automatically shows up once the website is visited. Those running vulnerable browsers were then automatically infected. The true targets of the attack appeared to be senior executives and professionals in major corporations, as indicated by Invincea [53], which further indicated that its customers in the defense industrial base were particularly targeted by the malware.

In all these attacks, the adversary apparently gathered information to identify the frequently visited websites trusted by the target organization, and also carefully selected those less protected, involving different types of vulnerabilities. Due to the stealthiness of these attacks and their targeting nature, only a very small set of watering hole cases were made public: only 29 so far. In addition, the technical details of the attacks were often not fully revealed. To understand and mitigate this threat, it is necessary to find more attack instances and collect more information regarding attack techniques.

Adversarial model. We consider an adversary trying to exploit trusted popular external sites to infiltrate a target. The adversary we studied in this case was able to acquire high profile information about the target, might leverage advanced techniques (e.g., zero-day vulnerabilities) and invest a lot of effort in orchestrating the attack. They can perform a number of actions such as arbitrarily changing the content of the website, redirecting traffic to other sites under their control, monitoring visitors to the website etc. However, they can not completely hide the traces of their campaigns, as the malicious payload has to be embedded in the web page of compromised sites, which could be observed by the network monitors deployed by the organization, web crawlers operated by security companies or security experts inevitably.

3. METHODOLOGY

A watering hole attack aims at the external websites frequently visited by the victim or her organization, exploiting their long established relation to find an easy avenue for infiltrating the target. On the other hand, such a relation can also be leveraged to improve the chance to find strategic compromises: after all, looking back at the long history of interacting with a familiar site, an individual or an organization has a pretty good idea about how it is supposed to behave, and how it evolves over time, even though the amount of observable information for each visit is limited. The key idea of Eyeson is to continuously monitor these strategic sites in a lightweight manner, model how fast and significantly they change (i.e., the observable difference between consecutive visits) based upon the long history of clean communication, and utilize the model to capture the changes considered to be rare. This simple change rate model, built on only the headers of HTTP requests, enabled us to inspect the evolution of a large number of selected sites in our research, and led to discovery of new watering holes.

In this section, we elaborate the methodology used in our study for finding watering holes, particularly the design of Eyeson Profiler (Section 3.1) utilizing a small number of attributes available in HTTP headers. Next, we build a list of strategic websites that are potential targets of watering hole attacks and collect their corresponding HTTP traffic to monitor and profile (Section 3.2). Finally, we evaluate Eyeson Profiler on a set of labeled watering holes and execute it on a larger set of monitored traffic to find 17 new instances (Section 3.3).

3.1 Profiling

Eyeson profiler performs continuous monitoring on the HTTP traffic generated by website visits to identify suspicious changes. This approach is based upon the observation that visiting a website multiple times in a short period of time rarely results in different HTTP requests, due to the fact that websites evolve gradually over time, which makes them appear static within a short time frame. Most frequently, what causes new HTTP requests is just real time contextual behavior such as display of different advertisements according to a visitor’s meta data (e.g. browsing location and time of day). On the other hand, when a target website is compromised, a visit to the website will generate a relatively new set of HTTP requests, which are quite different from those observed before. Such requests can point to new resource URLs on the same monitored domain and/or a new domain or sub-domain. Additionally, a compromise may change the resources already in use on the monitored site, e.g., modification of JavaScript libraries (e.g, jQuery) or injection of an iFrame [43].

As an example, by examining the history of http://cgdev.org, a water holed US think tank, from Jan to Oct 2014, we found that most content the website serves includes html pages, CSS files and some images, comes from the same domain. In the meantime, it also generates a number of HTTP requests to popular third party services such as fonts.net and googleapis.com. However on the 16th and 17th of Oct, 2014, we observed a compromise leading to a new set of HTTP requests, such as http://news.foundationssl.com/i/p.php and http://news.foundationssl.com/i/s.php?seed=[ ]&alivetime= [ ]&r=[ ]. These types of changes, new FQDN or new resources, are frequently observed in the HTTP traffic when legitimate domains are compromised, as reported by prior investigations [44, 49, 41]. Also, once the visitor’s system also gets infected, additional HTTP requests may show up, for the purposes such as downloading malicious executables or jar files.

To capture such sudden changes, which helps identify a compromised website, Eyeson first builds a profile from the website’s clean history. Secondly, for subsequent visits to the target, they are compared to the built profile and a visit change rate is calculated. If the change is considered to be acceptable, it will be added to the profile. Otherwise, an alert is reported.

Profile building. A profile is used to keep track of a target’s HTTP traffic history and its change rates. The history here is a collection of features extracted from HTTP headers and their values, including URLs, FQDN, sub-domain, content types, URL patterns per sub-domain and file names per sub-domain. For example, for the FQDN feature, we keep all domain names that appear in clean visits, and for the content type feature we keep all distinct content types the web site serves. An example of the profile for http://cgdev.org is shown in Table 1. Those features are selected in our research because they are known to be associated with the behavior of compromised sites: the adversary could create a sub-domain under the compromised domain for his malicious activity (e.g., [56]) and often add new types of files to be downloaded by the visitor (e.g., the executable never seen from the site); also new URLs almost always need to be generated to deliver malicious payloads and sometimes, even URL patterns (that is, the domain and path without the values for arguments) never seen before show up.

To construct a profile and analyze an observed change, Eyeson first removes the dynamic content brought in by legitimate ad networks and third-party services, since the content served by them varies frequently which introduces noise to our monitoring process. For this purpose, we utilizes a set of whitelists to filter out the requests related to legitimate ad networks and tracker networks
After the advertisement, tracker networks and popular domains are removed, the HTTP header information can now serve the purpose of profile construction and update. To this end, Eyeson uses a few clean visits (collected from the long history of interactions with the target) to set up the initial traffic history, filling in different features (URLs, content types, etc.) in the profile. Once the history is there, it waits for a few additional visits so as to construct a model for the change rate. Specifically, for each new visit to the target, all HTTP requests involved are inspected one by one. For each request, our system compares its feature values, such as sub-domains, URLs, Content type, URL patterns, etc. (see Table 1 for a complete list) with those in the profile. For each feature we count the number of new values that are not included in the profile, as the feature’s change rate. We aggregate the value of all features’ change rates into a score denoting the overall visit change rate.

For example, consider a request for the URL http://www.cgdev.org/cgd_stats/12989?oo5sv3bltb=, which is not included in the URL feature of the profile (Table 1), but matches one URL pattern once the value oo5sv3bltb is removed. In this case, the URL change rate is 1, but all other features’ rates are 0, resulting in a change score of 1.

More specifically, let us assume that we use $m$ features $F_1, \ldots, F_m$ and have built a profile from $n - 1$ visits. At the $n$-th visit, feature $F_j$’s rate of change is the number of new values observed in feature $F_j$ and is denoted by $N_j$. The visit’s overall change score is $R_n = \sum_{j=1}^{m} N_j$.

Based on the observed change rates $R_1, \ldots, R_n$ for $n$ continuous visits stored in the profile, a probabilistic model $P_n$ is constructed for representing the expected distribution for visits’ change rates (i.e., the probabilities $P[R_n \leq k]$ are explicitly stored for all integer values of $k$). Additionally, given the sample size (the number of visits in a profile), confidence intervals are also built at different confidence levels (e.g., 95%). Upon a new visit, the profile, change distribution and confidence intervals are continuously updated if the visit is not labeled as potentially malicious.

Change point analysis. Using the learned probability model $P_n$, Eyeson looks for abnormal rates of change in subsequent visits, i.e., outliers with respect to the historical distribution of change rates. We use a simple outlier detection method for this purpose [20]: If at visit $n + 1$, the rate of change $R_{n+1}$ falls outside the confidence interval for a certain confidence level (set at $L$), the visit is labeled as an outlier and an alert is generated. Otherwise, the rate of change $R_{n+1}$ is used to update the profile and the change probability distribution to $P_{n+1}$. This approach is very intuitive: we just determine a range of change rates that are observed most of time (e.g., over 95% of the visits); when a new visit causes a lot of changes, with a rate going beyond the range in the confidence interval, it is captured as an outlier and reported for further post-processing. If the visit is later cleared, the rate is added to the profile, causing an adjustment to the distribution and the confidence interval.

As an example, for the profile in Table 1, Eyeson set its 95% confidence interval to (0.94,1.3) in our model. When it comes to the visit to http://www.cgdev.org/cgd_stats/12989?oo5sv3bltb=, the change rate 1 is inside the interval and therefore considered to be acceptable and added to the profile. On the other hand, another visit to http://cgdev.org at the time the site was compromised, as we observed from archive.org, generated a lot of requests to the domain foundationssl.com and caused downloads of the types of files never seen before. The change rate in this case was calculated as 39, which is clearly an outlier and as such flagged by the system.

With its simplicity, this change-point analysis turns out to be quite effective. In our study, using the real data collected from a large organization and the traffic related to known watering holes, we found that this simple approach indeed helps us discover a set of watering holes never known before (Section 3.3).

3.2 Data Collection

In this section we describe the datasets collected by Eyeson. We start by building a list of potential watering hole targets to monitor. Next, we collect HTTP traffic of the targeted domains and other complementary datasets. Finally, we describe our process to generate a ground truth set of labeled visits to watering holes. Strategic target selection. In our research, we used two main sources for target selection. First, based on several sources of intelligence from our industry partners, we collected a list of 178 web sites considered to be high profile targets of watering hole attacks. On the list are governmental, political, defense contractor sites and 29 other strategic websites that have been water holed in the past (see Table 3) such as think tanks and SW engineering systems.

Our second source for target selection is the traffic of the organization which we aim at protecting from these attacks. If a nation state actor is interested in the proprietary information of a certain company, they will first attempt to do reconnaissance on the company to come up with targets such as business partners, subsidiaries, close by commercial businesses (e.g. restaurants and banks), forums and other websites related to the company’s industry sector. These potential targets should be screened by Eye-son and can be identified from the company’s HTTP traffic. In particular, we select a list of websites that are visited frequently by the company’s employees from Jan 1st to Sept 4th 2014, after removing extremely popular sites that have in general good security practices and would be more difficult to compromise, e.g. windowsupdate.com, resulting in a list of 121,473 sites that were visited at least 10 times every month. These sites are the ones that have a long standing relationship with that particular organization, and can be leveraged by attackers as potential targets since they are in general not as well protected as the extremely popular sites (e.g., those in the top Alexa ranking).

HTTP traffic Collection. Eyeson profiler is designed to find compromised websites by analyzing HTTP requests generated from a website’s visit. In an enterprise setting, such HTTP headers are usually collected through a common network product (namely a
web proxy system) without requiring additional data to be collected. In our research, however, we aim to evaluate the potential effectiveness of Eyeson profiler before deployment to an enterprise setting, such an evaluation requires both HTTP headers and responses to validate our results.

To this end we evaluate Eyeson on a much larger set of HTTP traffic by leveraging archive.org, a system that implements a dynamic crawler to crawl a list of URLs intermittently and maintains a snapshot for each visited URL. When an archive URL is triggered through a browser, any embedded requests in the snapshot are rendered, essentially recreating the visit to the URL at the date when the snapshot was taken providing us with both HTTP headers and responses. Searching the archive.org for the target list of 121,651 FQDNs, we collected over 1 million archive URLs. A detailed description of our archive data collection process along with an example of an archived visit is provided in Appendix B.

In addition to such archived HTTP traffic, we conducted our own real time monitoring of the manual list of 178 websites by crawling them with a dynamic crawler through the anonymity channel TOR [12], using 17 User Agents representing popular browsers and operating systems such as Internet Explorer, Chrome and Firefox on Windows, Linux and Android. In total, archive and real time crawling resulted in 14 million collected snapshots. Table 2 illustrates the HTTP data sets collected to monitor and screen 61K FQDNs over a period of 5.4 years.

Complementary data sets. We also collected a number of other data sets for different purposes as shown in Table 2. Whitelists were used by Eyeson to remove noise from the visits introduced by advertisement and tracker networks. A challenge here is that since the collected HTTP traffic set goes back a few years, the latest copy of Easylist [52] does not include the advertisement networks active several years ago but out of service now. In order to include those advertisement networks, we took advantage of the archives again as they also collected the snapshots of those lists. Further, we generated a list of the most popular sub-domains (i.e. indicative of third party services) from the same enterprise set used to select the domains to be monitored. Such popular sub-domains were determined by the number of visits they receive, at least 10K visits per day for those that ended up on our list. At the top of it are crl.microsoft.com & fast.fonts.net.

Ground truth. Ground truth for targeted attacks is hard to come by, because they happen rarely. Even though we can gather a few confirmed watering hole domains from various sources, it is still challenging to determine the period of time when they were infected. In our research, we manually gathered from technical reports a set of confirmed watering holes (shown in Table 3) and the Indicators of compromise (IoCs) for each case (as described in those prior reports) such as specific strings in malicious payloads (e.g. function names), URL pattern, malicious domain name and approximate compromise date. Using the IoCs and their compromised dates, we searched the collected archived HTTP traffic and located the snapshots with the IoCs and further manually verified the presence of infections there. For each compromised snapshot, we further took its snapshots 20 days before and after the infected one as the site’s clean versions, if they did not carry any IoC and also passed the sanity check performed by a content based anti virus system, MS SE [47]. This gave us a ground truth set with 14 watering hole FQDNs corresponding to 69 monitored URLs (aka, profiles). The total number of snapshots in the set is 23,532, each corresponding to a visit, including 1,682 compromised snapshots.

2Throughout this paper, a monitored URL refers to the start URL from which a visit starts.

3.3 Eyeson Evaluation and Results

Over the domains and traffic collected, we used Eyeson to identify a small set of FQDNs highly likely to be watering holes. Here we report our findings.

Evaluation on the ground truth. Using the collected ground truth set (Section 3.2), we bootstrapped the profile for each monitored URL with 10 visits (after initial 5 visits, the follow-up 5 for collecting change rates). We found that the 95% confidence interval yielded a false positives rate 19.4% and a zero false negatives. This demonstrates that our simple approach has the potential to be used a pre-filtering system, though a more extensive study on a larger dataset is needed.

Evaluation on the whole dataset. Then we ran Eyeson on all collected HTTP traffic covering 5.4 years (Table 2), a total of 14M snapshots for 133,527 monitored URLs, with a clean profile built per monitored URL. In our experiment, we bootstrapped the profile for each monitored URL with 10 visits (after initial 5 visits, the follow-up 5 for collecting change rates), and ran Eyeson over all the snapshots gathered in the order of their dates. The profile change rate series was reset at the beginning of every calendar year or whenever a time gap over 3 months was found (due to the missing data in the archive). We utilized a 95% confidence interval to scan the collected HTTP traffic based on the timeline of the snapshots (from both the archive and the real time visits).

Altogether, Eyeson detected 1.7M snapshots (aka, visits) whose changes were considered to be significant, detailed in Appendix A and Table 10. These changes corresponded to 2.7M URLs (i.e. embedded links) hosted on 17.6K FQDNs (not the FQDNs under monitoring) and 456 static IP addresses.

Eyeson, at its current state, is used as a measurement methodology to find more watering hole instances but is not ready as a full blown organizational pre-filter. That being said, Eyeson does have the potential to work as a pre-filtering system given the proper modifications and organization level evaluations. Looking at the toxicity levels (i.e. fraction of confirmed alerts), Eyeson outperforms Evilseed [35] where Eyeson has a 3.4%-7.4% confirmed malicious domains vs 1.12% by Evilseed. A closer look at Eyeson 2.0 is provided in Appendix A.

Finding compromised websites. As the system was run on 5.4 years of traffic, a large number of visits were detected as suspicious. Some changes were false positives which is attributed to either legitimate changes (such as upgrade of website design and templates), ad related URLs that were not covered by the white lists (particularly when these scripts were hosted on the monitored domain itself, e.g. OpenX ad platform) or malformed URLs. Also, the content related to many of the detected change URLs was missing from the archives making their legitimacy difficult to verify.

To screen out the outputs for new watering holes, we first utilized a multi-step validation process to confirm that a subset of reported visits were indeed compromised. This validation process includes blacklist cross matching, URLs and content scanning with VirusTotal [62] and lastly automated labeling and manual analysis.

Specifically, we first cross matched the detected changes with a number of blacklists (Table 2). Since the HTTP traffic we collected is often related to website snapshots that go back a few years, the compromises involving those sites may have already been detected and reported to blacklists. Therefore, to find malicious (or compromised) domains we collected the history of several blacklists (covering suspicious sub-domains, IP addresses and URLs) from the archives whenever available and utilized them to validate detected domains and their snapshots. Also used in our validation
was CleanMX virus watch [13], which maintains a large number of URLs. Many of them are legitimate domains that were compromised for short periods of time. To avoid false positives when using CleanMX lists, we linked the blacklisted URLs to the dates they were reported to be compromised before using them for cross matching.

Secondly, we leveraged VirusTotal [62] which provides an API allowing users to scan either URLs or files. Over the output of Eyeson, we scan URLs and files with VirusTotal with the exception of cloud related URLs when their content was available and could be scanned.

In order to maximize the chance to find new watering hole cases, we further considered suspicious those unconfirmed URLs indicating the hosting status as currently parked, currently down, using DDNS or url shortner, cloud related hosting or using static IPs. After that we clustered the new URLs observed according to their patterns (i.e., the tokenized sequence including paths, file names, other parameters without values). We further cluster the detected monitored domains by frequency of changes. Finally, we randomly selected from the clusters suspicious change cases for a manual analysis.

In our study, we were able to confirm many compromised domains in this way, many of which turned to be watering holes as explained next. Note that such a manual step is important as watering holes are rarely caught by blacklists and VirusTotal especially for recent ones (2014 and later).

**Finding new watering holes.** As a result of this three step validation process, we generated a set of confirmed compromised visits corresponding to 3.2K FQDNs, details are shown in Table 10.

After such local validation, we get to the point where we have to answer the most difficult question for our research: how to determine whether any of these confirmed compromises is indeed a watering hole attack? The question is difficult since we need to figure out the perpetrator’s intention, which can only be done by comparing the compromises with other known cases, looking for the attackers’ fingerprints, leveraging various cyber intelligence, and other means not available to the public. Therefore, the only thing we could do is to forward our findings to leading industry agencies that have experiences in watering hole analysis for a cross-agency validation. Note that even for the parties indeed having this capability, the validation process is complicated, painful and time consuming, largely depending on individual analysts’ experiences. As a simple example for the complexity of the issue, the site iphonedevsdk.com was compromised twice, one in 2010 and the other in 2014. The second one was confirmed to be a watering hole attack aiming at Apple and Google employees [48], while the first one turns out to be a random opportunistic compromise in which the attacker placed a Traffic Direction System (TDS) there.

Due to the high cost of this validation process, it is impossible for us to dump all 3.2K FQDNs to our industry partner (which is a leader on watering hole analysis). What we did is to hand-pick a few likely instances from all these confirmed attack cases. For example, we randomly selected a few compromised domains confirmed by VirusTotal or the blacklists if the attacks are marked as “exploit”. Also, from all the manually confirmed domains we picked out those whose suspicious changes happened recently and are therefore more likely to be validated using the data available. In the end, we forwarded 30 FQDNs to our industry partner, which together with other organizations, has analyzed so far 20 of them in the past 3 months, and confirmed 7 new watering hole attacks. Among them is the high profile JSONP attack (Section 4.1), a politically oriented case which was later reported in the media [23, 58, 51, 64].

Furthermore, we searched all the suspicious URLs (for those 3.2K domains), looking for the IoCs of known watering holes (e.g., ScanBox), with the hope to find new instances of known watering hole attacks. As a result, we were able to find an additional 10 new unreported watering holes.

### 4. UNDERSTANDING WATERING HOLES

#### 4.1 Results Overview

**Landscape.** As a result from the previously described manual validation and collaborations with our industry partners we have confirmed 17 new unreported watering holes corresponding to 16 FQDNs, with one domain being strategically compromised twice. Finding as many as 17 watering hole attacks is considered a big win in the research on targeted attacks as they are quite infrequent and stealthy, and therefore rarely found and confirmed. Our new discoveries increase the publicly-available attack instances by 59%.

---

**Table 2:** Eyeson collected data sets which are used for different purposes.

<table>
<thead>
<tr>
<th>Usage</th>
<th>Type</th>
<th>Source</th>
<th>Duration</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring</td>
<td>Archive and real time HTTP traffic</td>
<td>Manual</td>
<td>Jan 1, 2010 - May 9th, 2015</td>
<td>FQDN:178 Monitored URLs:57,006</td>
</tr>
<tr>
<td></td>
<td>Archive HTTP Traffic</td>
<td>Enterprise selected set</td>
<td>Jan, 2012 - Aug, 2014</td>
<td>FQDN:81,616 Monitored URLs:76,821</td>
</tr>
</tbody>
</table>

**Table 3:** 29 Reported watering holes. Other reported watering hole attacks did not name the watering hole website.

---

*Note: Full list of domain names is not included to maintain focus on the key points of the analysis.*
Tables 4 and 5 summarize all discovered new watering holes and their corresponding campaigns. We report two types of discoveries, a new attack and subsequently a new watering hole instance or a new instance for a previously known attack/campaign, e.g. the “ScanBox” campaign. A campaign here refers to a group of attack instances all characterized by a set of Indicators of Compromise (IoCs), such as URL patterns, domain name and compromised date, etc., with the exception of “ScanBox” which cannot be attributed to one attack, as explained later. Further, compromise dates and lifetimes are calculated for both watering holes and campaigns, depending on available data, which can only be considered as a lower bound for the times the attacks lasted since the archive has gaps (missing data) in the collected snapshots.

Overall, almost half of the discovered watering holes are popular None-Government Organizations (NGOs) mostly about human rights and the freedom of speech. These types of websites are prime targets for nation state actors, i.e. government sponsored, targeting a specific niche of people, e.g. dissidents. Such a claim is supported by our findings, shown in Table 4, where many NGO watering holes keep getting compromised multiple times. Moreover, their compromises last for a long time, e.g. 6 months for rsf-chinese.org (until the publicity of our work stopped it).

Of particular interest is one website, cartcenter.org, a human rights organization, which had been strategically compromised at least 5 times starting with the ever so popular “VOHO” campaign [29] in 2012. Two of the watering hole attacks have never been reported before but did not last for long and have been quickly cleaned up. The fifth attack, appeared in late 2014, is not in the table as it does not show up in our dataset but had been reported to us by our industry partner and we found its URLs flagged in the CleanMX virus watch [13].

To find out the reason behind such frequent compromise and long lifetimes, we ran Sucuri [11] to understand the security protection of the web servers used to host the websites. Sucuri [11] is a remote website malware and security scanner that checks the status of a website before a user visits it. We found that two of NGO websites, starred in Table 4, are actually running an Apache server that is two years behind in its updates. The same security issue (without being updated for two years) has also been discovered on two other NGO websites, adpl.org.hk and cfr.org.

In addition to NGO watering holes, we discovered industry specific watering holes targeting employees and clients of certain sectors. Their compromises, however, rarely last for long, just like what has been reported about other watering holes, such as iphone-devsdk.com, forbes.com and anthem.com. Additionally, the “ScanBox” framework is found to be the most prevalent tool utilized by this type of watering holes.

### Used Intermediaries

Oftentimes, compromises contain HTTP requests to external destinations such as exploit servers, C&C, redirectors, etc., which we call intermediaries. In the collective set of confirmed watering holes, we found that the intermediaries used fall in three categories: legitimate domains, malicious domains or DDNS (Dynamic Domain Name System) and URL shorteners. A rising trend that we observe is the use of legitimate domains, which are compromised to host malicious payloads and serve them to visitors of watering holes. Such use of legitimate domains helps in avoiding detection and bypassing security systems, particularly the protection mechanisms employed at enterprises, where newly registered domains are usually a red flag. Additionally, we observed the purchase and use of intermediaries that are similar to the watering-holed domains, e.g. scdusa.com (watering hole) vs. usascd.com (intermediary) & jquerycdn.com (watering hole) vs. jquerycdn.com (intermediary). Further, we checked the PassiveDNS provided by the Security Information Exchange [4] for the traffic received by the legitimate domains around their compromise dates. Specifically, we found that the watering hole procommons.org.hk contained links to the legitimate domain hotel365.co.kr, a Korean hotel ranked 891K by Alexa [21]; the site (hotel365.co.kr) was compromised from August to September 2013 according to our dataset and the PassiveDNS shows that it received 17.15 visits daily during the compromise, well above an average of 5.9 visits when it was clean.

### 4.2 In-depth Analysis of New Cases

Here we report an in-depth analysis on the newly discovered watering hole cases, which helps us better understand the APT actors’ motivations, strategies and techniques. We explore three po-
<table>
<thead>
<tr>
<th>Campaign</th>
<th>Start-End date</th>
<th>Lifetime</th>
<th>Indicators of Compromise (IoCs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>27th Sept, 2014</td>
<td>1 day</td>
<td>get忽略了，因为它的Indicators of Compromise (IoCs)</td>
</tr>
</tbody>
</table>
Table 6: Malicious files flagged by Virustotal as CVE-2014-6332 served through 6 different URLs on 3 watering holes.

<table>
<thead>
<tr>
<th>Watering hole</th>
<th>Malicious URL</th>
<th>Virus total Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>adpl.org.hk</td>
<td>d52c05964a80209f6692609f130778cc11614f952b123b0517107</td>
<td></td>
</tr>
<tr>
<td>hksa.org.hk</td>
<td>d52c05964a80209f6692609f130778cc11614f952b123b0517107</td>
<td></td>
</tr>
<tr>
<td>dlpl.org.hk</td>
<td>5b47ec22b22c2d1c5c68a77c8b7b302f7058b7394501a15dc6a0a0b8d5</td>
<td></td>
</tr>
<tr>
<td>ADPE.org.hk</td>
<td>419f2b61e9458e718f43e816e3e1bc5d523c0a82bdc4c15822b3a0c39</td>
<td></td>
</tr>
<tr>
<td>alp.org.hk</td>
<td>2d01a381899f109ff68f43d84f8437b45454545ae811d2012935f1929b53c201</td>
<td></td>
</tr>
<tr>
<td>youpai.org</td>
<td>32983ac3c680913ed7a7a9e9096dbb59c27191b49c59d6002031c09b4d</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Malicious files flagged by Virustotal as CVE-2014-6332 served through 6 different URLs on 3 watering holes.

The ScanBox Framework.

ScanBox is a framework that has been pervasively used in watering hole attacks. It was first reported by AlienVault on 28th Aug, 2014 [22]. Later on, PWC [30] revealed four attacks using the same framework followed by 20 more attacks [31]. These campaigns were carried out against sites in diverse sectors including energy companies, think tanks, etc. The framework mainly performs deep reconnaissance on its victims and in some cases keylogging, and then sends the collected information back to a C&C Center. More specifically, at the reconnaissance stage, ScanBox shows extensive fingerprinting activities. In addition to fingerprinting the operating system and language used by the visitor, it attempts to detect the existence of security systems installed on the victim’s machine. Flash version, web development tools, networking tools and more. After the reconnaissance stage, ScanBox has been reported to deliver malware to selected targets.

In our dataset, we detected 6 watering hole attacks using Scanbox, 2 of which (peoplepower.hk and cgdev.org) have been reported before while the remaining 4 are considered new discoveries. Using the scripts collected from all 6 watering holes and other analogues. The compromise using bcbridges.org is quite different, even though the attack happened around the same time as the other one. Specifically, it serves a script that attempts to fingerprint the visitor by finding her user agent, exact version of Shockwave and also checking the existence of AV systems. The earliest version of this script was found from the archives in June, 2010\(^3\). Also, the same set of infections were discovered on peacehall.com, apparently from the same perpetrator. Interestingly, the archived snapshots of boxun.com around the compromise period of 2012 show some web pages that are clean and don’t contain any malicious payloads, for example (blog.boxun.com/hero/200807/aige/2_1.shtml), indicating some clean-up effort might have been made but apparently not well executed.

These attack cases present evidence that indeed political websites are major targets of strategic compromises. Actually, they tend to be repeatedly exploited by the politically motivated attackers. A possible explanation here is that most of these sites are run and administered by volunteers and/or small teams, and therefore less protected compared with the targets operated by the IT professionals (e.g. forbes.com), which are rarely compromised twice. Also, our findings indicate a possible weak cleanup effort from the owners of the sites: oftentimes, we found that some web pages were clean while others were not.

---


scripts obtained from PasteBin and VirusTotal [17, 16, 19, 63], we performed an in-depth analysis on both the generated HTTP requests in our dataset and the scripts downloaded when crawling the snapshots and live sites using three user agents: IE, Firefox and Chrome.

We found that ScanBox serves different scripts depending on the user agent, with each script tailored to different agents. Specifically, in the case of IE, the fingerprinting (for identifying OS, language, protection mechanisms, etc.) is extensive, in an attempt to find out many installed applications such as the latest Windows OS updates. Additionally, IE specific reconnaissance scripts are found to look for a long list of both well known AV tools, such as Kaspersky, Norton, Bitdefender, etc. and the AV scanners popular in certain countries, e.g. QuikHeal, AhnLab, and Jiangmin, which are included in the fingerprinting. In the case of Firefox and Chrome, they only exhibit different behavior when fingerprinting the user agent, with each script tailored to different agents. Specifi-

Table 8: Sample of two compromises found on boxun.com.

<table>
<thead>
<tr>
<th>Water Site Hole</th>
<th>C&amp;C Host</th>
<th>Post Requests</th>
<th>UA</th>
<th>Get Requests (file names and arguments)</th>
</tr>
</thead>
<tbody>
<tr>
<td>spaceleaders.com</td>
<td>ntxeye.com</td>
<td>js.php</td>
<td>FF, Chrome</td>
<td>jq.php?v=webhp</td>
</tr>
<tr>
<td>SiteB</td>
<td>5886.com</td>
<td>js.php</td>
<td>-</td>
<td>css.js?RandomNum</td>
</tr>
</tbody>
</table>

Table 9: HTTP requests generated from the ScanBox watering holes. Domains tagged with + indicate new discovered watering holes (i.e. not reported before) and bold text indicates encrypted values. SiteB is a submission of one of ScanBox’s reconnaissance scripts on VirusTotal [63] which does not have the corresponding watering hole.

Figure 2: Similarities in reconnaissance scripts over functionalities (upper matrices), and software tools (lower matrices) across different watering holes through three user agents, Internet Explorer (1), Chrome (2), Firefox (3). Tempit is used here to indicate scripts found in online templates. SiteB is a submission of one of ScanBox’s reconnaissance scripts on VirusTotal [63] which does not have the corresponding watering hole.

on to the C&C centers through GET and POST requests (Table 9).

We found such C&C domains include legitimate domains (e.g. ntxeye.com), malicious domains (e.g. foundations.com) and static IP addresses. DDNS was also used in some ScanBox attacks reported by URLQuery, a free web scanning tool. Further, we found ScanBox encrypts some URL arguments and periodically probe the C&C with status update requests when they are being visited.

Of particular interest here is mss.ca, which was found to be compromised as early as late 2014, according to URLQuery. Our dynamic crawler started crawling the website in Nov, 2014 using university campus IP addresses. The monitoring failed to report any suspicious activities. However, after moving our dynamic crawler to a supercomputing center in San Diego, CA, we were surprised to find ScanBox reconnaissance code in the HTTP traffic. This could indicate that the actors behind the attack might be targeting industry users.

Live infiltration. gokhayrak.com in our dataset was also reported to be a watering hole by a blog [30], which was presumably compromised with a ScanBox framework. In our research, we further performed live monitoring on the domain and found that it rotated its redirections over a number of suspicious third-party domains.
One of them, theguardian.com.tw, was available for purchase and we bought it anonymously in an effort to sink-hole the domain. Specifically, we hosted the domain on the Amazon cloud using an Apache web server to log all its traffic, from June 26 to August 16th, 2015.

By analyzing the collected traffic logs and checking for request referrals, we discovered two more unreported watering holes: ibsahq.org for the International Buddhism Sangha Association, and HNN.hk for a Chinese news agency. These two sites turn out to have similar infections as gobayrak.com, but redirect their visitors to other malicious domains, as indicated by the IoCs in Table 5.

Altogether, the sinkhole collected HTTP traffic for 3 months from around 7K unique IP addresses (i.e. victims), mostly from Turkey, Taiwan and USA.

4.3 Recap

Our research shows that the APT actors are extremely active in the arena of politically oriented websites, repeatedly compromising those sites and utilizing all kinds of techniques (e.g., cookie) to track down the visitors and recover their identities (e.g., through JSONP vulnerabilities). A weakness of such a politically minded attack, however, is its targeting at a more generic audience, i.e., anyone who visits the website, and therefore can be less stealthy than the industry specific attacks, in which the watering hole may unleash its attack code only toward the individuals from a certain organization. Therefore, a web scanner tuned to the unique features of this type of websites could lead to new discoveries and raise the bar for this type of attacks. In the meantime, our research shows that the target sites are often less protected, compared with the industry sites. Enhancing their owners’ security awareness and getting help from professionals will certainly make the attacks less likely to happen.

When it comes to attacks aiming at industry targets, the use of legitimate intermediaries becomes an interesting feature. As mentioned earlier, we consider this trick to be an evasion technique, particularly in a corporate environment where a redirection to an unknown external domain could cause a red flag to be raised. However, what can still be found here could be an unusual relation (e.g., redirection) between two unrelated, though both legitimate domains visited by an organization’s employees. Such a relation, if rarely observed before, could become sufficiently unusual to warrant a close look at the domains involved. Again, new technologies leveraging such observations should be further investigated in the follow up research.

5. RELATED WORK

Understanding web site compromise. In terms of the risk factors associated with web site compromise, a recent study showed that sites using certain content management systems (e.g., WordPress or Joomla) or running particular web servers are at higher risk [61]. Regarding attackers’ objectives for compromising legitimate sites, the study by Moore et al. [49] reveals attackers’ strategy of using search engines to hunt vulnerable sites and compromise them for later use in Phishing campaigns. The work by John et al. [36] and Leontiadis et al. [41] elaborate on the abuse of compromised sites for boosting the search rankings of malicious sites owned by attackers. The attackers’ behavior after compromising legitimate sites is also thoroughly studied by Canali et al. [26]. Interestingly, it has been shown that many web hosting providers are not responding in a timely manner to compromise [27] and thus attackers can leverage a compromised site for a long time for criminal activities.

Outlier detection. To redirect visitors to malicious sites or directly drop malware on visitors’ machines, the attacker has to manipulate the web content delivered to the site visitors, introducing inevitable changes. Our system identifies such changes and highlights the ones that represent outliers with respect to the observed historical distribution of the web site structure. Different techniques can be used for this purpose [20, 25, 50]. We use a simple outlier detection method based on probabilistic models and confidence intervals applied to the site’s rate-of-change that proves to be effective in capturing the types of site changes we are interested in.

Pre-filtering systems. EvilSeed [35] is a pre-filter that works by generating search queries for identifying other web pages similar to the known malicious ones. Eyeson, however, does not need a seed of malicious pages and is able to find unknown ones. Additionally, Eyeson outperforms Evilseed in toxicity, later discussed in Appendix A. Prophiler [28] and Delta [24] are content-based pre-filtering systems that look into a combination of static features or changes in page structures to determine the pages owned or compromised by attackers. These approaches are more heavyweight and can be evaded if page content is obfuscated. On the contrary, Eyeson is built to profile the evolution of a target by only looking at lightweight features from its HTTP headers, which enables us to inspect targets at a large scale.

Advanced Persistent Threats. APTs are well-funded and carefully orchestrated targeted campaigns posing serious risks to various commercial and governmental organizations, naturally attracting attention from both the security industry and academic community. The existing work mainly focuses on dissecting APT campaigns [46, 39, 42, 60]. In addition, several mechanisms have been developed to assess the threat of targeted malware [33] and link different attacks [40]. The campaigns investigated by these studies leverage spear-phishing emails to infiltrate the victim organizations, while our study looks into another venue becoming more popular among malicious actors, watering hole attacks.

6. CONCLUSION

Our work contributes towards the understanding and mitigation of an emerging infection vector, strategic website site compromise, increasingly used for delivering malware in initial stages of a targeted campaign. By analyzing over 5 years of data from archive.org and carefully labeling ground truth using public sources, we discovered 17 watering holes never reported before, including a high impact politically minded attack, and shed new light on APT actors’ motivations, strategies and techniques. Looking forward, we believe that our new findings will inspire the follow up research on this emerging type of targeted attacks. Further study is also expected to enhance our methodology Eyeson, exploring the potential of running it as a pre-filtering system for organizations under the APT threat.

Acknowledgements

We thank our reviewers for their insightful comments. This work was supported in part by National Science Foundation under grant CNS-1223477, CNS-1223495, CNS-1527141 and CNS-1408874. Part of the work was done during Sumayah Alrwais’s internship at RSA. We thank Kent Backman from RSA FirstWatch and Todd Leetham from EMC CIRT team for their generous help in providing the list of compromised sites and investigations. We are grateful to Ronald L. Rivest, Kevin Bowers and Robin Norris for their feedback and suggestions. We also thank Xiaoru Pan for his help with the investigations and analysis of watering holes. Any opinions, findings, conclusions or recommendations expressed in this paper
Evasion. The set of profile features capture the anomalies in HTTP requests when a site is compromised and visited by the user. Though an attacker can manipulate the traffic to make one feature look legitimate, it is quite difficult for her to maneuver all features at the same time. For example, an attacker exploiting the vulnerability of Java plugin can deliver the payload through previously seen file name or URL pattern, but a new content type .jar will be inevitably observed from the traffic and the attack will be detected by Eyeson.

However, as the profile is based solely on the HTTP header, the system will only detect a compromise if it leads to changes in the HTTP requests. For example, if the malicious payload is injected into an existing website resource such as a home page, the change to the page will not be captured. But when that change triggers another new HTTP request to an uncommon destination, it will be detected. Even though for almost all the cases we are aware of, a compromised website does bring in observable changes to the visitor’s HTTP requests, we acknowledge that an attack could be carefully designed to avoid any significant change to requests. One such example is that the legitimate Adobe Flash file owned by the targeted site is tampered to include drive-by-download code.

We argue that this type of compromise is inelastic for the attacker’s operation and more likely to expose attacker’s traces, since the attacker has to keep presence on the compromised web host if he wants to adjust the payload or pause the attack, which frequently happens during the attack campaign. Besides, the design and implementation of Eyeson can be extended to foil such type of attack. The HTTP responses from the visited site can be profiled separately using features like file size and the anomalies can be identified through the same change point analysis.

In a nutshell. Eyeson’s preliminary evaluation on the archive HTTP traffic shows that the technique is accurate enough to serve a pre-filtering system, though the validation on the whole dataset was too complicated to yield a conclusion. It is important to point out here that the design of Eyeson makes it very suitable for operating under today’s corporate environment. This is because for the subject of a targeted attack, which is typically a large organization, the traffic its network produces is simply too large to be collected and analyzed efficiently. For example, the company we are working with has on average 120K hosts visiting at least 600K external domains every day; logging even part of the HTTP headers generated by the visits takes 662 gigabytes of storage space daily. As a result, only a small amount of information for each visit can be logged. Eyeson was designed to take advantage of a common organization network product (namely a web proxy system) without requiring additional data to be collected. It can perform persistent monitoring of a large number of strategic websites using only partial HTTP header information such as URL, referrer, content type and cookie. The outputs of the system can be delivered to the incident response team for further evaluation. For this purpose, further research is expected to understand its effectiveness in real corporate settings and enhance the technique with new APT features, including the ones learnt from our study.

B. ARCHIVE DATA

To monitor the changes that happen to a website over a long period of time, we leveraged archive.org, a system that implements a dynamic crawler to crawl a list of URLs intermittently and maintains the snapshot for each visited URL. To visit an archived snapshot on archive.org, one can render the corresponding archive URL in a browser which in turns renders the archived visit and all of its embedded archived URLs at the time the snapshot was captured by the archive.

For example, to visit a snapshot of forbes.com captured on the 28th Nov, 2014, one can browse the URL http://web.archive.org/web/20141128132335/http://www.forbes.com/ which we refer to as an archive URL. Table 12 shows a sample of the generated
HTTP URLs of the rendered archived visit. Since the rendering of the archived snapshot is through a browser, some dynamic requests might be generated at the time of the snapshot rendering in real time (e.g. the Google Analytics URL shown in the table). These real time URLs can be clearly distinguished from archived URL through the use of the domain archive.org. These real time URLs might be generated at the time of the snapshot rendering of the archived snapshot is through a browser, some dynamic re-

Table 10: Eyeson results; profiling alerts, validated alerts and their corresponding toxicity metric. Eyeson Toxicity levels are compared with EvilSeed [35] where applicable.

<table>
<thead>
<tr>
<th>Original URL</th>
<th>Generalized URL</th>
</tr>
</thead>
</table>

Table 11: URL generalization examples. The above examples show that URL parameters are dropped when found. Also, file names are dropped when a path exists, otherwise file name is kept.

HTTP URLs of the rendered archived visit. Since the rendering of the archived snapshot is through a browser, some dynamic requests might be generated at the time of the snapshot rendering in real time (e.g. the Google Analytics URL shown in the table). These real time URLs can be clearly distinguished from archived URL through the use of the domain archive.org in the URL. Additionally, the example shown in Table 12 is actually a snapshot captured during a watering hole attack, evident from the request to the malicious IP 74.207.254.87. At the time the archive.org attempted to crawl the malicious request, it was taken down already and thus the subsequent malicious requests described in the report [53] were not observed here in the archived visit. Still, the beginning of the compromise chain (i.e. request to http://74.207.254.87) was captured since the Forbes home page was still compromised at the time of archive crawling and thus this request would be detected by Eyeson as a significant change.

Archive Data Collection. We searched the archive.org for the target list of 121,651 FQDNs and collected over 1 million archive URLs. Unfortunately, many of the URLs had only one snapshot but most of them differ only in the URL arguments or file names. For example forum websites such as linuxforums.org have a unique postID for each post where the postID is passed as part of the file name. But all the posts (with different URLs) cause similar HTTP traffic once being visited as they use the same content management system and styling templates (e.g. vbulletin). In order to correlate the visits to those URLs (so the changes on them can be observed from multiple snapshots), we generalized URLs by removing their parameters and file names when the URLs all contain the same paths as illustrated in Table 11. In our study, we collected such generalized archive URLs for 61K FQDNs from the identified potential targets. The remaining domains either never showed up in the archives or had less than 10 snapshots and as such could not be profiled or monitored.

We further implemented a dynamic crawler as a Firefox extension and deployed it to a number of Virtual Machines (VMs). We instrumented our crawlers to crawl the collected list of archive URLs and capture all rendered HTTP requests generated from the visits. In addition to such archive HTTP traffic, we conducted our own real time monitoring of a small number of domains, specifically those in the manually selected list of FQDNs. A summary of the collected archive HTTP traffic is provided in Table 2.

7. REFERENCES

[1] Ad blocking with ad server hostnames and ip addresses. pgl.yoyo.org/as.


[31] DOMAN, C., AND LANCaster, T. Scanbox framework – who’s affected, and
Table 12: Archived snapshot of forbes.com during a water hole attack on it [53]