College of Computing and Engineering
NOVA SOUTHEASTERN UNIVERSITY | NSU Florida

Mario J Lorenzo

Search Poisoning Attacks and Blackhat SEO

Includes a Review of: "Surf: detecting and measuring search poisoning" (Lu, et. al, 2011)

Search poisoning is a deceptive technique used by blackhat SEO, where malicious sites exploit search engine ranking by presenting web pages that appear relevant to popular keywords with the goal of having their web page ranked high within search results. Once an unsuspecting search user clicks on the result and is directed to a landing page where they are quickly redirected to a series of other web pages before arriving at a terminal page. Once they have arrived at the terminal page, the user may be presented with a scam soliciting personal information or an option to download fake anti-virus or other malware. To accomplish this, an adversary may hijack a compromised web site to leverage a legitimate web site's reputation rank to boost the web page rank within a result. Another method is to conditionally serve web pages based on whether the requestor is a web crawler or someone referred from a search result. This allows the adversary to present a web crawler with a page that appears relevant to the keywords it targets but serve a different malicious page to a visitor who clicked on a search result and arrived at the landing page. The technique then redirects the user on to other web pages depending on whether there are affiliates involved in the search poisoning campaign. Eventually the user arrives at the malicious terminal page where they are presented with malware or other scams.

According to a source cited by Lu et al (2011), 70% of visitors reach a website through a search engine result. If true, this means a critical avenue for connecting users to websites requires searching, identifying the relevant search result, and clicking on the result to proceed to the intended web site. Search poisoning directly targets this critical avenue by diverting traffic to malicious sites and potentially exposing unsuspecting users to harmful malware or other schemes used to steal their identify or defraud them. Understanding this threat vector and studying the techniques, mitigations, and pervasiveness of search poisoning is important in maintain a safe environment for web surfers. During a 7-month study, the authors discovered that 50% of search keywords were poisoned. This alarming trend was observed to increase over the period of the study with evidence of sophistication and effectiveness by adversaries able to target holidays and current events.

There are various technical challenges involved in detecting search poisoning. Many of these challenges involves an implicit trust that web crawlers have when indexing a web site's content. This trust is violated by malicious web sites when they present to the web crawler a deceiving web page that appears to be highly relevant to targeted keywords. This tricks the web crawler to associate those keywords with the web page and promotes that web page in search results. This attack is technically difficult to detect by a search engine unless it attempts to masquerade as a browser and compares the web pages returned to look for inconsistency. This is computationally expensive to do at ultra-high scales required by search engines. But this also may be evaded by the adversary that look for known IP addresses used by search engines allow

Another challenge involves the dynamic and volatile use of domain names and IP addresses by the search poisoning campaign used to evade traditional blacklist security scanners that seek to identify known malicious websites. According to the authors, this evasion tactic typically involves using a landing page that may have been compromised to preserve a good reputation in order to boost its search result ranking and then immediately redirects a user when they visit the site from a search result onto other intermediaries before arriving at a terminal web page that presents the malicious content or scam to the user. In their 7-month study, the authors demonstrated the known malicious web site scanning tools failed to identify a malicious terminal page 78% of the time.

The growing popularity of this attack vector and the lack of mitigation together make this an important research topic for investigation.

Counter Measures and Detection Methods

The authors made several contributions including raising awareness of the dangers and growing trends of an attach vector they termed as "search poisoning". They provided valuable insight into search poisoning attacks by conducting a large and long study that spanned 7 months collecting data and statistics of suspected search poisoning instances. They studied the partitioned the dataset to reduce bias during their feature analysis phase of their work to understand the key variables (or predictive features) that appeared to be intuitively informative to a binary detection classifier. They then trained and evaluated a model using C4.5 decision tree classifier using a rigorous 10-fold cross validation mechanism and reported the average True-Positive-Rate (also known as Recall or Sensitivity) and False-Positive-Rates. They achieved a decent detection rate accuracy that they tuned the model using an ROC curve analysis (TPR vs FPR) by optimizing for a low False-Positive rate of .004 with a corresponding True-Positive-Rate of .9358. Since their work is considered first-of-a-kind there are no other benchmarks to compare the performance of the SURF classifier. But given other indirect comparisons such as the poor detection rate of existing malware scanners on terminal web pages, failing to detect 78.9% of malicious pages, their work can be considered a positive contribution to the inception of the Search Poisoning research field. There work also conducted valuable experiments to understand the feature robustness and model generality of their method through additional 3-fold evaluation experiments. These findings can serve as a foundation for future work in this area.

Using Redirection Graphs for Topology Discovery

Using redirection graphs as a modeling technique for tracking the site redirects that occurs when a search user clicks on a search result can help reveal the network and characteristics of the search poisoning campaign and help with detection and countermeasures in mitigating its impact.

A redirection graph can be defined formally as a directed acyclic graph (DAG) as follows:

> **G** = (**P**, **R**), where **P** is the set of all web pages visited and **R** is the redirection between web pages. Each web page *p* visited is a member of **P** ($p \epsilon \boldsymbol{P}$) and a redirection is defined as (p_1, p_2) ϵ **R** and $\{p_1, p_2\} \epsilon \boldsymbol{P}$

The redirection graph structure is an effective way to model the topology and connectivity of web page linking and therefore a valid approach for representing the organizational structure of search poisoning campaigns helping to identify and discover patterns useful for mitigating the threat. Since the scope of this work focused on the visited pages that are visited from a search result as well as the redirection between pages, a graph is a logical and valid structure to use.

Using Machine Learning Classifier

A key component to the SURF tool and its methodology for detection (Lu et al, 2011) employed the use of the J48 Weka decision tree classifier which is an implementation of the well-known C4.5 decision tree classifier. The authors state that decision trees are an efficient classifier for training and testing when compared to other classifiers. A well-known and defining characteristic of decision trees in general is the ability to inspect how the features and thresholds are defined within the decision tree. This yields valuable information about the importance and predictive value of each feature within respect to the trained model. Most other machine learning classifiers are opaque and difficult to extract insights from the model itself and instead require other feature analysis methods to be used in order to understand the importance of a given feature. One known drawback of using a decision tree is the danger of overfitting the training data, such as representing every feature value threshold from the training data as a separate decision point within the tree. Typically algorithms, like C4.5, apply a two-phase learning approach where the tree is grown and then later pruned to regularize and therefore yield a more generalizable model to unseen real-world data.

There exist many other classifiers that could be considered for this kind of problem including ensembles of decision trees such Random Forrest or Kernel methods such as SVM or presently Neural Networks that are effective against non-linearly separable data. Other classical methods include Logistic Regression classifier or Naive Bayes statistical classifier. All these methods can be quickly evaluated and compared against the C4.5 decision tree algorithm used to identify the most effective ML algorithm for this task.

Lack of Training Data

When evaluating a classification problem that SURF addresses, there needs to be a representative dataset that serves as ground truth (i.e. gold standard) that can be used as a control in

comparing the results from the classifier, in this case SURF. Because this problem was not previously studied, there was no known ground truth dataset established for this classification task. Because the authors selected the use of a machine learning decision tree classifier as the method for classification, a training dataset that includes the correct label (or class) for each input instance is required in order to fit the model to the training data.

The authors used what they described as a semi-manual method for collecting data using a cluster of browsers that would collect search results for popular keywords over a period of time. This data was then partitioned into two groups, a study group and an evaluation group. They conducted feature analysis over one set of data to identify the features used for training the decision tree and a separate set for training and evaluating the data using a known 10-fold cross validation technique.

An area of concern with the quality of the data generated involves the semi-manual curation of ground truth data which was done using some heuristic methods. The authors examined terminal pages that were deemed not malicious because they were absent in existing blacklists or because the website had a "fair" reputation. Using this decision criteria, they back tracked through the redirection graphs and labeled search results as poisoned or non-poisoned depending on the classification of the terminal pages in the redirection graph that was collected over a time period.

Unfortunately, there is potential for misclassification and bias to be introduced depending on how "fair reputation" is defined. There is also the high possibility of misclassification in cases where websites have been already compromised and were unknowingly serving malicious terminal pages. This means there is a potential for a malicious terminal page being labeled as non-poisonous if its not previously known (not in the blacklist) and hosted within a website that has a "fair" reputation.

Feature Analysis

The authors of SURF performed an extensive feature analysis based on initial feature identification that was based on intuition of the task definition using a set of collected data. The selected feature set was broken up into 3 feature groups that the authors describe as Redirection Composition, Chained Webpages, and Poisoning Resistance. The following list enumerates the features and a brief description of each:

- Total redirection hops – the number of redirections that occurs after click on the search result before arriving at the terminal page.
- Cross-site redirection hops – the number of redirections that occur across second-level domain boundaries
- Redirection consistency – identifies whether a redirection is conditional depending on whether the referrer is a search result
- Landing to terminal distance (topology / country distance) – distance measured by topological and geographical between IP address of the landing page and the terminal page
- Page load/render errors – whether a page yielded an error loading
- IP-to-name ratio – the ratio of URLs that contain IP addresses (vs domain names) throughout the total number of direction hops
- Keyword poison resistance – measures the average reputation score of the top 10 search result websites
- Search rank – the rank position of the candidate search result
- Good rank confidence – is the ratio of keyword poison resistance to the search rank (it's based on the previous two features)

Each of these features has a logical or intuitive rationale for providing some informative perspective on the overall classification task. These are the 9 features that the authors arrived from a total of 15 initial features. They conducted feature robustness tests to understand that impacts of an adversary evading a specific feature and what the impact on the overall model accuracy would be. They also identified the two dominant features of their model as the "redirection consistency" and "landing to terminal distance" which aligns with the intuition of those being a key aspect of carrying out the search poisoning campaign by deceiving the web crawler of the true intent of the landing page and then redirecting the

search user to another location, possibly through other intermediaries.

Some additional changes to the features could examining the addition of a “page load times”, based on the intuition that many of these attacks originate from low latency network environment sometimes hosted in a residence with low bandwidth and transfer speeds. Some features may exhibit multi-collinearity such as those that have inter-dependence like the “good rank confidence” on “search rank” and “keyword poison resistance” that may be redundant in its predictive value to the model and could be eliminated.

Overall, the selected features, provide information about the search result presented by the search engine or the redirection path of the landing page with respect to the terminal page and therefore is a good feature set to use.

Accuracy Analysis

The authors used 3 different types of performance evaluations for SURF classifier. The General Accuracy was focused on measuring the models ability to accurate identify a search result, the Generality Tests was used to demonstrate that the model performs well against different types of malicious categories of pages (such as fake AV, Pharma, and Drive-by downloads), and Feature Robustness to demonstrate that the features selected for the model performed well even if some evasion techniques reduced the effectiveness of a given feature. Different adjusted features impacted the model performance differently, but the two dominant features resulted in the largest degradation of model accuracy causing 80 of 100 misclassified samples. The authors justify this as being acceptable since those two dominant features are integral to the attack vector used and evading those features would require a different mechanism with additional costs such as reducing the redirection chain by using the landing page as a terminal page to present malware which could result in rapid blacklisting by the search engine.

The authors used a10-fold cross-validation, which is used as a rigorous method of validating that the model is robust and able to perform well without overfitting a particular data set. Using this method of validation, you train with 90% of the data and hold out 10% for validation, you then cycle the 10% holdouts 10 times until all the data has cycled through the training and validation set. By looking at the average accuracy across all 10 iterations, you gain a better understanding of the expected performance of the model in unseen real-world data.

Micro and Macro Evaluation

The authors present in their “Evaluation Results” (section 3.4) the accuracy of their classifier over an evaluation dataset where they present the True-Positive-Rate (more commonly referred to as Recall in ML analysis) and False-Positive-Rate (or 1 – Specificity). Their evaluation method using cross-validation was thorough, but several questions arise.

Typically, in Machine Learning model benchmarking, it is conventional to present a more complete picture of the accuracy metrics of the model by including F1 score, Precision metrics, and confusion matrix. F1 scores are an effective metric for comparing model performance because it measures both recall and precision accuracy. After applying some derivation, I was able to identify the F1 score as .9649 and .9909 for their corresponding False-Positive-Rates of .004 and .009 respectively. This is a very high F1 score, although there is no other equivalent model for this classification task at the time of their work to compare with and thus this sets the current state-of-the-art at the time.

No model performance was reported in the Empirical analysis section of the paper. The authors used two forms of analysis over a 7-month data collection timeframe that they broke up into 31 epochs. The micro analysis was focused on a 7-day window and the macro analysis was looking at the 31 epochs over the entire timeframe.

The Micro analysis measured “poisoning lag” to study how quickly adversaries can poison a new trending keyword. This analysis yielded several observations including the average time a poisoned landing page was present in a search result as 1.7 days. The authors stipulate that this means the adversaries were conserving the reputation of the landing page for other attacks to avoid blacklisting.

During the Macro analysis, the authors observed an uptick in poison searches with peak around Christmas time holiday and the SuperBowl. They observed that popular keywords were poisoned 50% of the time among the top 100 results and 15% of the time in the top 10 results.

The authors surveyed 350 random terminal pages over the 7-month period and manually categorized those terminal pages. They found that fake Anti-virus was the most popular early in the study, but later social engineering and scam pages increased in popularity. No accuracy metrics were reported comparing those manually categorized pages to the classification of the SURF system. This was a missed opportunity to evaluate and compare the model performance results in section 3.4 with the macro empirical evaluation phase of the study.

Future Improvements

The overall quality of the research is good. The authors explained the problem of Search Poisoning and identified goals of their research which were addressed by the methodology they presented for the SURF classifier. They provided a good rationale for the feature selection and performed a rigorous validation of their model using 10-fold cross validation and an additional 3-fold validation for studying the generality of the model beyond the malicious categories of malware.

There are several areas of improvements that could be made to the research and the paper itself. Some trivial improvements include defining the use of lesser known acronyms, such as "AV" (or anti-virus) or including conventional ML model performance metrics such as F1, Precision, and Confusion Matrix. Also providing a location to the dataset used to allow for peer-review and reproduction of the experiment results help improve the credibility of the work.

There are several mentions of subjective and vague terms that are not formally defined in the paper that would help provide a more precise definition for the criteria. For example, "fair reputation" is not defined within the scope of the paper, but rather indirectly through a citation. The term is subjective and ambiguous without a clear definition that serves as the underpinning for much of the semi-manually labeled data used as the ground truth for the performance evaluation of that system. The presented work often relies on citated blacklists that naturally evolve over time and therefore unavailable for reproducing the results of the work presented.

There are some quality issues with the references that are relied on to establish the significance of the research work such as the reference to 70% of web visitors use a search engine to visit a web page. Upon review, this reference did not look to be of any authoritative or academic quality. No authors or specific study was named. Instead, only a date and a website that is no longer available and ironically itself is now redirecting several times to an SEO seller's page. Additionally, the study itself, if it could be trusted, was 5 years old at the time of the research, arguably very stale data given the short existence of the internet and web wide web.

The study makes several unsupported assumptions that are not obvious or proven with any data. The authors claim that only malicious sites use search poisoning techniques because a legitimate company would suffer reputational harm if they misled visitors by luring them into their website using this technique. However, an analogous example can be made using the infamous and aggressive Geico Insurance company marketing campaign that inundates TV and radio with lots of commercials. They use a technique, that one would argue is very similar to search poisoning, by deceiving a viewer into thinking that their commercial is a movie or some breaking news alert only to suddenly switch to their insurance promotion. Applying the same logic provided by the authors would mean that Geico would have suffered reputational damage that would hurt it's sells in the long term, yet Geico is ranked the 2nd largest auto insurance company in the United States (according Consumer Reports website).

Another assumption about only malicious sites using multiple cross-domain redirections, which they represent as a feature in their model, may not always be true. Some legitimate companies, such as IBM, own many business units that have different domains. When initially navigating to a web page, they may redirect multiple times before arriving at the terminal web page. Additionally, companies like IBM use tiny url service to reduce the size of lengthy URL by levering a mapping service that redirects from the tiny url to the target URL. This is also done to count traffic through certain promotional links. Furthermore, IBM performs workload balancing across a global cluster of web server that may redirect you to a distant geographic location, further undermining the distance features used by the model. Given the SURF reliance on these features, it is possible that an IBM search result link could be misclassified as a malicious site.

Another area of concern involves the factor played by the search engine itself when evaluating the performance of SURF. Because SURF uses several features that rely on the Search Engine including the

search rank of a result, it is expected that a Search Engine would rank down nefarious web pages in their results. This means that methods to address this problem of Search Poisoning by the Search Engine could be implicitly improving the results of the SURF system. Additional experiments would need to be conducted to isolate the SURF system independent of a Search Engine to better assess the accuracy of the SURF methodologies.

Lastly, the reliance on semi-manual datasets helps quickly grow the dataset in size, but introduces the possibility of misclassification in the dataset used to train and evaluate the model. An additional blind set of ground truth should be constructed by an independent group to validate the performance of the model against an independent dataset.

10. The above work was later extended in another research effort (Stringhini, et. al, 2013). Read the paper. What are the major critiques regarding the SURF approach mentioned in this publication? How were the critiques addressed in (Stringhini, et. al, 2013)? Do you agree or not? If not, what are the alternative approaches?

Recent Work

Recent work by Stringhini et al (2013), called SpiderWeb, expanded on the SURF by presenting a system for detecting malicious web pages. Their approach is not limited to search engine results but rather any link presented to a web user through social media, news article, forums, or search engine. As such, they designed a system that does not depend on search engine information such as web site reputation or keyword ranks. In their critique of SURF, (Stringhini et al, 2013) consider these limitations and weaknesses of the system when applied to the broader task of detecting malicious web pages given a link. In their critique, they also point out that their system does not depend on the actual web page content, unlike SURF that performs consistency checks of the web page as a feature to their classifier. (Stringhini et al, 2013) also point out the computational inefficiency of visiting links twice in order to obtain a consistency feature score and consider this a detractor that can slow down web surfing experience.

(Stringhini et al, 2013) implemented their own version of the SURF classifier using most, but not all, the features defined by the SURF authors. In their results they point to the very poor F1 score of SURF when compared to the SpiderWeb classifier (Stringhini et al, 2013). They recognize that the evaluation of SURF may not be fair given the differences in evaluation tasks and the absence of certain features that are not available when operating outside of a search engine context. (Stringhini et al, 2013) did not include consistency and page load error features in their implementations of the model. They did note that there were some malicious web pages that only SURF could identify, while others that only SpiderWeb could identify. They also reported that their implementation of SURF achieved a very good False-Positive-Rate (0%) but a very high False-Negative-Rate (81%).

(Stringhini et al, 2013) addressed these critiques by designing a classifier that was based on the redirection graph and extracted features that did not require other contextual information. Their classifier only required the information of the URL, IP addresses, and the pages visited to construct their redirection chain graph which was used as the input for an SVM classifier based on Weka's implementation. They also aggregate information collected from a different user to help generate and contribute redirect chain graphs. Because they operate outside the context of a search engine, they do not make use of search rank or reputation rank information, nor do they extract feature from the content of the web page itself.

(Stringhini et al, 2013) also developed a ground truth dataset by leveraging data collected from an anti-virus program that flagged malicious sites and manually vetted the redirect chains to label them as malicious or benign. This data is of higher quality when compared with SURF because it was manually vetted. They also produced a more general solution to the problem of malicious web page attacks by creating a system that can function regardless of where the link was surfaced. These are all strengths of the SpiderWeb classifier.

(Stringhini et al, 2013) preserve a similar assumption to SURF that implicitly assumes redirects across 2nd level domains are malicious. This however can lead to a higher False-Positive-Rates for benign uses of redirects. One counter example to this assumption is the Nova Southeastern (NSU) Library feature that enables access to various journal publications such as ACM and IEEE. When logged into NSU library and a search is initiated using Google

scholar, results are shown with a special link to the full journal article. When that link is clicked from Google scholar results, the page is directed to the journal and then the journal redirects the page back to NSU identify management to validate the user session before redirecting back to a page checks the valid session and finally redirects to the targeted journal article. This pattern of redirection is a common pattern used throughout for achieving Single-Sign-On and granting access to resources across different domains. Systems such as SpiderWeb and SURF are likely to misclassify these scenarios.

Overall, the SpiderWeb system presented by Stringhini et al (2013) provides several good contributions to the subject by defining a higher quality dataset, broadening the scope of the task to fit most scenarios where users are likely to encounter malicious pages, and developing a classifier that does not depend on a search engine or the content of the page.

Another recent proposal by Zhang et al (2016) called VisHunter differs from SURF in several ways. First and foremost, VisHunter focuses on the broader task of classifying any redirection path as malicious or benign and not just from within the context of a Poisoned Search scenario. One key difference between the two system involves the "visibility" of a server. They assert that only a portion of the redirection chain is needed to help classify a web page as malicious or benign, mainly the transition from visible to invisible or what they call the "entrance" to the malicious infrastructure. They reason that malicious organizations want to keep their core servers hidden from public view and instead establish specific entrances to lure unsuspecting users through redirection. By focusing on this characteristic, VisHunter is able to avoid leveraging features that rely on topology-based that require crowd sourced information, such as SpiderWeb, and content-based analysis, such as SURF, that relies on consistency between page visits.

In addition to these differences, VisHunter does not rely on the geolocation of the landing and terminal pages, like SURF does, because they claim it can be easily manipulated. In their evaluation dataset, they showed that 72.47% of benign redirections occurred across 2nd level domains and 39.97% of benign redirections included URLs with domain to IP addresses. In their own implementation of the SURF classifier, they report that SURF missed all malicious redirections in their evaluation set.

This comparison however is not equivalent since VisHunter used an implementation of SURF that only included 3 of the 9 features. Additionally, SURF was focused on the task for detecting Search Poisoning scenarios and not the general problem of detecting malicious page redirects from any link location.

Both VisHunter and SURF used the J48 decision tree classifier to train 12 features that represent location, graph, role, and relation of the visible to invisible portion of the redirect chain. They argue that their features are more resilient than the related systems including SURF. For example, they argue that their location features which leverage Whois, AS, and IP location to identify the physical location of the server is more reliable and less likely to be evaded or manipulated.

As such, if comparing between SURF and VisHunter on a classification task that more generally detects malicious web pages, the VisHunter method appears to be a better approach. However, both SURF and VisHunter have dependencies on the search engine to provide key information represented as features to their respective models. VisHunter uses the search engine in its decision flow to determine the visibility of a server. This involves looking through a couple of whitelists first and if not found, proceeds to conducting a search and inspecting the top100 search results for the existence of the web server, if the server does not appear, it is assumed to be invisible. This can be considered a point of weakness in the solution since it relies on an external system that is subject to change its behavior and therefore impacting the accuracy of the system.

Linguistic-Collisions and Search Poisoning Attacks

The work by Joslin et al (2019) was well written and organized representation of their study. They clearly defined the problem of linguistic collisions and how they can be exploited by blackhat SEO techniques to promote malicious web pages among the top search results for this category of misspelled words. They presented thorough and insightful analysis of the prevalence of search poisoning targeting linguistic collisions with a rate of 1.19% for English words that are not autocorrected by the search engine. Their work also contributed a method for classifying the probability of a linguistic collision misspelling by using a Recurrent Neural

Network (LSTM) trained over a large set of vocabulary words able to detect how likely a word was not going to be autocorrected. Their study sets the foundation for future work to build on their insights and model.

(Joslin et al, 2019) validates SURF in some ways, but also undermines or contradicts some assumptions and assertions presented in the SURF paper. The fact that (Joslin et al, 2019) use Search Positioning as a valid problem to support the significance of their linguistic collision contribution validates the importance of this threat vector that Lu et al (2011) first identified and studied. There are some contradictions that exist such as (Joslin et al, 2019) citing that 70-90% of users click on search results on the first page, while SURF (Lu et al, 2011) include the top 100 search results in their evaluation and therefore increasing the prevalence of the threat to a higher percentage. (Joslin et al, 2019) finds in their study that only 1.19% of English words with linguistic collisions occurred among the first page of search results. Although at scale this may represent a concerning number of potential opportunities for a search user to be presented with a malicious web page, it is a very small number of the total actual searchers across all keywords. One observation is the large discrepancy between the 1.19% of poisoned search results presented by Joslin et al (2019) for the first page of results compared to the 15% of poisoned search results for keywords in the top 10 results (i.e. approximately the first page) reported in the SURF results. This may indicate that search engines are becoming more sophisticated in identifying and mitigating malicious search results or that inconsistent criteria is being used across studies to determine a malicious web page.

There are various trends observed over the evolution of the literature on this topic. One observation is the evolution of the problem addressed over time changing depending on the current-state-of-the-art and the significance of the problem itself. Initially the problem was focused on search poisoning based on the assumptions that most web visitors will visit a webpage via a search engine. The problem was then generalized to handle links presented from different sources including social media which has become a popular platform where users may encounter links. Each study contributed additional insights to the problem and solution domain by building upon the previous work and evaluating their contribution relative to the body of literature for the topic. The use of machine learning was a consistent aspect of the methodology throughout the papers reviewed, and on two occasions leveraged the same classifier algorithm (J48/C4.5) and the most recent paper following ML trends focused on using RNN in their methodology.

Throughout the different works, the reliance and dependency on search engines to support various features to their model varied. SURF and VisHunter relied in different ways on a search engine whereas SpiderWeb did not.

Overall, these papers collectively helped to fill in the body of knowledge on the general topic of detecting malicious web pages by presenting different perspectives and optimizing their method for different objectives.