

# WebCop: Locating Neighborhoods of Malware on the Web

Jack W. Stokes  
Microsoft Research  
One Microsoft Way  
Redmond, WA, 98052  
jstokes@microsoft.com

Reid Andersen, Christian Seifert, Kumar Chellapilla  
Microsoft Search  
One Microsoft Way  
Redmond, WA, 98052  
reidan,chriseif,kumarc@microsoft.com

## Abstract

In this paper, we propose WebCop to identify malicious web pages and neighborhoods of malware on the internet. Using a bottom-up approach, telemetry data from commercial Anti-Malware (AM) clients running on millions of computers first identify *malware distribution sites* hosting malicious executables on the web. Next, traversing hyperlinks in a web graph constructed from a commercial search engine crawler in the reverse direction quickly discovers *malware landing pages* linking to the malware distribution sites. In addition, the malicious distribution sites and web graph are used to identify neighborhoods of malware, locate additional executables distributed on the internet which may be unknown malware and identify false positives in AM signatures. We compare the malicious URLs generated by the proposed method with those found by a commercial, drive-by download approach and show that lists are independent; both methods can be used to identify malware on the internet and help protect end users.

## 1 Introduction

Preventing malware from infecting computers is a critical problem facing computer scientists. Malware is often downloaded by users clicking on email attachments, but more recently, attackers are infecting computers at an alarming rate from malicious executables hosted on the internet. To help discover malware on the web, we propose WebCop: a system for identifying malicious webpages and neighborhoods of malware on the internet. These malware neighborhoods consist of malicious *landing sites* (LSs) and *distribution sites* (DSs) [9] directly connected by hyperlinks to form a subgraph in the internet. A malware distribution site is the location (i.e. URL) of the malicious binary file on a remote server hosting malware, while a malware landing site is a user accessible website that either provides a link to one or more known malicious distribution sites or contains an embed-

ded malware executable. For example, figure 1 shows a simple malware neighborhood consisting of two malicious landing sites and two malware distribution sites. Similar benign neighborhoods also exist for the distribution of legitimate executables. We provide an analysis of all malicious and benign neighborhoods found on the internet in section 4. These malicious landing pages can then be added to a list of malicious URLs and used by an internet browser or search engine to provide warnings of or block access to malicious websites. Once

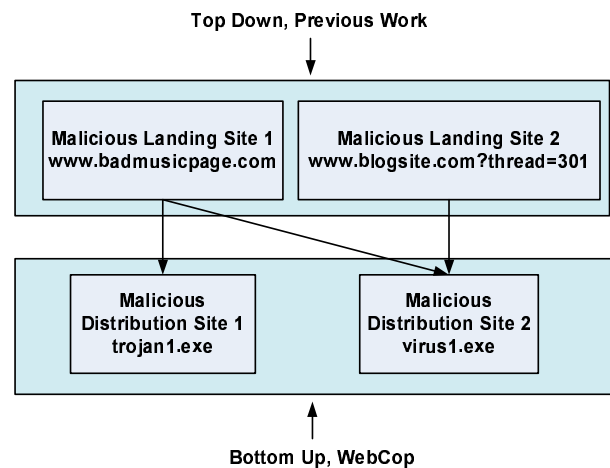


Figure 1: Sample malware neighborhood. Malicious landing site detection can be achieved using a bottom-up or top-down approach.

these neighborhoods are identified, we can use the graph structure to find unknown executables which may be new types of malware being distributed on the internet as well as identify false positives in Anti-Malware (AM) signatures.

WebCop uses a new, bottom-up method for identifying malicious landing sites on the web. We provide an overview of the WebCop system in figure 2. Millions of computers running a commercial anti-malware client

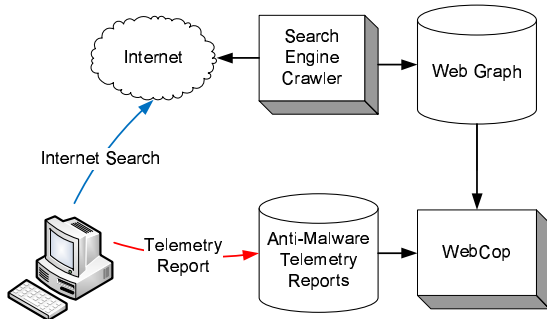


Figure 2: Overview of the WebCop System.

first detect malicious distribution sites when malware is downloaded from the internet. Described in section 2, the AM system sends a telemetry report which includes the URL of the distribution site and file hash if it detects that the executable is malicious or was not signed by a trusted organization.

In parallel, a production search engine crawler constructs a *Web Graph* of the internet where the nodes represent webpages and the edges identify links between webpages. WebCop uses the web graph to determine which landing pages directly link (i.e. via a hyperlink) to the distribution sites identified by the AM system. For the direct links, users click on a hyperlink to download an executable stored on the internet. Modern-day operating systems require the user to give explicit permission before downloading the file. The link determination results from the web crawl are described further in section 3. Referring to figure 1, the WebCop system identifies the landing sites using a bottom-up approach starting with the final destination distribution sites and following the web graph hyperlinks in the reverse direction to identify the higher level landing sites. Although we are primarily interested in discovering malware landing sites and neighborhoods, some of the unknown files are also labeled as benign (i.e. clean) by analysts on the backend thereby indicating benign neighborhoods.

The WebCop approach differs from previous top-down solutions for identifying malicious landing pages. Considering figure 1 again, these top-down strategies rely on first identifying suspicious landing sites at the top of the graph and using a crawler to search for malicious payloads either through direct links to known binaries [6, 14] or more commonly from state changes in a virtual machine (VM) to detect drive-by downloads [15, 10]. A drive-by download occurs when a user visits a website which manages to install a new executable (.exe, .dll, etc.) on the host machine; often this drive-by installation is accomplished simply by visiting the website without user interaction.

The bottom-up approach used in WebCop offers sev-

eral advantages over the top-down methods. For drive-by downloads, identifying the suspicious landing sites to crawl is problematic. Algorithms to find suspicious landing sites have been proposed [10, 12, 13, 2], but attackers can adapt and learn to evade detection. Similarly, detecting state changes in a virtual machine for drive-by downloads can be difficult. Malware will often not run in a virtual machine to avoid detection. Again, attackers can learn to modify their tactics to hide. By starting with the list of URLs hosting known malware that was generated by the AM clients, WebCop only deals with hard classifications using a distributed, targeted detection of the malware executables.

For previous, top-down approaches which involve a crawler and a scanner [6, 14], a very big issue is that downloading all executable binaries from the entire internet and evaluating them with an AM engine is problematic. Today, commercial crawlers are optimized to traverse the internet as fast as possible and do not download executable files found on the web. A list of web-based executables could be generated by the crawler and downloaded off-line, but this strategy includes several delays. The simplest detection method involves computing a unique identifier (UID) for the unknown file (e.g. SHA-1, SHA-256 hash) and comparing the UID to a list of known malware. Simply downloading and computing the UID of, potentially large, unknown files is expensive and time consuming. In addition, this method fails to detect polymorphic variants of existing malware families. Scanning with an AM engine further adds significant delay to the process. To avoid re-imaging the test machine, each new scan must also be run in a VM which introduces the problems noted above. Instead of relying on a centrally located service to analyze the files, WebCop distributes this evaluation to the millions of the individual clients running one of the AM services; end user machines identify new sites hosting malware more quickly than the backend. Furthermore, the malware is run as designed on the native operating system (i.e. not in a VM). Accordingly, the true malicious activities can be detected by the AM service. We discuss additional differences between WebCop and the prior work in section 6.

Results in section 4.3 show that the bottom-up and drive-by download detection methods are complementary. The end result in either case is to construct a list of URLs of malware landing sites; thus lists of URLs discovered from both methods can be combined for better coverage of the web. The main contributions of this paper include providing:

- A large scale evaluation of malicious and benign internet neighborhoods composed of direct links.
- A targeted, bottom-up approach for detecting malware on the internet.

- A new way to detect false positives in an AM service using the internet web graph.
- A new method to discover potential malware.

## 2 Distribution Sites

This section provides an overview of the data used to identify malicious and benign binaries on the web. Telemetry reports are generated by four Microsoft security products including Windows Defender, Microsoft Security Essentials, Windows OneCare, and Forefront Client Security when executables are installed on the computer. Windows Defender is an anti-spyware product which includes signatures for known spyware but does not include signatures for viruses, trojans, etc. Microsoft Security Essentials, Windows OneCare and Forefront Client Security include anti-virus signatures in addition to anti-spyware signatures. These security products will automatically submit reports to Microsoft when a malicious executable or an unknown executable which is not signed by a trusted authority is installed on the computer. In addition to reports generated during normal operations (e.g. installing a new program from a CD, etc.), reports are also transmitted when a user attempts to download and install an executable from the internet. These reports include the hash and the URL of the executable being installed. The various security products have differing policies on opting-out but users are informed of the data collection (URL, hash) during installation in the corresponding privacy statement (e.g. [5]). In this paper, we limit our analysis to executable files, but the method could also be applied to other types of files (e.g. .jpg, .doc, etc.) that could be malicious. We analyze a sample of the most recent one million labeled distribution sites consisting of 837,882 malicious distribution sites and 162,118 benign distribution sites from reports collected through the end of May 2009.

The number of malware distribution sites significantly outweighs the number of benign distribution sites. Reports are not submitted for executables which are installed and signed by a respected authority. Furthermore, AM analysts only try to investigate and label unknown files which are suspected to be malware since AM engines use signatures which detect malware and do not typically include signatures for specific clean files. Thus, the list of labeled distribution sites is strongly biased towards those containing malware.

A unique executable could be downloaded from many different distribution sites on the internet. A malware author may host their executable on many infected servers in order to provide redundancy in case the malicious binary is discovered on one of the servers and deleted. We found 6046 distinct malicious binaries distributed across

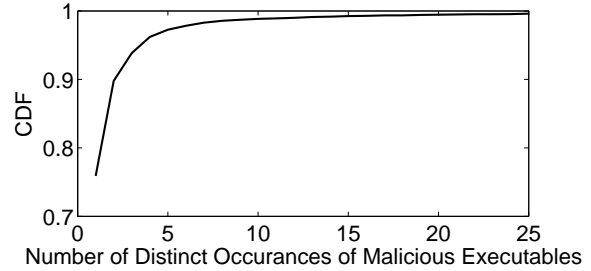


Figure 3: Cumulative distribution function of the number of distinct, malware executable files from the AM reports.

10,853 malware distribution sites. Figure 3 provides the cumulative distribution function (CDF) of the number of distribution sites (up through 25) associated with a single malicious executable. The  $x$ -axis is the number of instances ( $n$ ) a malicious binary is detected on the web, and the  $y$ -axis represents the cumulative distribution function of files found to occur less than or equal to  $n$  number of times. Benign files exhibit a similar CDF. To obtain this plot, we generated a list of all distinct URLs and the corresponding SHA1 hash of all malware executables identified by the AM service. The figure indicates that most malware binaries are only hosted at a few distribution sites on the web. For example, 75.9% (4589/6046) of the malware binaries were encountered only once on the internet while individual malware files which occur 20 or more times account for only 0.6% of the binaries.

## 3 Link Determination

After detecting the malicious and benign distribution sites using the AM services, the next step is to determine which landing sites link to the distribution sites. To do so, we search for all results found by the crawler where the destination URL matches a URL generated by the AM service. The results in table 1 summarize the total number of *intersecting* distribution sites included in both the one million most recent distribution sites identified through May 31, 2009 and the web graph created on June 1, 2009. The table also provides the number of landing sites which link to these distribution sites in the web graph. Of the original 837,882 malicious distribution sites identified by the Anti-Malware telemetry data, 10,853 were also included in the web graph and were linked to by 391,893 malicious landing sites. Likewise, 1,460 benign sites found by the AM data were also in the web graph with links from 2,850,883 benign landing sites. The number of intersecting distribution sites is approximately 6.8 times more for malicious distribution sites than for benign sites. This statistic is an artifact from the AM system since, as mentioned earlier, the

number of benign sites is actually much larger, but AM reports are not generated for appropriately signed executables.

Measure	Count
Number of intersecting benign distribution sites	1,460
Number of intersecting malware distribution sites	10,853
Number of benign landing sites	2,850,883
Number of malware landing sites	391,893

Table 1: Total number of distribution sites found in both the Anti-Malware reports and in the internet crawl. The landing sites which link to the matching distribution sites are also provided.

One question raised by table 1 is why the percentage of intersecting distribution sites found in both the AM telemetry and the web graph is so small. After reviewing the data, we found the reason is because the vast majority of telemetry reports for both malicious and benign distribution sites are only seen within a one month period. For example, only 8.7% of the distinct malware distribution sites observed in May were also reported in April. As a result, most of the URLs for the distribution sites in the AM telemetry are not detected by the AM client or do not exist in the web graph on June 1, 2009. To test how Web-Cop might perform in production, we found 2763 unique malicious distribution sites had links from 158,533 landing pages found in the AM telemetry received only in May 2009. Similarly, 212,688 landing pages contained links to 4633 unique malicious distribution sites in the most recent three months of AM telemetry.

## 4 Malicious Neighborhoods

### 4.1 One Hop Neighborhoods

In this section, we analyze all of the one hop graph data (i.e. connected by a single hyperlink) described above to understand the topology and frequency of the malware and benign neighborhoods. Pages located two or more hops away from the AM distribution sites are considered in the next section. Analyzing the data, we were able to identify the different types of malware neighborhoods shown in figure 4. In the most common “Single Edge” topology in (a), a single landing site links to a single distribution site. In the “Fan-In” layout in (b), multiple landing sites link to a single distribution site and a single landing site has links to multiple distribution sites in a “Fan-Out” subgraph (c). Likewise (d) is a “Complex” graph which contains two or more landing sites and two or more distribution sites. The example in figure 1 is a complex graph; this graph structure provides some redundancy in case a single landing site is discovered and shut down.

Number of subgraphs	Malware	Malware Percentage	Benign	Benign Percentage
Single Edge	2984	46.5 %	263	47.6 %
Fan-In	2498	38.9 %	245	44.4 %
Fan-Out	388	6.0 %	14	2.5 %
Complex	547	8.5 %	30	5.4 %

Table 2: Subgraph topology counts.

The corresponding counts for the topologies are provided for both malware and benign subgraphs in table 2. We first note that the malware neighborhoods far outnumber the benign neighborhoods due to the labeling bias noted earlier, but the subgraph percentages are roughly equivalent. While the benign counts do not reflect the correct distribution in the wild, the counts for the various types of malware neighborhoods do represent the total number found on the web.

We are surprised at the relatively small number of malicious neighborhoods identified by the system. One reason is because the counts only reflect those malware neighborhoods constructed with direct hyperlinks and do not include graphs associated with drive-by downloads. The largest number of subgraphs consist of single edge topologies; the attacker has not provided any redundancy for either the landing site or the distribution site. The number of fan-in, subgraphs far exceeds the count of fan-out and complex subgraphs which is to be expected. The attackers choose to embed hyperlinks in many different landing pages which direct the user to a single instance of the malware. The fan-out and complex topologies are somewhat easier to detect by system administrators; both require at least one landing site to include hyperlinks to two or more distribution sites. Multiple hyperlinks increase the chance that the webpage designer or administrator will identify the malicious links.

Next, we investigate the overall statistics of the Fan-In, Fan-Out, and Complex subgraph topologies. Table 3 provides the median and average number of edges, landing and distribution sites for both malicious and benign neighborhoods. The subgraphs in figure 4 reflect the median, malware topologies (i.e. median number of LS, DS, links) given in table 3. For the case of complex neighborhoods, we have excluded a single, very large malware subgraph and another very large benign subgraph from the computation of the statistics since including these two neighborhoods significantly skewed the “Average” statistics. The table indicates the median counts of the complex malware and benign neighborhoods are very similar while the average counts are 2-3 times larger for the malware subgraphs when compared to the benign subgraphs.

A portion of a malicious, complex subgraph is listed in table 4. The entire subgraph includes 37 landing sites, 33 distribution sites, 814 edges, and 1 distinct executable. From the example, we can identify two separate distribu-

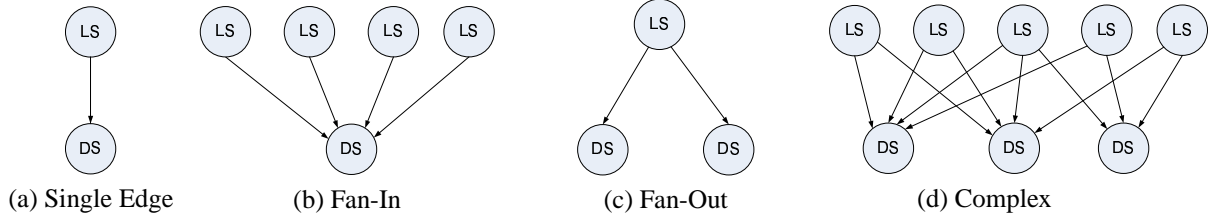


Figure 4: Graph layout for malware and benign neighborhoods consisting of landing sites (LS) and distribution sites (DS).

Landing Site	Distribution Site
http://wwwr.skycn.com/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://www1.skycn.com/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://www.onlinwww.skycn.com/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://www.skycn.net/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://www.skycn.com/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://www2.skycn.com/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://cnc.skycn.com/soft/48428.html	http://cc163.skycn.com/down/easyvideo.zip
http://works.skycn.com/soft/48428.html	http://hdncn1.skycn.com/down/easyvideo.zip
http://crc.skycn.com/soft/48428.html	http://hdncn1.skycn.com/down/easyvideo.zip
http://wwwr.skycn.com/soft/48428.html	http://hdncn1.skycn.com/down/easyvideo.zip
http://tele.skycn.com/soft/48428.html	http://hdncn1.skycn.com/down/easyvideo.zip
http://www.3.skycn.com/soft/48428.html	http://hdncn1.skycn.com/down/easyvideo.zip
http://www1.skycn.com/soft/48428.html	http://hdncn1.skycn.com/down/easyvideo.zip

Table 4: Portion of a complex malware subgraph.

Measure	Topology	Type	Median	Average
Number of Landing Sites	Fan-In	Benign	4	16.1
	Fan-In	Malware	4	31.3
	Complex	Benign	5	17.5
	Complex	Malware	5	33.7
Number of Distribution Sites	Fan-Out	Benign	2	3.5
	Fan-Out	Malware	2	2.9
	Complex	Benign	2	2.4
	Complex	Malware	3	4.9
Number of Edges	Fan-In	Benign	4	16.1
	Fan-In	Malware	4	31.3
	Fan-Out	Benign	2	3.5
	Fan-Out	Malware	2	2.9
	Complex	Benign	8	24.1
	Complex	Malware	11	72.2

Table 3: Subgraph statistics.

tion sites hosting a file determined by analysts to modify the browser. Two landing pages each link to both distribution sites.

## 4.2 Discovering Potential New Malware

As shown in figure 5(a), we next identify unknown distribution sites (UDSs) located two hops away from the malware distribution sites (MDSs) identified by the AM service which should be considered suspicious and may be previously undetected malware. To do so, we locate all destination nodes having the malware landing sites found in the previous section as the source node and the URL contains a binary ending in a wide variety of possible extensions associated with executable files (e.g. .exe, .dll, .zip., etc.). These unknown distribution sites share a landing site with a known malware distribution site. After removing the known, malicious and benign distribu-

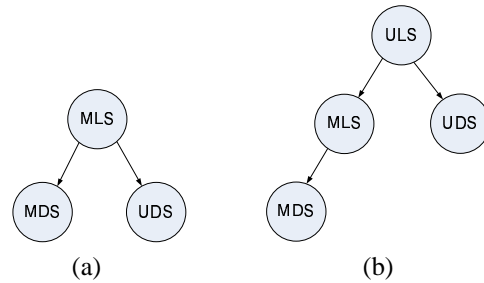


Figure 5: Multi-hop topologies consisting of malware landing sites (MLSs), malware distribution sites (MDSs), unknown landing sites (ULSs) and unknown distribution sites (UDSs).

tion sites from the list, we identified 346,084 unknown distribution sites. This result corresponds to approximately 32 suspicious unknown files for each labeled malware distribution site. The binaries associated with these URLs could be downloaded and scanned by the AM service. If not detected as malicious by the AM scan, these unknown executables should be subjected to more thorough automated analysis such as behavioral monitoring or placed near the top of a ranked list for analysts to investigate. Thus, WebCop can potentially, proactively discover new malware.

Although we did not carry out this investigation, another possibility is to consider landing pages located two or more hops away from the original malicious distribution sites as shown in figure 5(b). While a search engine

can block landing pages linked directly (1-hop) to a malicious distribution site, unknown landing sites (ULSs) located two or more hops away should not be blocked. However, these unknown landing pages can also be used as a starting point to search for additional unknown distribution sites as shown in figure 5(b). All distribution sites found three or more hops away from the original malware distribution site can also be submitted for more in-depth analysis but should be given lower priority compared to the unknown distribution sites located two hops away from the original malicious distribution site (figure 5(a)).

### 4.3 Comparison With Top-Down Methods

In this section we compare the WebCop results with the top-down, drive-by method used by Wang *et al.* [15]. A production scale version of Wang’s drive-by detection system has identified millions of drive-by download sites. Next, we generated a list of URLs detected using the drive-by download method from April 6 through June 1, 2009 and compared the results with the WebCop landing and distribution sites. The comparison revealed two matching distribution sites and no matching landing sites; the matching distribution sites were the payloads for both a drive-by download and a separate hyperlink delivery malware system. Of the two matching distribution sites, the drive-by download system located 212 landing sites. The results are not particularly surprising; if the attackers went to the trouble of creating a drive-by download page, it might be considered a bit of overkill to embed a redundant hyperlink to the malware. This experiment suggests that WebCop is producing lists of malware landing sites which are orthogonal to those generated by the drive-by detection methods, and the lists generated by the two methods can be combined.

### 4.4 HostName Impurity

In the section, we investigate how often attackers create landing sites and distribution sites which share the same hostname. We define the hostname impurity score, based on the entropy, as

$$hi(n) = - \sum_j P(\omega_j) \log_2 P(\omega_j) \quad (1)$$

where  $\omega_j$  is the fraction of landing sites and distribution sites in subgraph  $n$  which share the same hostname. A similar score can also be defined for the domain names. If the hostname impurity score is low, most of the landing sites and distribution sites share a common hostname; the attacker may have set up an exploit server to host both the malicious landing sites and the vulnerabilities. On the other hand, if the hostname impurity score is high, the hostnames vary across the landing and distribution

sites, and the attacker has done a good job of exploiting a large number of servers. A histogram of the hostname impurity score for the complex, malware subgraphs is shown in figure 6. The two large peaks are located at zero and one bits: a large percentage of malicious subgraphs all share the same hostname (i.e.  $hi(n) = 0$  bits) or two hostnames (i.e.  $hi(n) = 1$  bit).

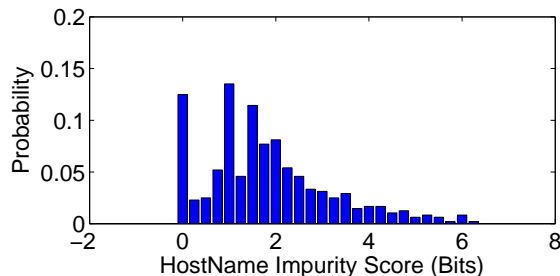


Figure 6: HostName Impurity Score Histogram

### 4.5 Identifying False Positives

The graph structure can provide features which can classify unknown files as either malicious or benign and help analysts identify false positives in the anti-malware signatures. While reviewing the data output from the WebCop system, we found a malicious, fan-in subgraph with hundreds of thousands of landing sites linking to a distribution site which appeared to be legitimate. The in-degree of node  $x$  is the total number of edges where  $x$  is the head. Figure 7 provides the histogram of individual malware and benign distribution sites with the highest in-degrees. The in-degree of the questionable distribution site labeled as malware by the AM signatures is over an order of magnitude higher than the largest in-degree observed for the malware distribution sites. The file did turn out to be a false positive by the AM engine. Similarly, other features of the graph structure (e.g. out-degree, host-name impurity) can be also used to help distinguish malicious and benign files.

## 5 Discussion

In this section, we discuss several issues related to WebCop. Quickly identifying new threats on the internet is critical. There are several issues which affect the time required to identify new threats using WebCop. First, the Anti-Malware engine employed at the client must have signatures or other methods which detect malware downloaded from the internet. Once the telemetry reports are received at the backend, they must be aggregated in a timely manner to be processed by the WebCop algorithm. The recent trends in the AM industry is towards much

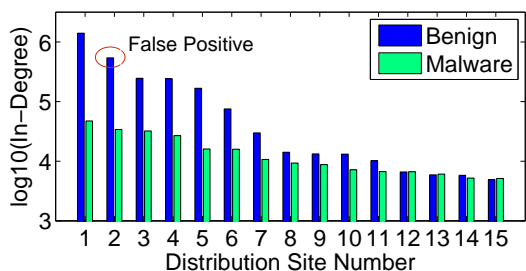


Figure 7: Largest in-degrees for distribution sites.

more frequent and automatic signature updates so early detection should not be an issue for WebCop.

Another potential problem is whether or not the distribution sites and corresponding landing sites were previously crawled by the search engine. Since many of the malicious landing sites are not frequently visited, the search engine must crawl the depths of the internet often. As previously noted, Anti-Malware telemetry reports for malicious distribution sites are usually only received for a short length of time (e.g. less than 30 days). If the crawler takes too long to discover a short-lived distribution site or landing site, WebCop will fail to protect the user. Search engine companies are employing many more computational resources for crawling the internet which should help to alleviate this problem. The web crawler could also be programmed to crawl malicious neighborhoods and high level domains identified by WebCop more often to search for new potential malicious landing and distribution sites.

A third issue related to latency involves determining the landing sites linked to the distribution sites identified by the AM service. This computation involves running a program on a very large cluster hosting the entire web graph. For a production search engine, many other services such as ranking must also be run. In order to put WebCop into production, enough processing resources must be allocated. One way to minimize the computation time is to significantly reduce the number of malicious distributions sites considered. For each landing page in the Web Graph, we must determine if each destination page is contained in a list of known malicious distribution sites. In a production environment, the new malware distribution pages can be included in an incremental list (i.e. daily, hourly) which contains only entries recently identified by the AM service. Engineering effort combined with parallelism on the cluster can help minimize the latency involved with implementing WebCop in production.

Although we have implemented the WebCop system by querying the historical AM telemetry on a monthly basis due to the database schema, the system can easily be run much more frequently. The crawler continuously

traverses the internet to build the web graph. We can query the AM telemetry on much finer granularity (e.g. daily, hourly) and compare to a cached version of the web graph. We envision WebCop being run on a daily basis to help protect end users.

## 6 Related Work

There are two methods malware is downloaded from the internet: installation via a direct link and a drive-by download. Both methods have been intensively studied by the research community. Studies that focus on drive-by downloads identify malicious landing sites through a top-down approach (crawler or search engine integration) with high-interaction client honeypots. These systems visit suspected malicious landing sites with a dedicated – often virtualized – vulnerable system and monitor for unauthorized state changes (e.g. a new file appearing in start up folder) [15, 10, 9, 6, 16, 11].

In [15], Wang *et al.* report on the analysis of a combination of suspected malware sites and the top 10,000 most popular websites. The analysis consists of three stages based on a high-interaction client honeypot technique. The first stage uses a virtual machine running on an unpatched version of Windows XP, followed by various stages that include additional patches. Finally, the last stage uses a virtual machine running on a fully patched version of Windows XP. Using this method, researchers were able to identify unknown attacks. WebCop uses a bottom-up approach based on AM telemetry data.

Alternative methods of detection that do not rely on virtual machines and monitoring the system for unauthorized state changes are being explored as a means to increase efficiency and effectiveness of the detection method [7]. Researchers are also exploring techniques to identify suspicious pages with a light-weight crawling mechanism before sending them to the virtualized system for inspection [10, 12, 13, 2]. Similar techniques could be incorporated into WebCop to augment the web graph with suspicious web pages that may not be included in the graph.

This paper is most closely related to Moshchuk *et al.* [6] who propose a top-down approach using a web crawler (as well as a drive-by download method) to discover spyware and malware. They found that 13.4% of the executables found on the web in May 2005 were spyware as identified by a random-walk crawl and a classification of the binaries using an Anti-Spyware solution. Stammering *et al.* [14] also utilize a top-down crawler approach to identify malware on the web. Compared to Moshchuk *et al.*, they augment their detection techniques to utilize an online database of spyware-related identifiers, signature-based scanners, and behavioral-

based malware detection techniques. WebCop’s detection is initiated based on AM telemetry data. As mentioned earlier, downloading and scanning all executables on the web using a top-down approach is problematic and currently not feasible; WebCop’s use of distributed AM detection alleviates this problem. Furthermore while currently focused on telemetry data collected using signature-based scanners, WebCop is not limited to such. As Anti-Malware software incorporates advanced detection techniques, WebCop will directly benefit from the increased detection coverage.

In addition to the academic papers discussed above, commercial products also identify and can block malicious landing sites on the internet [4, 3, 8, 1].

## 7 Conclusions

In this paper, we have presented a new, bottom-up method to discover malicious webpages linked to malware on the web and the neighborhoods determined by these links. Malware binaries are first identified on the internet using an Anti-Malware service. A crawl of the web is then used to construct a web graph that finds malicious landing sites linked to the malware.

Malware changes very rapidly. Legitimate sites previously tested and deemed safe may be hacked to include links to malware at any point in the future. Malicious websites may be quickly created also putting users at risk. New forms of malware are quickly identified by large-scale, production AM services. Instead of waiting long periods of time for a centrally located service to download and scan unknown files from the web as in top-down methods, suspicious telemetry reports from the AM services running on millions of distributed clients quickly identify new malicious distribution sites. Also, AM services detect new malware from many different sources such as email and other social engineering attack methods. If a new instance of malware is detected from other sources, WebCop can immediately identify new malware landing sites based on the hash of the files associated with the malware distribution sites. What makes WebCop particularly effective is the combination of the small subgraphs identified by the crawler and then labeled by the known distribution sites from the AM service allowing the system to easily discover neighborhoods of malware.

In section 3, we showed that WebCop identified almost 400,000 malicious landing sites on the internet. Furthermore, WebCop also identified approximately 350,000 unknown distribution sites in malware neighborhoods likely to be malicious. While the results given in this study do not approach the scale reported by [9], we have shown that landing sites identified by WebCop show almost no overlap with the large list of known drive-by

landing sites: WebCop is complementary to drive-by download detection systems. This result is to be expected since drive-by downloads involve using an exploit to download malware, and the malware executable is often not accessed from a URL. Given the results included in this paper, we believe WebCop can work well in a commercial system to further protect users from harm.

## References

- [1] GOOGLE. Google safe browsing apis. <http://code.google.com/apis/safebrowsing>.
- [2] LIKARISH, P., JUNG, E., AND JO, I. Obfuscated malicious javascript detection using classification techniques. In *Malware 2009* (Montreal, 2009).
- [3] MCAFEE. Siteadvisor. <http://www.siteadvisor.com>.
- [4] MICROSOFT. Bing search engine. <http://www.bing.com>.
- [5] MICROSOFT. Microsoft security essentials privacy statement. [http://www.microsoft.com/security\\_essentials/privacy.aspx#mainNav](http://www.microsoft.com/security_essentials/privacy.aspx#mainNav).
- [6] MOSHCHUK, A., BRAGIN, T., GRIBBLE, S., AND LEVY, H. A crawler-based study of spyware on the web. In *Proceedings of the 2006 Network and Distributed System Security Symposium (NDSS06)* (2006), pp. 29–40.
- [7] NAZARIO, J., AND HOLZ, T. As the net churns: Fast-flux botnet observations. In *International Conference on Malicious and Unwanted Software (MALWARE)* (2008).
- [8] NORTON. Norton safeweb. <http://safeweb.norton.com>.
- [9] PROVOS, N., MAVROMMATIS, P., RAJAB, M., AND MONROSE, F. All your iframes point to us. In *17th USENIX Security Symposium* (2008), pp. 1–15.
- [10] PROVOS, N., MCNAMEE, D., MAVROMMATIS, P., WANG, K., AND MODADUGU, N. The ghost in the browser: Analysis of web-based malware. In *Proceedings of HotBots’07* (2007).
- [11] SEIFERT, C., DELWADIA, V., KOMISARCZUK, P., STIRLING, D., AND WELCH, I. Measurement Study on Malicious Web Servers in the .nz Domain. In *Proceedings of the 2008 ACM symposium on Applied computing* (2009), pp. 8–25.
- [12] SEIFERT, C., KOMISARCZUK, P., AND WELCH, I. Identification of malicious web pages with static heuristics. In *Australian Telecommunication Networks and Applications Conference* (Adelaide, 2008), IEEE.
- [13] SPOOR, R., KJIEWSKI, P., AND OVERES, C. The honeyspider network: Fighting client-side threats. In *First* (Vancouver, 2008).
- [14] STAMMINGER, A., KRUEGEL, C., VIGNA, G., AND KIRDA, E. Automated spyware collection and analysis. In *12th Information Security Conference* (Pisa, 2009), pp. 202–217.
- [15] WANG, Y.-M., BECK, D., JIANG, X., ROUSSEV, R., VERBOWSKI, C., CHEN, S., AND KING, S. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In *Proceedings of the 2006 Network and Distributed System Security Symposium (NDSS06)* (2006), pp. 35–49.
- [16] ZHUGE, J., HOLZ, T., SONG, C., GUO, J., HAN, X., AND ZOU, W. Studying Malicious Websites and the Underground Economy on the Chinese Web). In *Managing Information Risk and the Economics of Security* (2009), pp. 1–20.