A New Service for Increasing the Effectiveness of Network Address Blacklists*

Jian Zhang[1], Phillip Porras[1], Johannes Ullrich[2]

(1) SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

(2) SANS Technology Institute
8120 Woodmont Avenue Suite 205
Bethesda, MD 20814

Abstract

We recently established a new experimental Internet service for creating customized source address blacklists for DShield.org contributors. This new service utilizes a radically different approach to blacklist formulation that we refer to as Highly Predictive Blacklists *(HPB). A highly predictive blacklist is a list of malicious Internet addresses that is formulated through an analysis of the roughly 30 million firewall log entries that are contributed to the DShield repository each day from across the Internet. The HPB service employs a link analysis algorithm similar to the Google PageRank scheme to cross-compare contributors in search of overlaps among the attackers they report. The attacker addresses included within an HPB are selected by favoring the inclusion of those addresses that have been encountered by contributors who share degrees of overlap with the target HPB owner. Our experiments show that highly predictive blacklist entries consistently yield filters that are exercised at higher rates than those from conventional blacklist methods. In addition, this increase in blacklist filter "hit rates" can last multiple days into the future. In this paper, we provide an overview of our algorithm and present our usage experiences. We discuss the envisioned benefits that we believe HPBs can provide toward reducing unwanted communications for those networks that utilize this service.*

1 Introduction

Source address blacklisting is a well-established method for preventing undesirable network traffic from entering one's network. A source address blacklist represents a collection of IP addresses that have been deemed undesirable, typically where those included addresses have been involved in some previous illicit activity. Such blacklists are often converted into filtering logic, fortifying the port-based policies of a firewall with malicious address blocks to be ignored in their entirety. To date, two common methods for formulating address-based blacklists have become well established across the Internet: the Local Worst Offender List (LWOL) and the Global Worst Offender List (GWOL).

The LWOL-based blacklisting strategy is an inherently reactive technique, which asserts filters against network addresses that have been seen to flood, probe, or conduct intrusion attempts against local network assets. For example, a local network may produce an LWOL to capture those addresses that are the most prolific producers of unwanted incoming packets, or that have been detected as significant attack sources from local IDS logs. LWOLs have the property of capturing repeat offenders that are indeed more likely to return to the site in the future ([9] presents experiments that demonstrate this assertion). Unfortunately, while an LWOL can be effective in reducing unwanted traffic, by definition it cannot include an address until that address has demonstrated significant hostility or has saturated the local network with unwanted traffic.

The GWOL-based blacklisting strategy addresses the inherent reactiveness of LWOL strategies by extending the observation pool of malicious source detectors. A GWOL attempts to capture and share a consensus picture from many collaborating sites of the worst sources of unwanted network traffic. For example, sites such as DShield.org compile blacklists of the most prolific attack sources and regularly post firewall parsable filters of these addresses to help the Internet community fight back [6]. With approximately 1700 contributing sources providing a daily perspective of malicious activities across the Internet, DShield's daily GWOL provides a regularly updated snapshot of those Class C subnets that are among the bane of the Internet with respect to unwanted traffic. Unlike LWOLs, GWOLs have the potential to inform a local network of highly prolific attackers, even when those attackers have not (yet) been seen by the network. Unfortunately, the GWOL strategy also has measurable limitations. For example, GWOLs often provide subscribers with a list of addresses that will simply never be encountered at their local sites [9]. Second, GWOLs may miss certain significant attackers that prefer to choose their targets more strategically, focusing on the known vulnerable networks [2].

*This material is based upon work supported through the U.S. Army Research Office (ARO) under the Cyber-TA Research Grant No.W911NF-06-1-0316.

In this paper, we introduce the Highly Predictive Blacklist service, which is now integrated into the DShield.org portal [10]. The HPB service employs a radically different framework to blacklist formulation than that of contemporary blacklist formulation strategies. Our objective is to construct a customized blacklist per DShield repository contributor that reflects the most probable set of addresses that may attack the contributor in the near future. Researchers have recently observed that there are long-lived attack correlations between DShield contributors; that is, some contributors share quite a few common attackers. These correlations are independent of address proximity [4]. Here, we exploit this observation in a probabilistic inference scheme, in which future-attack probabilities are estimated based on the previous observations of DShield log contributors and the attack sources that have been reported by those contributors.

In formulating an HPB for a given DShield contributor, we assign each attack source address within the repository a rank score. The score uses historical attack patterns to assess a source's probability to attack the target HPB owner in the future. Each DShield contributor is provided a custom HPB that consists of the sources with the highest scores. In essence, this ranking score is derived not by considering how many contributors the source has attacked in the past (which is the case in formulating the worst offender list), but rather by considering *which* contributors it has attacked.

The HPB framework also employs another technique to estimate a source's attack probability even when it has been observed by only a few contributors. This technique models the contributors and their correlation relationship as a graph. The initial attack probability derived from the evidence (the few attacks reported) gets propagated within this graph and the ranking score is then inferred using the propagated probability. To compute this probability, we employ a random walk procedure similar to the link analysis algorithm that is well known as the Google PageRank scheme [1].

Our preliminary experiments, based on a corpus of more than 600M DShield records collected from 1088 DShield contributors[1] show that for most contributors (more than 90%), HPB entries exhibit higher hit counts over a multiday prediction window than contemporary blacklist methods [9]. In the best case, one HPB of length 200 successfully predicted 195 attacks in comparison to only two addresses from the GWOL. The results also suggest that HPB's performance is consistent over time, and its advantage remains stable across various list lengths and prediction window sizes.

The main contribution of this paper is the presentation of our new experimental DShield service [10], which is now available for public use. We provide an overview of the HPB algorithm and its integration into the DShield Internet portal. If successful, our HPB service has the potential to offer yet more incentive for network operators to participate in collaborative sharing schemes, such as those embodied in [7, 3, 8]. We discuss the implementation and use of this service, and explore example experiences in constructing blacklists from DShield contributor data. We discuss the anticipated benefits of this service in helping individual networks address the issue of reducing unwanted network traffic, and conclude with a brief discussion of future features we plan to explore for this service.

2 High Predictive Blacklisting in a Nutshell

Intuitively, the objective of the HPB method is to use the target patterns of attackers stored within the repository from the recent past to construct a probabilistic estimate of which attackers are most likely to visit a given data contributor in the near future. That is, we seek a ranking scheme that tends to favor the inclusion of a source address in a given contributor's blacklist, if that source address has recently been observed by other peer contributors who share a historical attacker overlap with the target blacklist owner.

We view a window of recent attack activities as an attack table. (We discuss our deployed window sizes in Section 3.) As a simple example, Table 1 depicts a collection of attack reports. The rows of the table represent attack sources and the columns represent contributors (victims). An "*" in the table cell indicates that the corresponding source has reportedly attacked the corresponding contributor (victim).

	v_1	v_2	v_3	v_4	v_5
s_1	*	*			
s_2				*	
s_3	*	*	*		
s_4		*	*		*
s_5			*		*
s_6				*	
s_7					*

Table 1. Attack Table

As an intuitive illustration of how HPB estimates attack probabilities, consider ranking some of the sources for victim v_1. v_1 shares the largest number of common attackers with v_2. Since s_4 has been observed by v_2, it would be given a favored ranking score. Next, v_1 and v_2 also share overlapped attackers with v_3, which has itself encountered attacker s_5. s_5 would therefore also be

[1] While DShield maintains a larger contributor set that fluctuates up to 1700 contributors, this analysis considered only those contributors who consistently contributed a minimum threshold of alerts throughout the duration of the experiment.

slightly favored. Finally, while v_1 shares no overlap attacker with any other victim, v_2 and v_3 who are correlated with v_1 do share an overlapped attacker with v_5. Therefore, s_7 which attacked v_5 would be given a non-zero score. Effectively, this favoring scheme forms a transitive closure of overlap relations.

The HPB algorithm employs a framework that formalize the above ranking intuition. In particular, we model the correlation relationship between contributors as a *correlation graph*. The correlation graph is a weighted directed graph. The nodes in the graph are the contributors. (We denote by m the number of nodes/contributors in the graph.) There is an edge from node i to node j if contributor i is correlated with contributor j. The weight on the edge is proportional to the strength of their correlation.

We refer to the unique source addresses who have been reported within the security logs of the contributors as *attackers*. (We observe that innocuous addresses are also often logged. In Section 5, we discuss metrics that may help to reduce these cases.)[2] Each attacker can be associated with a vector $\mathbf{x} = \{x_1, x_2 \ldots x_m\}$ where $x_i = 1$ if contributor i reported this attacker, and $x_i = 0$ otherwise. We call such a vector the *feature vector* of this attacker. Given an attacker that exhibits activities captured by $\mathbf{x}$, we would like to obtain a ranking score that reflect its probability of future attacks. We infer this probability in a very simple fashion: the reported attacks captured in the feature vector form an evidence trail, each suggesting that the source may attack in the future. The degree to which each piece of evidence (each attack) may contribute to the source's attack probability on contributor i is proportional to the weight of the edge leading from the contributor who reported that attack to contributor i. Let $\mathbf{W}$ be the transpose of the adjacency matrix of the correlation graph, the ranking score can then be computed by $\mathbf{W} \cdot \mathbf{x}$. ($\mathbf{W} \cdot \mathbf{x}$ is a vector whose i-th entry gives the source's ranking score with respect to contributor i.)

One may use more sophisticated techniques such as naive Bayesian estimation, least square regression or maximum likelihood regression [5] to construct a matrix better than $\mathbf{W}$. However, there is another issue in the above ranking framework that requires more attention. We note that for an emerging attack source, we can claim at best an incomplete set of observations of the attacker's activities. While an attack source may attack a large set of contributors over time, during its initial activity it may only be reported by a few contributors. The corresponding feature vectors therefore contain only partial information. An ideal blacklisting strategy should provide an ability to incorporate "well-timed" filters, which do not require saturation from an attacker before including it in the blacklist.

Toward this end, even if an attacker has not been observed by contributor i, we may want to set the corresponding entry x_i in its feature vector to a value that reflects contributor i's anticipation of being attacked by this source. Note that our ranking score (after proper scaling) can be used for this purpose. Therefore, we update and obtain a new feature vector $\mathbf{x}(1)$ by the equation: $\mathbf{x}(1) = (1-\alpha)\mathbf{x}(0) + \alpha \cdot \mathbf{W} \cdot \mathbf{x}(0)$. With $0 < \alpha < 1$, the new feature vector $\mathbf{x}(1)$ is a blend of our actual observation of the attacker's activity (i.e., $\mathbf{x}(0)$) together with our anticipation of its future activity (i.e., $\mathbf{W} \cdot \mathbf{x}(0)$).

The anticipation of future attacks is determined by the feature vector. Once we update the feature vector (from $\mathbf{x}(0)$ to $\mathbf{x}(1)$), the anticipation also changes. This, in turn, leads to an even newer feature vector. Let $\mathbf{x}(n)$ be the feature vector at step n in this process, we have the following recursive relationship: $\mathbf{x}(n+1) = (1-\alpha)\mathbf{x}(0) + \alpha \cdot \mathbf{W} \cdot \mathbf{x}(n)$. With proper normalization, the sequence converges to a stable solution. The feature vector $\mathbf{x}(\infty)$ can then be used to produce the ranking scores for the source.

Intuitively, this can be viewed as a probability propagation process. A source's attack probability with respect to contributor i propagates, following the outgoing edges of i in the correlation graph, to the neighbors of i. Each neighbor receives a share of this probability proportional to its strength of correlation with i. The probability received by the neighbors is then further distributed, in the similar fashion, to their neighbors. This process is essentially similar to the random walk used in Google's PageRank link analysis.

3 Usage Scenario and Expected Benefits

The envisioned benefit of the HPB strategy is to introduce a more predictive methodology for blacklist construction that can lead to the adoption of more effective dynamic firewall filters. In practice, we observe that the filter sets of perimeter defense devices may range from a few dozen to several hundred entries. Thus, when deciding which entries to incorporate into a given perimeter defense mechanism, the *opportunity cost* associated with incorporating poor performing filters provides a key motivation for our blacklist generation research. Of particular interest for our present experimental service is to provide blacklists that

- Exhibit a higher probability of being exercised once they are deployed. That is, a blacklist filter is exercised when an IP address that it incorporates attempts to send traffic to the target network and is subsequently blocked. We refer to the rate at which blacklist filters are exercised as the **hit rate** metric.
- Provide a **timely appearance** of entries in the firewall filter set. That is, we believe it is preferable for

[2] Innocuous address inclusion is not a unique HPB issue. All blacklisting schemes must include methods to reduce their occurrence.

a blacklist to not require that its included addresses saturate the network before they are incorporated. Better yet, a blacklist entry that has not yet been encountered by the network, but is subsequently exercised, provides a degree of proactive protection.

With respect to the issue of timely appearance, worst offender lists generally suffer from the fact that a source does not achieve candidacy until it has produced a sufficient mass of communications. GWOLs have the tendency to incorporate only sources that have already achieved mass proliferation. LWOLs are particularly limited by poor timing, as they are entirely reactive to attackers that are actively pounding the local network with unwanted traffic. With respect to the issue of hit rate, LWOLs generally achieve a much higher hit rate than GWOLs. Our examination of LWOL hit rates, based on DShield contributor logs, suggests that an attacker who is sending unwanted communications to the target will continue sending this bad traffic over time. GWOLs, unfortunately, do not exhibit high hit rates for most users, which provides the underlying motivation for extending the DShield blacklist services with HPBs. (In Figure 1 we will illustrate the disparity between GWOL and LWOL hit rates, relative to their performance against HPBs. While HPBs perform better than both strategies, they do less well with respect to LWOL, as LWOL hit rates are nearly uniformly higher than GWOL hit rates.)

In [9] we provide a detailed analysis of our performance assessment of HPB blacklist construction relative to the GWOL and LWOL strategies. Due to the length limitations of this paper we summarize our findings here, and refer the reader to the more in-depth discussion of HPB performance characteristics in [9]. The analysis used data from a 20-day window. We produced HPBs from the first 5 days, and evaluated their hit rates for all 1088 contributors over the subsequent 5-day prediction window. We then repeated this procedure over each 5-day interval for the 20-day window. The analysis processed more than 600 million log entry contributions. Attack sources were masked to their /24 addresses (DShield's GWOL also uses /24 masking). There are approximately 1.5 million such /24s in our analysis. Table 2 summarizes the HPB hit-rate improvements relative to equal-sized GWOL and LWOLS constructed from the same time interval. Hit-rate improvement is ploted as the hit count increase of the HPB blacklist over the other two blacklists during the 5-day period that followed the construction of the blacklists. We partition contributors into three classes corresponding to each table row: (rows 1-2) contributors that achieved improved hit counts using HPBs over GWOL and LWOL, (rowss 3-4) contributors with a zero hit count improvement over GWOL and LWOL, and (rows 5-6) contributors with hit counts that faired worse using HPBs than the GWOL and LWOL. For each group we indicate the average, meadian, and standard deviation hit count diffrence from using the HPB versus the corresponding GWOL and LWOL. Our experiments indicate that HPBs provide at least some improvement for the vast majority of contributors who would otherwise use DShield's traditional GWOL or their own LWOL. 3% of the DShield contributors experienced no hit rate improvement using either the GWOL or HPB, whereas 6% of users received no improvement from either the LWOL or HPB. 5% of the DShield contributors faired worse using their HPB than using the GWOL, and 11% faired worse using HPBs rather than LWOLs. We discuss possible reasons for good and bad performance later in this section.

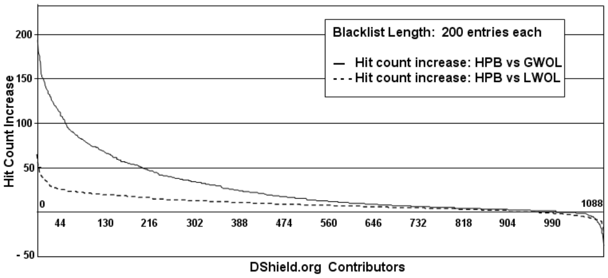

Figure 1. HPB Hit Rate Performance vs. GWOL and LWOL

Figure 1 provides a visual assessment of the hit rate performance of HPBs over all 1088 DShield contributors relative to comparable LWOLs and GWOLs of the same length. All blacklists are based on a 5-day construction window, with the hit rate comparison performed over the subsequent 5-day period. The Y-axis represents the delta-hit count change from using the HPB versus using the GWOL or LWOL. The X-axis represents the contributors, and the plotted lines are independently sorted from the most improved hit count increases to the least improved (including negative improvement). The lower line represents the HPB improvement over LWOL, which as we indicated earlier have hit rates that tend to exceed the GWOL rates. The improvement provided by HPB over LWOL include addresses that had not been encountered previously by the local network or had not crossed a threshold that would allow inclusion into the LWOL. In either case, the HPBs proved to provide a degree of timeliness in filter generation over LWOL. In general, we observe that the three blacklisting strategies do favor different kinds of source address behavior patterns. Through an analysis of the IP address count, unique destination port count, and contributor count we classify source addresses by their aggressiveness toward IP address scanning, the breadth or selectiveness of their port sweeping, the nature of their destination ports (malware, service, or application ports), or combinations of these elements. For example, the datasets contain source addresses that are observed by

	Contributor Percentage	Average Improvement	Median Improvement	StdDev Improvement
Improved vs. GWOL	91%	29	15	34
Improved vs. LWOL	83%	10	9	8
Neutral vs. GWOL	3%	0	0	0
Neutral vs. LWOL	6%	0	0	0
Poor vs. GWOL	5%	-7	-4	8.1
Poor vs. LWOL	11%	-4	-3	3.6

Table 2. Hit Count Performance: HPB vs (GWOL and LWOL) Using a 200 Entry Blacklist

hundreds of DShield contributors as they sweep thousands of high-number ports across large IP ranges (breadth and depth scanning). We also observe large-scale IP address sweeping on highly selective ports with strong association to malware, as well as other targeted port sweeps to standard network services. While these forms of behavior (large-scale IP scanning) make up the entirety of GWOL data, they are also well represented within the HPBs of many contributors. However, HPBs dominate the hit rate over GWOLs through their ability to also incorporate small-scale targeted scans that are observed by a few contributors.

In many cases, sources that are seen by fewer contributors but are caught by HPBs are often from more concentrated address spans within a /16 address range. In other cases, HPBs discover the common sources among certain contributors that display heavy malware-port scans. For some contributors, HPBs align very well with port sweeping activity across hundreds of high-order ports, but only targeting a few dozen or more contributors. For contributors that experience such breadth of port scanning patterns, both LWOLs and HPBs (due to this strong overlap pattern among a small set of contributors) begin to naturally align. In this sense, HPBs exhibit an ability to incorporate sources represented from both the GWOL and LWOL depending on the current experiences of the HPB's target contributor.

4 An Experimental DShield HPB Service Release

We now discuss the release of an experimental HPB service that is made available for free use to all DShield data contributors. DShield began providing a global worst offender blacklist soon after its inception, in the form of a text file. The text file was designed to be human readable as well as easy to parse by simple scripts. A number of open source as well as commercial firewalls use this list. HPBs are offered in a very similar form. To provide these customized blacklists, a credential exchange occurs (discussed below), followed by a release of a URL containing the contributor's unique HPB. Our default HPB length is 200 entries.

HPBs are calculated daily for the entire contributor base, not on demand (avoiding potential DoS conditions). At each daily computation interval, the algorithm computes the full set of HPBs for all DShield contributors from the activity records of the previous 5-day window (approximately 150 million record entries) within approximately 40 minutes. This is done with the anticipation that interested contributors may pull their updated HPB periodically, anywhere from every 1 to 5 days. For those contributors who fall into the *Improved* group of Table 2, the hit rate improvement of HPBs lasts through at least the 5-day prediction window. (See [9] for a detailed experimental demonstration.)

A DShield user is provided with an account to the DShield website in order to review reports. To ease the automatic retrieval of a user's HPB, we do not require the user to log in via the standard web-based procedure. Instead, the user can generate a unique token. This random hexadecimal sequence will be appended to the URL and identify the user. This token has a number of advantages over using the user's username and password. For example, the user may still change the password without having to change the information used by automated scripts retrieving the blacklist.

To provide further protection of the integrity and confidentiality of the HPB, it is offered to users via https. A detached PGP signature can be retrieved in case https is not available or not considered sufficient to prove the authenticity of the list.

```
# DShield Customized HPB Blacklist
# created 2007-01-19 12:13:14 UTC
# For userid 11111
# Some rights reserved, DShield Inc.,
#
# Creative Commons Share Alike License
# License and Usage Info:
# http://www.dshield.org/blocklist.html
1.1.1.1  255.255.255.0  testnet 1
2.2.2.2  255.255.255.0  testnet 2
# End of list
```

Figure 2. Sample HPB File

The HPB itself uses a simple tab delimited format. The first column identifies the network address, and the second column provides the netmask. Additional columns can be used to provide more information about the respective offender, like type of attacks seen, name of the network, and country of origin. Initially, such additional columns are intended for human review of the blacklist. All comments start with a # mark. Figure 2 shows a sample HPB.

5 Future Work and Conclusion

We view the HPB service as a first experimental step toward applying one branch of predictive data analysis, a link analysis method, to the important problem of high-quality blacklist generation. Most contemporary forms of blacklist selection have been based on the concept of ostricizing the worst offenders, which remains a well-intentioned strategy with clear merit. Unfortunately, such methods are often prone to deliver the blacklist subscriber with filters that will seldom be exercised or that report addresses well after they have saturated one's network. In this paper, we present the highly predictive blacklisting as a new alternative strategy, and discuss some preliminary testing results as well as an experimental service on DShield to provide HPBs.

As our experimental service progresses we anticipate the need for significant refinement and envision potential future extensions toward continuing to increase the quality of blacklist entries that appear within the DShield website. One future extension is to incorporate into blacklist formulation attacker severity metrics that reflect the benignness or aggressiveness of the attackers historical log patterns. In practice, DShield's current blacklist generator requires care to avoid including benign sources that are accidentally logged by contributors, such as Internet measurement services, web crawlers, or software update services. These sources are also prefiltered from consideration by the HPB construction algorithm. Also, a common false positive arises when servers respond slowly to client requests, resulting in replies being blocked if the firewall times out. In order to eliminate these logs, care is taken to filter logs that arise from commonly timed out services, such as TCP ports 53, 25, 80, and 443. Alternatively, some addresses may be prioritized based on a combination of attributes, including their propensity toward address sweeping and malicious port scanning. Such filters and priority attributes could be incorporated into prefiltering mechanisms or not applied until the final blacklist entry inclusion decisions.

There is another issue we plan to address in our future work. Attackers who are aware of the HPB formulation may take measures to avoid being listed. There may be two types of such measures. An attacker can actively generate specially crafted attacks such that these attacks lead to a different correlation patterns among the contributors. An attacker may also take a passive measure to elude the HPB: it computes the correlation patterns the same way as HPB and then select targets that would not give it high ranking score if detected. However, we believe that the two types of measures are quite expensive for the attackers, assuming that there are many attackers and they are not colluding. In the active scenario, since the correlation patterns are extracted from the statistics of the activities of the whole attacker population, to affect the patterns, an attacker (or a few attackers) has to trigger reports that are significant in quantity comparing to that produced by the whole population. In the passive scenario, to avoid being listed, the attacker is forced to select targets that may not be as valuable as without HPB. In particular, the attacker may have to target networks that are not interesting to the majority of the attacker pool. Hence to avoid HPB, it will require much more work from the attacker and it may put the attacker in a disadvantaged position among the attacker population. Despite this, we do think that these attacker measures pose an issue and we would like to investigate in the future work how to make HPB more resilient to such measures.

References

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[2] Zesheng Chen and Chuanyi Ji. Optimal worm-scanning method using vulnerable-host distributions . In *International Journal of Security and Networks (IJSN) Special Issue on "Computer & Network Security"*, 2003.

[3] Symantec Corporation. Deepsight threat management system home page. http://tms.symantec.com, 2007.

[4] S. Katti, B. Krishnamurthy, and D. Katabi. Collaborating against common enemies. In *Proceedings of the ACM SIGCOMM/USENIX Internet Measurement Conference*, October 2005.

[5] Steven M. Kay. *Fundamentals of statistical signal processing: estimation theory.* Prentice Hall, 1993.

[6] Johannes Ullrich. DShield global worst offender list. https://feeds.dshield.org/block.txt.

[7] Johannes Ullrich. DShield home page. http://www.dshield.org, 2007.

[8] V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the domino overlay system. In *Proceedings of Network and Distributed Security Symposium*, June 2004.

[9] J. Zhang, P. Porras, and J. Ullrich. Highly predictive blacklisting. Technical report, SRI International, 2007. available at http://www.cyber-ta.org/releases/HPB/HPB-SRI-TR-APril2007.pdf.

[10] Jian Zhang, Phillip Porras, and Johannes Ullrich. Dshield highly predictive blacklist service. http://www.dshield.org/hpbinfo.html.