ERIC LANGHEINRICH

http:BL: taking DNSBL beyond SMTP

Eric Langheinrich is CTO and Co-founder of Unspam Technologies, Inc., and an expert in the field of detection and identification of malicious network activity. Eric and his team at Unspam pioneered the Project Honey Pot.

eric@eric.unspam.com

FOR MANY YEARS, EMAIL RECIPIENTS have benefited from the use of various Domain Name Blacklists (DNSBLs) in the fight against spam. Through efficient DNS lookups, mail servers can check individual connecting clients against various blacklists. Major DNSBLs include SpamHaus, SORBS, SURBL, and MAPS. These DNSBLs provide mail servers with the ability to decide how client requests are handled from hosts based on individual blacklist criteria. Hosts are able to decide to block requests, allow requests, or perform extra spam filtering scrutiny on messages from hosts based on results from blacklist lookups.

Mail servers, however, are not the only network resource that needs to be protected from malicious machines. Web servers face a constant assault from malicious Web robots that are harvesting email addresses, looking for exploits, and posting comment spam. At the most basic level, these malicious robots rob a site of significant bandwidth. More importantly, by allowing these robots to troll your site you open yourself to the possibility of future attacks via spam or Web-based vulnerabilities.

Project Honey Pot's (www.projecthoneypot.org) new http:BL service is similar to a traditional mail server DNSBL, but it is designed for Web traffic rather than mail traffic. The data provided through the service empowers Web site administrators, for the first time, to choose what traffic is allowed onto their sites. By stopping malicious robots before they can access a Web site, the http:BL service is designed to save bandwidth, reduce online threats, and decrease the volume of spam sent to the gateway by preventing spammers from getting email addresses in the first place.

Each day, thousands of robots, crawlers, and spiders troll the Web. Web site administrators have few resources available to tell whether a visitor to a site is good or malicious. Project Honey Pot was created as an open community to provide this information to Web site administrators, enabling them to make informed decisions on whom to allow onto their sites.

Project Honey Pot is a distributed network of decoy Web pages that Web site administrators can include on their sites to gather information about robots, crawlers, and spiders. The project collates

data on harvesters, spammers, dictionary attackers, and comment spammers and makes this data available to its members to help them protect their Web sites and inboxes.

Web site administrators who want to participate in providing data to Project Honey Pot do so by installing a script on their site. Web site administrators include hidden links on their existing pages to the honeypot script. The links are designed to be hidden from human visitors but followed by robots. The honeypot script, when accessed, produces a Web page. Hidden on the page are trap elements, including unique email addresses and Web forms. If information is sent to these trap elements, then it is recorded by Project Honey Pot and included in the http:BL. Scripts are published open source and are currently available for PHP, Perl, ASP, Ruby, ColdFusion, and SAP NetWeaver.

Currently, Project Honey Pot has tens of thousands of installed honeypots and members in over 114 countries spanning every continent but Antarctica. Members can also participate in the project by "donating" MX records from their domains to the project. Donated MXs extend the network, allowing Project Honey Pot to track spam servers and dictionary attackers. Donated domains allow Project Honey Pot to generate a virtually unlimited number of spam trap email addresses that are difficult to detect. Together, these resources help gather information on malicious Web robots.

The http:BL service makes this data available to any member of Project Honey Pot in an easy and efficient way. To use http:BL, a host need simply perform a DNS lookup of a Web visitor's reverse IP address against one of the http:BL DNS zones. Then http:BL's DNS system will return a value that indicates the status of the visitor. Visitors may be identified as search engines, suspicious, harvesters, comment spammers, or a combination thereof. The response to the DNS query indicates what type of visitor is accessing the Web site, the threat level of the visitor, and how long it has been since the visiting IP was last seen on the Project Honey Pot network.

Each user of http:BL is required to register with Project Honey Pot. Each user of http:BL must also request an Access Key to make use of the service. All Access Keys are 12 characters in length, are lowercase, and contain only alpha characters (no numbers).

All queries must include your Access Key followed by the IP address you are seeking information about (in reverse-octet format) followed by the List-Specific Domain you are querying. Imagine, for example, you are querying for information about the IP address 127.9.1.2 and your Access Key is abcdefghijkl, then the format of your query should be constructed as follows:

```
abcdefghijkl.2.1.9.127.dnsbl.httpbl.org
[Access Key]  [Octet-Reversed IP]  [List-Specific Domain]
```

Two important things to note about the IP address in the query: First, the IP address is of the visitor to your Web site about which you are seeking information; second, the IP address must be in reverse-octet format. This means that if the IP address 127.9.1.2 visits your Web site and you want to ask http:BL for information about it, you must first reverse the IP address to be formatted as 2.1.9.127.

Note that if you reverse the order of the octets (the numbers separated by the periods) you do not reverse the IP address entirely. For example, if you were querying the IP address 10.98.76.54, the following are examples of correct and incorrect examples of reverse-octet format:

Query: 10.98.76.54
Right: 54.76.98.10
Wrong: 45.67.89.01

Three scenarios exist for responses from the http:BL service. The cases include (1) not listed, (2) listed, and (3) known search engine. A majority of IP addresses do not appear in http:BL's records. If the IP queried does not appear, http:BL will return a nonresult {NXDOMAIN}. A query for a listed entry or search engine will receive a reply from the DNS server in IPv4 format with three of the four octets containing data to provide information about the visitor. The intention is for this to allow flexibility in how the Web site administrator treats the visitor rather than a simple black-and-white response (e.g., the administrator may want to treat known harvesters differently from known comment spammers, by blocking the former from seeing email addresses while blocking the later from POSTing to forms).

Responses for listed entries will have one of two predefined formats depending on whether the entry is for a known search engine or for a malicious bot. The fourth octet represents the type of visitor. Defined types include "search engine," "suspicious," "harvester," and "comment spammer." Because a visitor may belong to multiple types (e.g., a harvester who is also a comment spammer) this octet is represented as a bitset with an aggregate value from 0 to 255. A chart outlining the different types is shown in Table 1. This value is useful because it allows you to treat different types of robots in different ways.

Value	Meaning
0	Search Engine
1	Suspicious
2	Harvester
4	Comment Spammer
8	[Reserved for Future Use]
16	[Reserved for Future Use]
32	[Reserved for Future Use]
64	[Reserved for Future Use]
128	[Reserved for Future Use]

TABLE 1

Because the fourth octet is a bitset, visitors who have been identified as falling into multiple categories may be represented. See Table 2 for an explanation of the current possible values.

IPs are labeled as "suspicious" if they engage in behavior that is consistent with a malicious robot but malicious behavior has not yet been observed. For example, on average it takes a harvester nearly a week from when it finds an email address to when it sends the first spam message to that address. In the meantime, the as-of-yet-unidentified harvester's IP address is seen hitting a number of honeypots, not obeying rules such as those set forth by robots.txt, and otherwise behaving suspiciously. In this case, the IP may be listed as suspicious.

The third octet represents a threat score for the queried IP. This score is assigned internally by Project Honey Pot based on a number of factors, such as the number of honeypots the IP has been seen visiting and the damage done

Value	Meaning
0	Search Engine (0)
1	Suspicious (1)
2	Harvester (2)
3	Suspicious & Harvester (1+2)
4	Comment Spammer (4)
5	Suspicious & Comment Spammer (1+4)
6	Harvester & Comment Spammer (2+4)
7	Suspicious & Harvester & Comment Spammer (1+2+4)
>7	[Reserved for Future Use]

TABLE 2

during those visits (email addresses harvested or forms posted to). The score ranges from 0 to 255, where 255 is extremely threatening and 0 indicates that no threat score has been assigned.

Project Honey Pot assigns threat scores to IP addresses observed on the Project Honey Pot network as part of the http:BL service. Threat scores are a rough guide to determine the threat that a particular IP address may pose and therefore should be treated as a rough measure. Although threat scores range from 0 to 255, they follow a logarithmic scale, which makes it extremely unlikely that a threat score over 200 will ever be returned.

Different threats calculate threat scores slightly differently. For example, a threat score of 25 for a harvester is not necessarily as threatening as a threat score of 25 for a comment spammer. A harvester's threat score is determined based on its reach (the number of honeypots it has hit), its damage (the number of email messages that have resulted from its harvests), its activity (the frequency of visits over a period of time), and other factors.

The second octet represents the number of days since the last activity was observed by the IP on the Project Honey Pot network. This value ranges from 0 to 255 days. This octet is useful in helping you assess how stale the information provided by http:BL is and, therefore, the extent to which you should rely on it.

The first octet is always 127 and is predefined to not have a specified meaning related to the particular visitor.

The following is an example of a hypothetical query and hypothetical response, which will be referenced throughout the rest of this section:

Query: abcdefghijkl.2.1.9.127.dnsbl.httpbl.org
Response: 127.3.5.1

The response means the visitor has exhibited suspicious behavior on the Project Honey Pot network, has a threat score of 5, and was last seen by the project's network 3 days ago.

Search engines represent a special case. Known search engines will always return a value of zero as the last octet. It is not possible for a search engine to be both a search engine and some kind of malicious bot. Search engines found to be harvesting or comment spamming will cease to be listed as search engines.

In the case of a known search engine indicated by the fourth octet being 0, the third octet becomes a serial number identifier for the specific search en-

gine. The second octet is reserved for future use.

With the launch of the http:BL service, Project Honey Pot released the module mod_httpbl Apache. The mod_httpbl module provides an efficient mechanism for Web site administrators to take advantage of Project Honey Pot's http:BL service. The mod_httpbl module provides for server-level decision making based on http:BL data.

Rules are defined within the httpd.conf (Apache configuration) file and are indicated by the HTTPBLRBLReqHandler directive. Rules are structured in the form of [A] : [B] – [C] : [D] – [E] : [F] Action String where:

[A]—A bitmask [0–255] of the HTTP methods (in decimal representation). For example:
1—GET
2—POST
4—HEAD
8—PUT
[B]—The lower bound for DNSBL value octet 2
[C]—The upper bound for DNSBL value octet 2
[D]—The lower bound for DNSBL value octet 3
[E]—The upper bound for DNSBL value octet 3
[F]—A bitmask [0–255] of the offending type (in decimal representation) that should match
ACTION_STRING—Action to take when rule is matched. Currently supported options include "allow," "deny," and "allow-xlate-emails." For example, consider the following Action String:

```
# Serve all search engines, but replace email address text and links in HTML content.
HTTPBLRBLReqHandler 255:0-255:0-255:0 allow-xlate-emails
# Deny known comment spammers the ability to POST.
HTTPBLRBLReqHandler 2:0-255:0-255:4 deny
# Serve all harvesters, but replace email address text and links in HTML content.
HTTPBLRBLReqHandler 255:0-255:0-255:2 allow-xlate-emails
# Deny known exploiters the ability to request via HTTP method HEAD
HTTPBLRBLReqHandler 4:0-255:0-255:8 deny
# Deny any requests originating from IPs known to Project Honey Pot to be suspicious or offensive.
HTTPBLRBLReqHandler 255:0-255:0-255:255 deny
```

The Apache module allows for granular rule sets to be defined based on an HTTP method bitset at the directory, virtual host, and server levels. The mod_httpbl rules extend basic "allow or deny" functionality to include redirects to virtual Honey Pot pages and the rewriting of email links and email address text to customized values automatically based on the http:BL query result values. For example, consider this rule set:

```
<VirtualHost>
HTTPRBLReqHandler (criteria_aaa)
<Directory ~ ^/dir1/>
HTTPRBLReqHandler (criteria_bbb)
HTTPRBLReqHandler (criteria_ccc)
</Directory>
HTTPRBLReqHandler (criteria_ddd)
<Directory ~ ^/dir1/images/>
HTTPRBLReqHandler (criteria_eee)
HTTPRBLReqHandler (criteria_fff)
</Directory>
HTTPRBLReqHandler (criteria_ggg)
</VirtualHost>
```

Members of the Project Honey Pot community have also provided http:BL implementations back to the community, including implementations for Drupal, phpBB, WordPress, and OddMuse. Additionally, members have posted sample code for several scripting languages in the http:BL development bulletin boards.

In a continuing effort to reduce threatening traffic, Project Honey Pot, in conjunction with its members, will continue to identify sources of malicious traffic and provide open platform tools to Web site administrators to help safeguard systems, save bandwidth, reduce online threats, and decrease the volume of spam sent to the gateway by preventing spammers from getting email addresses in the first place.