# Setting HoneyTraps with ModSecurity: Project Honeypot Integration

www.trustwave.com/Resources/SpiderLabs-Blog/Setting-HoneyTraps-with-ModSecurity--Project-Honeypot-Integration/

Posted By Ryan Barnett

Following up my previous blog post which outlined how to activate additional HTTP ports to catch automated scanners, this blog post will discuss more HoneyTrap techniques for catching malicious users who visist your real site. I will show how ModSecurity can be used to share data with Project Honeypot.


## Project Honeypot Background

Project Honeypot gives this brief overview:

> *Project Honey Pot is the first and only distributed system for identifying spammers and the spambots they use to scrape addresses from your website. Using the Project Honey Pot system you can install addresses that are custom-tagged to the time and IP address of a visitor to your site. If one of these addresses begins receiving email we not only can tell that the messages are spam, but also the exact moment when the address was harvested and the IP address that gathered it.*
>
> *To participate in Project Honey Pot, webmasters need only install the Project Honey Pot software somewhere on their website. We handle the rest — automatically distributing addresses and receiving the mail they generate. As a result, we anticipate installing Project Honey Pot should not increase the traffic or load to your website.*
>
> *We collate, process, and share the data generated by your site with you. We also work with law enforcement authorities to track down and prosecute spammers. Harvesting email addresses from websites is illegal under several anti-spam laws, and the data resulting from Project Honey Pot is critical for finding those breaking the law.*
>
> *Additionally, we will periodically collate the email messages we receive and share the resulting corpus with anti-spam developers and researchers. The data participants in Project Honey Pot will help to build the next generation of anti-spam software.*

## Registering Your Honeypot

After creating your ProjectHoneypot account and logging in, you can then register your website domain and obtain the main honeypot script to host on your website. When malicious clients access this script, it will dynically communicate with Project Honeypot to obtain tagged email addresses and Form data to display to the client. This is the data that is tracked by Project Honeypot to see if the email address receives any email. If it does, then the web client that accesed the script is tagged within Project Honeypot as a SPAMMER. This concept is a bit similar to real-world examples of marking paper currency for tracking purposes after a bank robbery.

## Adding Honeypot Links with ModSecurity

Now that you have the main honeypot script registered and placed on your website, the next step is to seed your web application with links pointing to the honeypot script. Keep in mind that you do not want legitimate clients to

stumble upon these links, therefore, the links use various HTML tricks make them non-visible to humans using web browsers. Malicious web clients, however, will easily scrape the URLs in the links and follow the links to our honeypot script.

Rather than manually editing raw HTML files on your website to add in these links, you can use ModSecurity's new data substitution operator (@rsub) to add in the links to the outbound response data. The following rules will dynamically add in an href link pointing to the honeypot script to the bottom of all HTML pages to your site:

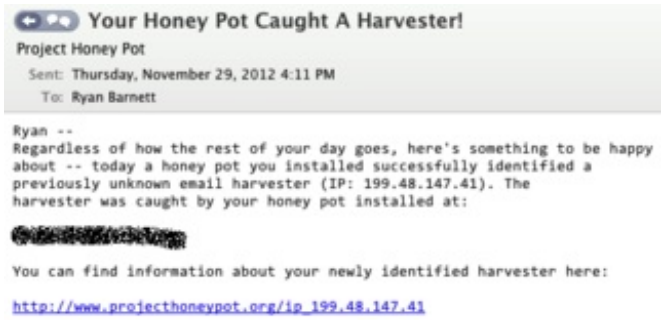
```
SecContentInjection On
SecStreamOutBodyInspection On

SecRule RESPONSE_CONTENT_TYPE "@contains text/html"
"chain,id:'999001',phase:4,t:none,nolog,pass"
  SecRule STREAM_OUTPUT_BODY "@rsub s/<\/html>/<a
href=\"http:\/\/www.yoursite.org\/cgi-bin\/somefile.cgi\"><\/a><\/html>/"
```

Once added to your site, this how the bottom of the raw HTML page would look with the link at the bottom:

```
<script type="text/javascript">

  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-34751004-1']);
  _gaq.push(['_trackPageview']);

  (function() {
    var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();

</script>
</body>
<a href="http://[redacted]cgi"></a></html>
```

### Flagged Client Notifications

If any SPAMMERS send emails to the addresses returned in the html of the honeypot script, then the client will be marked as malicious and I will be notified via email:


Your Honey Pot Caught A Harvester!

Project Honey Pot

Sent: Thursday, November 29, 2012 4:11 PM

To: Ryan Barnett

Ryan --
Regardless of how the rest of your day goes, here's something to be happy about -- today a honey pot you installed successfully identified a previously unknown email harvester (IP: 199.48.147.41). The harvester was caught by your honey pot installed at:

[redacted]

You can find information about your newly identified harvester here:

http://www.projecthoneypot.org/ip_199.48.147.41

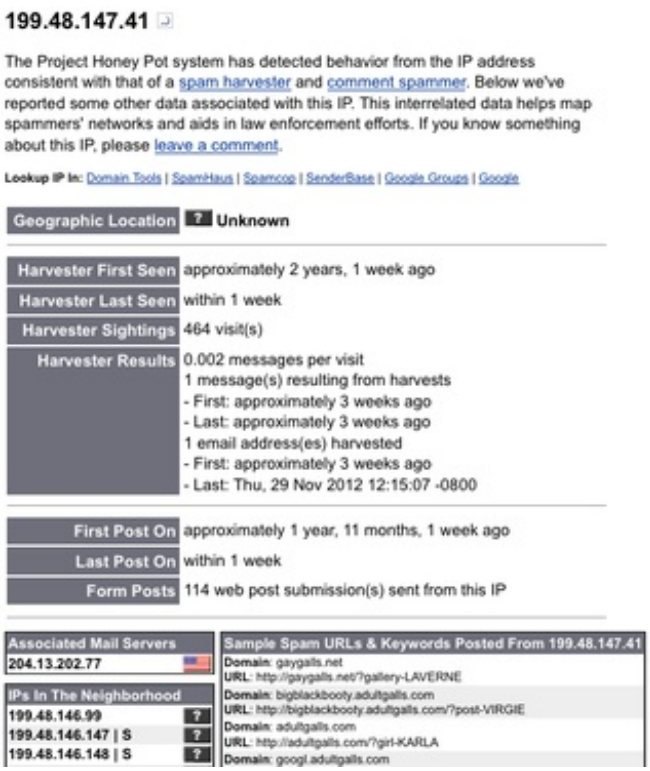
The IP app on the project honeypot page provides an overview of all intelligence about this malicious IP address:

### Utilizing the HTTP BlackList (HTTPBL) API

Another very useful tool provided by Project Honeypot is the HTTP Blacklist (HTTPBL). They describe the HTTP BL as follows:

> *The HTTP Blacklist, or "http:BL", is a system that allows website administrators to take advantage of the data generated by Project Honey Pot in order to keep suspicious and malicious web robots off their sites. Project Honey Pot tracks harvesters, comment spammers, and other suspicious visitors to websites. Http:BL makes this data available to any member of Project Honey Pot in an easy and efficient way.*
>
> *Http:BL provides data back about the IP addresses of visitors to your website. Data is exchanged over the DNS system. You may query your local DNS server and receive a response back that indicates the type of visitor to your site, how threatening that visitor is, and how long it has been*

> *since the visitor has last been seen within the Project Honey Pot trap network.*




This is useful data as it tracks IP address of clients who have been flagged as malicious by the Project Honeypot's trap network which means that there is a very low chance of false positives. In the latest ModSecurity version (2.7), we added the capability to use the Http:BL API by allowing the ModSecurity user to specify their registered API key with the new SecHttpBlKey directive.

> ***SecHttpBlKey***
>
> ***Description:*** *Configures the user's registered Honeypot Project HTTP BL API Key to use with @rbl.*
>
> ***Syntax:***
> `SecHttpBlKey [12 char access key]`
>
> ***Example Usage:*** `SecHttpBlKey whdkfieyhtnf`
>
> ***Scope:*** *Main*
>
> ***Version:*** *2.7.0*
>
> *If the @rbl operator uses the dnsbl.httpbl.org RBL (http://www.projecthoneypot.org/httpbl_api.php) you must provide an API key. This key is registered to individual users and is included within the RBL DNS requests.*

You can then use rules similar to the following to check the client IP address against the HTTP BL:

```
SecHttpBlKey whdkfieyhtnf
SecRule TX:REAL_IP|REMOTE_ADDR "@rbl dnsbl.httpbl.org"
"id:'99010',chain,phase:1,t:none,capture,block,msg:'HTTPBL Match of Client
IP.',logdata:'%{tx.httpbl_msg}',setvar:tx.httpbl_msg=%{tx.0}"
        SecRule TX:0 "threat score (\d+)" "chain,capture"
                SecRule TX:1 "@gt 20"
```

If a malicious client (such as 188.92.75.41) connects to your web server, this rule will inspect the "threat score" data returned by the HTTP BL and then it will trigger an alert if it is above the defined threshold limit (20 here). An example alert would be generated and the client would be blocked (depending on your configuration).

```
[Tue Dec 18 16:22:44 2012] [error] [client 173.44.37.234] ModSecurity: Warning.
Operator GT matched 20 at TX:1. [file
"/usr/local/apache/conf/crs/base_rules/modsecurity_crs_15_custom.conf"] [line "2"]
[id "999010"] [msg "HTTPBL Match of Client IP."]
[data "RBL lookup of whdkfieyhtnf.234.37.44.173.dnsbl.httpbl.org succeeded at
REMOTE_ADDR.
Suspicious comment spammer IP: 1 days since last activity, threat score 80"]
[hostname "MacBook-Pro-2.local"] [uri "/cgi-bin/printenv"] [unique_id
"UNDeo8CoAWoAACDARMkAAAAC"]
```

## Conclusion

I hope that the data presented in this blog post will help to show how to contribute to Project Honeypot and to leverage their outstanding data.