



A Student IT Security Awareness Initiative by

The Philippine Honeynet Project

Honey pots 101: A Honey pot By Any Other Name

By Ryan Talabis

The concept of the Honey pot is not new; in fact, it's been around for a while. It just seems so because until recently there has been no clear and widely accepted definition. In fact, in my dealings with IT security professionals here in the Philippines, I still get as many different definitions with as many different people when talking about Honey pots.

For example, some of the people¹ I have talked to defined a Honey pot as tool for law enforcement, some think of it as an NIDS while others think of it as a deception device. They are all correct because Honey pots are indeed all that but as you can see, it is here where the problem starts.

I believe that the difference in Honey pot definitions among security professionals, at least here in the Philippines, seems to stem from the misconception that in order to define a Honey pot, one must define how it is used rather than defining it through its nature and value.

One of the best definitions on what a Honey pot is comes from the Honey pot mailing list, a list consisting of about 5000 different security professionals (including the author) working with Honey pot technology. The definition is:

*A Honey pot is a security resource whose value is in being probed, attacked or compromised.*²

Let's break this definition down into parts so we can better understand it. Let's begin with the first part:

A Honey pot is a security resource...

As stated in the definition above, a Honey pot is a security resource. This security resource may come in different shapes and sizes. In fact, a Honey pot could just as simply be one of your old PC's, a script or even a digital entity³ like some made-up patient records.

For example, in one of my first Honey pot attempts, though at that time I had no idea that what I had was a Honey pot, I had this marvelous idea (or so I thought then) to study viruses and worms by actually getting them from the wild⁴ and then disassembling them.

To accomplish this, I took an old computer, a Pentium I machine with a Windows 98 default installation with an out of the box virus scanner installed. I then connected it to a dial-up

¹ Initial Honey net Project feasibility study

² Security focus Honey net mailing list (<http://www.securityfocus.com>)

³ A Honey token (<http://www.securityfocus.com/infocus/1713>)

⁴ The Internet

connection and waited. It probably took me a little less than 5 minutes to get my first worm. Suffice to say, I caught a lot of worms and viruses that day. Unfortunately, the only product of the experiment was an unusable computer that had to be reformatted.

But even though the experiment was not quite as successful as I had hoped, I was to my surprise, able to effectively “stage” a security breach in a “controlled” environment. It was a very interesting experiment and it led me to a better understanding, as well as respect for worms and viruses but more importantly, it was a sort of “initiation” to Honeyd for me though as I’ve said, I didn’t know that my experiment was a Honeyd at that time.

In the experiment I did above, I used an actual physical computer but as I’ve said, Honeyd could come in different shapes and sizes. The important thing is that it’s a security resource. For example, one interesting Honeyd implementation is Honeyd⁵. Honeyd is a script that can simulate multiple computers, multiple operating systems and even run multiple services so that hackers would think that they are actually “hacking” a real computer.

In my more recent experiments, I installed Honeyd in a machine at our local University network. From it, I created a dozen different “fake” computers with different operating systems running different services. Once created, it began to pick up a dozen different probes targeted towards the fake computers that I created. Since they were fake computers that just went online then why should anyone be trying to use them? This now bring us to the next part of our definition...

...whose value is in being probed, attacked or compromised.

Try thinking of it this way. If you put a big patch of wet cement in front of your door, you’ll know if someone entered your house through the door right? Simple enough.

Just like the wet cement, the concept of the Honeyd is quite simple too. Think of the Honeyd as our patch of wet cement and our door our network. If anyone “touches” our Honeyd, then we know that someone’s creeping around in our network. And since a Honeyd is not a production system, no person or resource should be communicating with it. Thus any incoming traffic or more dangerously, outgoing traffic would be considered unauthorized traffic.

In my example I used above where I used one of my old PCs as a Honeyd, my machine was not a production system thus any traffic that would be going through it could only be unauthorized traffic since no one except me had any access to that computer.

In another one of my early experiments, this time when I was starting with the Philippine Honeyd Project⁶, I installed the IDS Snort in a Win ME machine then used a dial-up connection to connect to the PC to the Internet. I then ran snort and just watched for about 30 minutes. Since I’m the only one using that computer, no one should be accessing it. Lo and behold, in just a few minutes, I was getting connections and probes from different IP’s already. I checked these IP’s out and found out that they we’re coming from China and surprisingly, a company in Alabang here in the Philippines.

Why would people in China or people from a company in Alabang be doing accessing my computer? My computer was not a server and besides, it was using a dynamic public IP! As you can see, since we’re sure that no one else should be accessing our Honeyd, we know that anyone accessing it would most likely be unauthorized “visitors”. This is what makes the Honeyd such a valuable security resource, its inherent ability to detect potentially malicious traffic because there should be no traffic going through it in the first place. From there, we can now do all sorts of things. Since all traffic coming to our Honeyds would most likely be malicious, we have now created for ourselves a resource where we can detect and

⁵ Honeyd by Niels Provos (<http://www.honeyd.org>)

⁶ Philippine Honeyd Project (<http://www.philippinehoneydproject.org>)

study “malicious” internet traffic easily and efficiently without going through tons of logs and analyzing which traffic are indeed malicious or not. As I’ve said in the earlier part of this paper, we can now explore the myriad of uses⁷ of Honeyd’s like law enforcement, intrusion detection, intrusion prevention, trends analysis, reverse engineering and exploratory research. The list is not limited to this since as I’ve stated again and again, the Honeyd is a resource and its up to the security professional how he or she is going to use that resource.

On that note, let us now briefly reiterate what a Honeyd is:

1. A Honeyd is a security resource whose value is in its being probed, attacked or compromised.
2. A Honeyd could come in different sizes. It can be one of your old PC’s, a script like Honeyd or even more complicated setups like the Honeydnet⁸.
3. A Honeyd looks and acts like a production system but in reality is not so. Since its’ not a production system, no ones supposed to use it thus should have no valid traffic. So if we detect traffic, most likely its potentially malicious traffic.

Simple enough right? But we’ve only scratched the surface about Honeyd’s. In our next paper, we shall be discussing the history of Honeyd’s. From there I shall introduce you to the pioneers and some of the pioneering examples of Honeyd’s like Back Officer Friendly, The Deception Toolkit and Honeyd to name a few.

About the author

Ryan Talabis is the founder of the Philippine Honeydnet Project. He is currently working as a web and database consultant for the Asian Development Bank. Formerly, he was the Director of Software Development and Technology for Slingshot Interactive, a start-up web development firm in Manila. He loves to work with honeyd’s, has a pretty good grasp in making pretty websites and loves to make charts for everything. In his free time, he is an aggressive inline skater and hopes to compete in the X-Games someday. You can contact him at talabis[at]gmail.com.

About the Philippine Honeydnet Project

The Philippine Honeydnet Project (<http://www.philippinehoneynet.org>) is a non-profit, all volunteer group dedicated to promoting honeydnet and IT security research in the Philippines. It is a member of Honeydnet Research Alliance (<http://www.honeydnet.org>), an organization of Honeydnet initiatives around the world.

⁷ More on this on an upcoming Honeydnet 101 series

⁸ The Honeydnet Project (<http://www.honeydnet.org>)