Pseudoservers - Honeypots without the Hassle – Part 1.
Paul M.Wright July 27th 2003

After taking part in Lance Spitzners excellent and informative Honeypot training session at SANSFire Washington this month my mind has been on the subject of Honeypots, Honeynets and Honeytokens as can be read at http://www.honeynet.org and http://www.tracking-hackers.com/ .

There are some key points surrounding this technology which are very important I believe to the development of IT security in general.

Firstly is that Honeynets are undoubtedly one of the most exciting and fast developing areas of IT Security and a very cleverly conceived way to "Know your enemy". Secondly, as such a new technology there may be some room for further development of concepts to overcome legal and practical objections that have arisen.

Honeypots, Honeynets and Honeytokens relate to a server, network or file/record on a server that attracts a hacker to it and then informs the admin that a hacker has interacted with it somehow and then allows the admin some choice about how to make further interaction with hacker. There are wider definitions but the idea of **attracting** a hacker is implicit in the use of the word **Honey** as to a Bee of course. There can be high interaction Honeypots or low interaction depending on how much the software systems and admin of the Honeypot interact with the hacker. The idea being that a research Honeypot could afford to interact highly and take on the risk of upsetting them as there is less to lose but a production Honeypot on commercial network would be low interaction to minimise this risk. This is the nub of the current issues with Honeypots. Before I address this risk I would like to outline the general advantages and disadvantages of Honeypots. (This list is not exhaustive).

Advantages of Honeypots, Honeynets and Honeytokens.

1. Gives information about hackers on your network.
2. Can give initiative to admin as the hacker thinks they are in control but in fact the admin is leading them on.
3. Can act as an alert to an insider threat.
4. Cheap.
5. Can take the hackers attention from the actual network like a decoy.
6. Easily detectable, recognisable. As Honeypots are not part of the legitimate network any traffic on them is by definition unauthorised and probably from a hacker either inside or outside. (Employee port scanning perhaps).

Disadvantages of Honeypots

1. Entrapment could be used as an argument to evade prosecution by the hacker if the case ever went to court
2. Privacy of the hacker is being invaded by snooping on their messaging and their recipients messaging.
3. A false weakness may attract attention that otherwise may not have arisen to your network.
4. An attack may be made from your network honeypot to another companies network which you may then be held liable for.

5. A public announcement of a hack maybe made even though it was just a honeypot which may destroy confidence and affect the companies brand value.

To my mind it would be useful to make a further classification here in order to tease away the disadvantages of Honeypots and leave the advantages.

Now the first Disadvantage listed, that of entrapment, only occurs as a way to escape prosecution if the law enforcement agencies have been involved but it is still a big issue both legally and morally. **Entrapment is implicit in the current terminology of Honeypots.** Honey attracts the Honey-Bee-hacker. The Honey being a juicy vulnerable unpatched server. Perhaps what is needed is a very low, interaction Honeypot without any honey in it at all. Think about it. A server with all the alerting, logging of a Honeypot with the same premise of having zero authorised access can still provide all of the advantages with none of the disadvantages mentioned above. Simply take away the unpatched known vulnerability and the enticement factor. If a hacker attempts to hack this new box they cannot claim that they were enticed. This hacker is deliberately invading a computer for which they have not been given any hint of permission. Who wants to attract hackers to their network anyway?! Almost certainly not on a production network. Also the value of allowing a hacker to run an exploit I know about already does seem to be quite small compared to the risks that it involves.

So the idea I am proposing is to have a patched server that is not part of the interrelated workings of the production system. We have a honeypot without any honey in it. We could call this a Pot but perhaps there may be some confusion with this terminology. For ease of use we will call this a “Pseudoserver”. The Pseudo dictionary meaning is “false”, “deceptive”, “apparently similar to…” (a normal production server).

A Pseudoserver is not addressed by any email, DNS, web-serving etc. It simply sits on the network and waits to be scanned and hacked. When it is scanned then an alert is sent. The machine is patched as normal so if the attacker does successfully run an exploit then this has implications for the rest of the actual production machines. This is an exploit we really want to know about –the Zer0 day. A Honeypot does not report a Zero day as it has an old vulnerability deliberately placed on it to attract the hacker. Here comes another bonus, a Pseudoserver is a lot easier to test for exploits that have been ran against it, it can be audited easily as it is not actually being used. MD5 checksums of the operating system files can be made that are not going to change so they can be checked easily. Forensic analysis is made a lot easier as no other systems are dependant on the Pseudoserver. Also logs are easier to read as there is no authorised traffic to the machine. The nightmare that we all go through of millions of alerts and log entries can be simplified in this case. The Pseudoserver can provide an executive summary of unauthorised network activity without the disadvantages of the Honeypot mentioned above. The extra cost to the organisation is small as a Linux box running SNORT will suffice (depending on the network). The art of running a Pseudoserver will be in how well it can be made to look like a normal production server. This is not a problem as it WILL be a normal production server except that it isn’t legitimately being used; therefore any sign of use must be from an illegitimate source.

Back to our list. Disadvantage number 1 has already been removed. Now as regards privacy (Disadvantage number 2), the hacker's right to privacy will be lowered as they have not been enticed onto the machine. I have not asked them, attracted them or even wanted them on the machine. As long as the machine keeps quiet and I know it is working then as admin I am happy. The hacker has broken in against my wishes unlike a Honeypot where they are virtually invited. There will also be less exploits ran on the machine therefore less of the function is to snoop on hackers more to alert their presence. Who wants to listen to hackers anyway? We have better things to do, right? Well actually No. I am very interested in what the hackers are saying to each other. As an IT security researcher this information is gold and is the great benefit of a Honeypot that is wired to listen to the Hackers IRC. However If I am on a production network I most probably just want them to go away. Bleeding edge research and commercially sensitive production networks do not mix.

Disadvantage three that of attracting attention is also lost as there is no purposeful weakness advertised and zero interaction. This minimises the risk of stirring up a hornets nest (excuse the pun).

Disadvantage four is removed as I cannot be held responsible for a further hack from this Pseudoserver to another machine as it is patched.

Disadvantage five is also removed as I am not leaving myself open to a hacker claiming a successful hack when in fact it was a Honeypot that was deliberately left open to them.

The Pseudoserver concept can be extrapolated into other areas much the same as the Honey concept into Pseudonetworks or Pseudotokens. Exactly the same as a Honeynet/Honeytoken but without the enticement and without the associated disadvantages for production networks. Of course Honeypots are going to be better if you want to have prolonged interaction with Hackers which is certainly the case in the research area. My company is contemplating running research Honeypots separate to the production network to provide valuable intelligence. This paper proposes the use of Pseudoservers in the production network and Honeypots in research networks that are not legally part of the companies network. Research labs and employees home networks are great places for Honeynets and a way for employees to increase there value to the company if they can provide intelligence about the hacker community whilst taking the disadvantages away from the company.

This paper is a small conceptual contribution I think but maybe quite important as I know of many large institutions that really want to install Honeynets but are held back by the disadvantages I have mentioned above. Pseudoservers could provide an answer. Even if the difference is as subtle as a terminology change and the removal of enticement I think it is a useful concept to the development of hacker control technologies. I have been following the Honeynet project now for over a year and am impressed by the contribution made by the line up of expertise that is involved. I have implemented Honeypots and Pseudoservers here on my network now for a while and plan to produce part 2 of this paper soon which will have more about the practical implementation and results from my work.

I would appreciate feedback on this paper to paul.wright@qoda.com