

# HoneyPot Farms

*Lance Spitzner* 2003-08-13

For the past six months this [series](#) of papers has covered a breadth of honeypot topics. We have covered everything from what honeypots are, their value and different types, to common misconceptions and legal issues. However, one thing we have yet to discuss is deployment. How can you deploy honeypots in your environment? For small organizations, this may be easy -- nothing more than installing a honeypot on a single computer and placing it on your local network. But what about organizations with hundreds of networks and thousands of computers? How can honeypots be easily deployed and managed in such large, distributed environments? One approach is that you don't. Instead, you simply consolidate all of your honeypots in a single honeypot farm, then you let the bad guys come to you.

## The Problem

For the past couple of years, honeypots have repeatedly demonstrated their value. Everything from production honeypots used to [detect insider attacks](#) to research honeypots capturing [automated credit card fraud](#). Honeypots have many unique advantages, including dramatically reducing if not eliminating false positives, working in IPv6 and encrypted environments, and the ability to capture new or unknown behavior. Because of these advantages, organizations are beginning to deploy honeypots, and in large numbers. However, with these deployments can come challenges.

One weakness of honeypots is they have a limited field of view, in that they only see activity that interacts with the honeypot. Unlike IDS sensors, honeypots do not passively monitor and capture all network traffic. Instead, the bad guys have to probe, use or communicate with the honeypot for it to work. This means for a honeypot to monitor the activity on a network, the honeypot should be deployed on that network. There are methods you can use to direct unauthorized activity to a honeypot, such as [honeytokens](#), hot zoning, and [bait-n-switch](#). However, in general, the more honeypots you deploy, and the more IP's you monitor, the greater the chance you will detect or capture unauthorized activity. Many large organizations have hundreds, if not thousands of networks distributed around the world. For honeypots to be effective for such large organizations, you would most likely have to deploy multiple honeypots. Unfortunately, like most technologies, the more honeypots you deploy, the more resources are required. Humans are needed to configure, deploy, and maintain the solution. The more honeypots you deploy, the more manpower is required to administer them. This in fact has

been one of the greatest challenges with most technologies, such as IDS sensors or firewall gateways. With firewalls, administrators may have to connect to systems around the world, update hundreds of rules, and review overwhelming firewall logs. With IDS sensors, admins have to add new signatures and review thousands of alerts. Distributed honeypots could face many of the same challenges.

The challenge becomes greater for high-interaction honeypots such as [Honeynets](#) . Low-interaction honeypots, such as [KFSensor](#), [Honeyd](#), and [Specter](#) are in many ways even easier to deploy than IDS sensors or firewalls. For these simple solutions, you install the software, deploy the honeypot, and sit back and wait. These systems are simple to monitor in that they capture very little data, but it is data of very high value. They are simple to maintain as there are no rulebases, signatures, or modifications that have to be made on a daily or weekly basis. All three solutions mentioned here also have remote administration and logging capabilities. High-interaction honeypots, while far more powerful and flexible, can also be more complex and time consuming. Think about it, every HoneyNet is its own separate network of isolated systems. Not only does each of these honeypots within the HoneyNet have to be built and maintained just like real systems, but the HoneyNet architecture, specifically the [Honeywall gateway](#), are complex security mechanisms requiring advanced configuration.

Once deployed, high-interaction honeypots require extensive tender loving care. In fact, such a solution may require its own, dedicated team. These folks ensure that the high-interaction solutions are safely maintained, and while capturing extensive information, cannot be used to harm non-honeypot systems. Also, because high-interaction solutions can capture so much data, once compromised administrators may be overwhelmed with information to analyze. If organizations intend to conduct full forensic analysis of compromised, high-interaction honeypots, they can expect up to thirty hours of work for every thirty minutes an attacker spends on the system. Now imagine taking such a solution and deploying hundreds, if not thousands of high-interaction honeypots throughout a large organization. While such a deployment would make an extremely powerful detection and information collecting solution, it could be cost prohibitive and time consuming. For deploying multiple, distributed honeypots, especially high-interaction solutions, we should have some alternative options for deployment.

## Farming - A Possible Solution

There is one possible solution to simplifying large honeypot deployments, honeypot farming. The concept of farming is simple. Instead of deploying large numbers of honeypots, or

honeypots on every network, you simply deploy your honeypots in a single, consolidated location. This single network of honeypots becomes your honeypot farm, a dedicated security resource. Attackers are then redirected to the farm, regardless of what network they are on or probing. Remember, honeypots don't passively capture traffic from your network, so they don't have to be physically connected to the network. Honeypots only have to be virtually on your network. Honeypots farms do this by deploying redirectors. A redirector acts as a proxy or 'worm hole', it transports an attacker's probes to a honeypot within the honeypot farm, without the attacker ever knowing it. The attacker thinks they are interacting with a victim on a local network, when in reality they have been transported to your honeypot farm. Figure A below demonstrates the concept of redirecting attackers to honeypot farms.

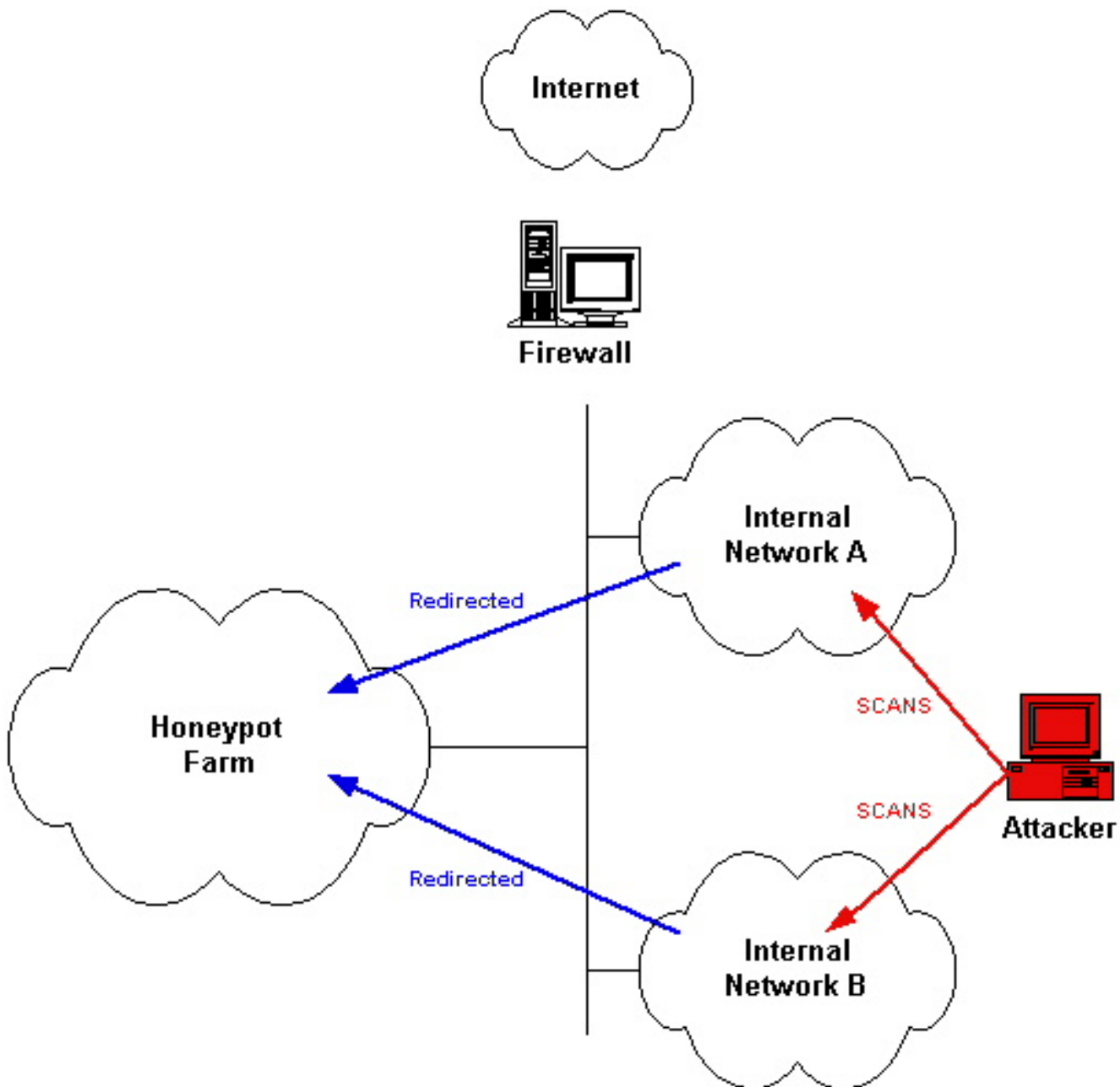


Figure A: HoneyPot Farm

The potential advantages are enormous. Deploying honeypots becomes an extremely simple affair. Instead of having individuals all over the world build, customize, deploy, and maintain a separate honeypot for your every network (keeping in mind many organizations have thousands of networks), you build and deploy a single, centralized honeypot farm. This honeypot farm could become part of your SOC (Security Operations Center) where you have the manpower and resources already dedicated to build such a solution. You then have the trained professionals on site that can deploy and maintain the farm. Instead of having honeypots distributed all over the world, you have them all in one place. The farm can be as simple as several low-interaction honeypots, to something as involved as a large Honeynet made up of hundreds of real systems and applications waiting to be attacked. This centralization then makes maintaining, standardizing, and analyzing the honeypots far simpler. To 'virtually' deploy your honeypots on other networks, you simply mail a redirector to each network administrator you want to have your honeypots on. These redirectors, once physically placed on a network, will then redirect all attackers or unauthorized activity to the centralized honeypot farm, where SOC personnel will monitor and analyze all of the captured information in real time. To deploy the redirectors, you simply ship one to each network admin for each network you want monitored. These redirectors can be nothing more than a black box, one which the network administrators simply plug into their local networks. No configuration is required, as all configuration was done by the SOC personnel. That black box can then monitor predetermined IP's (or dynamically monitor unused IP's). When attackers interact with those IP's, or interact with systems in a malicious or unauthorized manner, the black-box magically transports the attacker to the honeypot farm. This simplifies deploying distributed honeypots to nothing more than FedEx'ing a box to your network admins.

Updating and administering your honeypots, especially high-interaction honeypots, also far becomes easier. Instead of reaching out to each remote system, security admins merely update the honeypots in a single location. Instead of having to maintain multiple Honeynets distributed around the world, they have only one physical Honeynet to maintain, saving them a great deal of time. This time can then be used to develop more advanced honeypots that mirror your environment. For example, instead of having a honeypot emulate a Solaris box with a SQL emulator, you now have the time and resources to deploy a real Solaris honeypot running Oracle. You can then even populate the database with bogus information, to see how attackers interact with the database, and what data they attempt to recover. Honeypots farms can exponentially increase the effectiveness of honeypots, especially high-interaction solutions such as Honeynets.

A third advantage becomes one of mitigating risk. High-interaction honeypots have a great deal of inherent risk in them. The risk being that once an attacker takes over one of the honeypots, they can then use that honeypot to attack other non-honeypot systems. This risk is compounded when multiple high-interaction honeypots are deployed. HoneyPot farms can help address this. By consolidating all of your high-interaction honeypots to a single farm, you mitigate risk by having all of your honeypots in a single location. This means you require only a single place for data control, where you contain the attacker's inbound and outbound activity. Your security team can then monitor this single network, as opposed to having to monitor and administer multiple honeypots. This also allows them to ensure they have the most current technologies and solutions in place.

## Implementing HoneyPot Farms

The concept of honeyPot farms is extremely new and powerful, however few off the shelf solutions exist. The concept of redirectors working with honeyPot farms has many challenges to address. For example, the redirectors have to transport an attacker from one network to another (the honeyPot farm) without the attacker knowing it. What activity should the redirectors transport, to which honeypots within the honeyPot farm, and how? Do you simply monitor unused IP space, and redirect all activity to the farm, or do you only monitor specific high value IPs? How do you ensure that what the attacker probes is similar to the honeypot they are directed to? These and many other questions will have to be addressed by honeyPot farming technology.

One of the most basic examples of such a capability would be to utilize the proxying feature of Honeyd. As discussed earlier in this series, Honeyd is an OpenSource honeypot that has the capability to monitor your network's unused IP space. Any connection to an unused IP is assumed hostile and intercepted by Honeyd, which then interacts with the attacker. One of the options of Honeyd is to proxy the intercepted connection and forward it to another IP address. Utilizing this proxy feature, the attacker's connections could then be forwarded to a centrally located honeyPot farm, in this case one that supports a HoneyNet. We now have potentially combined the advantages of Honeyd (a low-interaction honeypot) with a HoneyNet (a high-interaction honeypot). Honeyd monitors all unused IP space, which could be thousands of IP addresses. By redirecting any connection to an unused IP to the HoneyNet, we dramatically increase the likelihood of detecting and capturing the attacker's activities. In effect, we have virtually deployed hundreds of HoneyNets, but physically only deployed one. By redirecting this

activity to a Honey-net, we can learn far more than Honeyd could, as the Honey-net provides real applications and operating systems for the attacker to interact with. Of course, there are still many issues to deal with, such as will the attacker notice the redirection and will the honeypot the attacker interacts with behave as the system the attacker thought they were probing. This example demonstrates some of the potential advantages and challenges facing honeypot farms.

Other honeypot farm solutions are under active development or have already been released. One example is [NetBait](#). NetBait is a commercial solution that implements honeypot farms (which they call ServerFarms). Within these farms you can place any systems you want. They have redirectors that will take an attacker's activity and redirect them to pre-determined systems within the ServerFarm. An attacker probes or attacks a specific IP. That attacker continues to interact with that same IP, however the attacker does not realize the system they are interacting with physically resides in the ServerFarm. NetBait even take this concept a step further, offering a honeypot farm as a service. They will maintain a farm for an organization. All the organization has to do is deploy redirectors on their networks that direct all unauthorized activity to NetBait's farms. Instead of becoming a tool, honeypot farms become a service. Organizations no longer have to maintain or analyze the data from honeypots, nor worry about liability or risk. They can gain the power and advantages of honeypots, without resource or risk issues.

## Conclusions

Honeypot farms an exciting new concept with tremendous potential. They represent one of the newest methods for large deployments of distributed honeypots, especially high interaction solutions such as Honey-nets. Expect to see a great deal of more development in this new field. Next month we will discuss what I consider to be the perfect honeypot, the dynamic appliance.

### Author Credit

View [more articles by Lance Spitzner](#) on SecurityFocus.

[Privacy Statement](#)

Copyright 2006, SecurityFocus