

# Honeypots: Definisi dan Nilai dari Honeypots

**Wahyu Wijanarko**

wahyu@wahyu.com  
<http://wahyu.com>

## ***Lisensi Dokumen:***

*Copyright © 2005 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

## **Abstrak:**

Honeypots adalah salah satu teknologi baru yang membuat gempar karena potensinya yang besar dalam komunitas keamanan jaringan komputer. Konsep yang pertama kali diperkenalkan oleh beberapa orang di bidang keamanan komputer, terutama Cliff Stoll dalam buku *The Chuckoo's Egg*, dan dalam *paper* Bill Cheswick "An Evening with Berferd". Sejak saat itu, honeypots telah melanjutkan perkembangannya, dan dibangun menjadi alat keamanan yang kuat yang ada saat ini. Tujuan dari tulisan ini adalah untuk menjelaskan secara tepat tentang honeypots, keuntungan dan kerugiannya, dan nilainya untuk sistem keamanan.

**Keywords:** keamanan jaringan komputer

## **1. PENDAHULUAN**

Langkah pertama untuk mengerti tentang honeypots adalah mendefinisikan arti dari honeypots itu sendiri. Ini dapat menjadi lebih sulit daripada kedengarannya. Tidak seperti *firewall* atau *Intrusion Detection System (IDS)*, honeypots tidak menyelesaikan masalah secara khusus. Di dalamnya, sistem ini merupakan perangkat yang sangat fleksibel yang dapat berada dalam berbagai bentuk dan ukuran. Ia dapat melakukan semuanya, mulai dari mendeteksi serangan terenkripsi pada jaringan IPv6 sampai menangkap data siapa yang online pada saat terjadi pemalsuan kartu kredit. Ini

adalah fleksibilitas yang memberikan kekuatan sesungguhnya dari honeypots. Fleksibilitas ini jugalah yang menantang kita untuk dapat mendefinisikan dan mengerti tentang sistem ini. Lance Spitzner ([www.tracking-hackers.com](http://www.tracking-hackers.com)) mendefinisikan honeypots sebagai berikut:

**Suatu honeypots merupakan sumber sistem informasi yang menghasilkan nilai palsu pada saat terjadi penggunaan sumber daya yang tidak sah atau tidak diijinkan.**

Ini merupakan definisi umum yang dapat mencakup semua pernyataan berbeda mengenai honeypots. Berdasarkan pengertian ini dapat dijelaskan lebih lanjut bahwa sistem dengan

honeypots akan ‘menipu’ atau memberikan data palsu apabila ada orang yang memiliki maksud yang tidak baik ketika ia masuk ke suatu sistem informasi. Secara teori, honeypots tidak akan mencatat trafik yang legal. Sehingga bisa dilihat bahwa yang berinteraksi dengan honeypots secara kebanyakan adalah user yang menggunakan sumber daya sistem secara *illegal*. Jadi honeypots seolah-olah menjadi sistem yang ‘berhasil disusupi’, oleh orang jahat, padahal penyerang tidak masuk ke sistem sebenarnya, tetapi malah masuk ke sistem yang palsu.

Penggunaan suatu sistem tentu saja menimbulkan berbagai macam keuntungan dan kerugian. Keuntungan dan kerugian dari penggunaan honeypots adalah sebagai berikut:

## 2. KEUNTUNGAN HONEYPOTS

Honeypots merupakan suatu konsep yang sederhana, yang memiliki kekuatan yang luar biasa.

*Kumpulan data kecil dengan nilai yang tinggi:* Honeypots mengumpulkan beberapa informasi kecil. Sebagai gambaran, misalnya jika sistem melakukan *logging* 1 GB data perhari, honeypots cukup melakukan *logging* dengan data sekitar 1 MB. Apabila sistem biasanya mengeluarkan *alert* sebanyak 10000 kali perhari, maka honeypots hanya mengeluarkan 10 kali *alert*. Yang perlu diingat, honeypots hanya menangkap data pada aktifitas yang ‘jahat’ atau aktifitas yang dicurigai atau yang tidak diijinkan. Dapat dilihat bahwa sistem ini lebih sederhana dan lebih murah untuk dapat dianalisa, karena data yang dianalisa cukup kecil.

*Taktik dan tools yang baru:* Honeypots dirancang untuk dapat menangkap taktik dan tools baru yang digunakan penyerang yang belum pernah diketahui sebelumnya.

*Sumber daya minimal:* Honeypots hanya memerlukan sumber daya yang sangat kecil. Ini berarti komputer pentium yang tua dengan memory 128 MB sudah cukup untuk dapat menghandle dengan mudah suatu jaringan kelas B pada jaringan OC-12.

*Enkripsi atau Ipv6:* Tidak seperti kebanyakan

teknologi keamanan (misal seperti sistem IDS), honeypots dapat bekerja dengan baik pada data terenkripsi atau lingkungan IPv6. Jadi tidak masalah jika ada orang yang tidak baik masuk ke honeypots, maka honeypots akan dapat mendeteksi dan merekamnya.

*Informasi:* Honeypots dapat mengumpulkan informasi yang di bawahnya, apabila ada teknologi lain yang sesuai dengannya.

*Lebih sederhana:* Terakhir, konsep dari honeypots sangat sederhana. Tidak ada algoritma khusus yang perlu dibuat, tabel yang harus dimaintain, atau tanda tangan (digital) yang perlu diupdate. Kesederhanaan suatu teknologi, berkurangnya kemungkinan terjadi kesalahan atau miskonfigurasi.

## 3. KERUGIAN HONEYPOTS

Seperti teknologi pada umumnya, honeypots juga memiliki berbagai kelemahan. Hal ini dikarenakan ia tidak menggantikan teknologi yang ada, tetapi bekerja pada teknologi yang ada saat ini.

*Terbatasnya view:* Honeypots hanya dapat merekam dan menangkap aktifitas yang secara langsung berinteraksi dengannya. Honeypots tidak akan merekam serangan terhadap sistem lain, selama penyerang tidak berinteraksi juga dengan honeypots pada saat terjadi serangan.

*Resiko:* Semua teknologi keamanan memiliki resiko. Firewall memiliki resiko untuk dapat dipenetrasi, enkripsi memiliki resiko untuk dapat dibobol, sensor IDS memiliki resiko kesalahan dalam mendeteksi serangan. Honeypots tidak memiliki perbedaan, ia juga memiliki kelemahan. Secara spesifik, resiko honeypots adalah ketika dia dapat diambil alih oleh penyerang atau orang yang bermaksud tidak baik untuk digunakan menyerang sistem yang lain. Resiko ini bervariasi untuk honeypots yang berbeda-beda. Tergantung dari tipe honeypots itu sendiri, ia dapat memiliki resiko yang tidak lebih besar dari sensor IDS, dan banyak juga beberapa honeypots yang memiliki resiko tinggi.

#### 4. TIPE DAN JENIS HONEYPOT

Honeypots ada dalam banyak sekali bentuk dan ukuran, sehingga sangat banyak macamnya. Untuk lebih memudahkan, kira akan membagi honeypots dalam 2 tipe, yaitu *honeypots interaksi rendah* dan *honeypots interaksi tinggi*. Kategori ini akan mempermudah kita untuk dapat melihat kekuatan dan kelemahan dari berbagai macam honeypots. Interaksi mendefinisikan tingkat aktifitas yang diijinkan oleh honeypots kepada penyerang. Honeypots interaksi rendah bekerja dengan kemampuan terbatas, biasanya bekerja dengan cara mengemulasikan servis atau mengemulasikan suatu sistem operasi. Misalnya suatu honeypots mengemulasikan FTP server pada port 21 dan mendukung beberapa perintah pada koneksi FTP. Keuntungan dari sistem interaksi rendah adalah kesederhanaannya, karena dapat dengan mudah dibangun dan dimaintain dengan resiko minimal. Biasanya cukup dengan menginstall software, memilih sistem operasi apa yang akan diemulasikan dan dimonitor, dan memulai mengaktifkan penggunaan honeypots. Dengan sistem ini akan lebih mudah dalam penggunaannya atau implementasinya. Kelemahan utama dari honeypots interaksi rendah adalah sistem ini hanya merekam informasi log yang terbatas dan hanya didesain untuk menangkap aktifitas yang sudah didefinisikan atau diketahui sebelumnya. Servis teremulasi hanya dapat berinteraksi secara terbatas. Selain itu, penyerang juga sangat mudah dalam mendeteksi suatu honeypots berinteraksi rendah, tidak masalah bagaimanapun bagus emulasi, penyerang yang sudah berpengalaman dapat mendeteksi bahwa dia telah masuk ke honeypots. Contoh dari honeypots berinteraksi rendah adalah Specter, Honeyd, dan KFSensor.

Honeypots berinteraksi tinggi berbeda, karena biasanya solusi kompleks terdapat pada sistem ini, karena honeypots benar-benar merupakan sistem operasi atau aplikasi yang nyata. Tidak ada yang diemulasikan, karena kita memberikan kepada penyerang sesuatu yang nyata. Misal ketika anda ingin membangun honeypots Linux yang menjalankan FTP server, maka kita juga harus benar-benar membangun sistem Linux

yang juga benar-benar menjalankan server FTP. Manfaat atau keuntungan dari sistem ini ada 2 macam. Pertama, anda dapat menangkap banyak informasi ekstensif. Dengan memberikan kepada penyerang suatu sistem yang nyata, anda dapat mempelajari tingkah laku para penyerang tersebut, mulai dari segala sesuatu tentang *rootkits*, sampai sesi IRC internasional yang mereka gunakan. Manfaat kedua dari sistem ini adalah tidak perlu adanya asumsi apa yang akan dikerjakan oleh penyerang. Di dalamnya honeypots tipe ini memberikan lingkungan terbuka yang menangkap semua aktifitas. Dengan cara mengijinkan solusi dengan interaksi tinggi, maka kita dapat belajar tingkah laku yang tidak kita harapkan. Contoh bagus dari hal ini adalah tentang bagaimana HoneyNet menangkap perintah backdoor yang terencode pada protokol IP yang tidak standar. (khususnya protokol IP 11, *Network Voice protocol*) . Bagaimanapun juga, hal ini juga dapat meningkatkan resiko dari honeypots yang digunakan untuk menyerang sistem lain yang bukan honeypots, ketika sistem honeypots berhasil diambil alih oleh penyerang. Sebagai penyelesaiannya, teknologi tambahan telah diimplementasikan untuk mencegah penyerang supaya tidak dapat merusak sistem lain yang bukan merupakan honeypots. Pada umumnya, honeypots dengan tingkat interaksi tinggi dapat melakukan semua yang bisa dilakukan oleh honeypots berinteraksi rendah, dan dapat melakukan lebih banyak lagi fungsi lain. Bagaimanapun juga, honeypots jenis ini lebih kompleks dalam maintain dan penggunaannya. Contoh dari honeypots berinteraksi tinggi adalah *Symantec Decoy Server* dan *HoneyNets*.

#### 5. NILAI KEGUNAAN HONEYPOTS

Untuk lebih mengetahui nilai kegunaan dari honeypots, kita akan mengamati tujuan penggunaannya. Honeypots sendiri memiliki tujuan, yaitu untuk kepentingan produksi, maupun untuk keperluan penelitian.

Dalam dunia produksi, honeypots digunakan untuk menjaga keamanan perangkat lunak dari suatu badan atau organisasi. Hal ini termasuk dalam mencegah, mendeteksi, atau membantu

suatu badan atau organisasi (misalnya perusahaan) dalam merespon suatu serangan pada jaringannya. Untuk kepentingan penelitian, honeypots digunakan untuk mengumpulkan informasi. Informasi ini berbeda-beda untuk tiap peneliti, tergantung dari kebutuhannya. Ada yang menggunakan untuk mempelajari trend dari aktivitas seorang *cracker*. Ada juga yang menggunakan untuk membuat semacam peringatan awal, jika akan terjadi suatu serangan yang lebih besar. Untuk kepentingan produksi, biasanya digunakan honeypots dengan interaksi rendah, sedangkan untuk kepentingan penelitian biasanya digunakan honeypots dengan tingkat interaksi tinggi. Untuk lebih jelasnya kita akan melihat kegunaan masing-masing dari honeypots.

Pertama, honeypots dapat digunakan untuk mencegah serangan dalam banyak cara. Ia bisa mencegah serangan otomatis, seperti *worms* maupun *autorooters*. Serangan ini berbasis pada *tools* yang secara acak melakukan *scanning* pada suatu jaringan, untuk mencari system yang *vulnerable*. Apabila system yang *vulnerable* ditemukan, maka *tools* (worm) ini akan menyerang dan mengambil alih system, dengan mereplikasikan dirinya atau mengcopy dirinya ke system yang berhasil diserang. Salah satu cara honeypots dapat membantu bertahan dari serangan adalah dengan cara memperlambat laju serangan, bahkan kalau bisa system honeypots akan menghentikannya. Disebut sebagai *sticky honeypots*, system dengan solusi ini memonitor IP yang tidak terpakai. Ketika mengetahui adanya semacam aktivitas *scanning*, maka honeypots akan berinteraksi aktivitas ini dan memperlambat serangan. Honeypots melakukan hal ini menggunakan banyak variasi dari trik TCP, seperti *Windows size of zero*, ataupun meletakkan penyerang ke dalam pola tertahan. Cara ini sangat bagus untuk mencegah penyebaran worm ke dalam system kita. Honeypots juga dapat mencegah system kita dari *human attackers*. Cara kerjanya adalah dengan membuat bingung penyerang, karena akan membuat penyerang menghabiskan waktu yang lebih lama untuk menyerang honeypots, sementara dalam waktu tersebut kita sebagai system administrator dapat secepatnya mengetahui kalau sedang ada serangan, sehingga

kita dapat secara cepat menangani serangan ini. Paling tidak bila penyerang tahu kalau di system yang dimiliki korban terdapat honeypots, maka ia akan menjadi bingung, karena sulit untuk membedakan system mana yang merupakan honeypots, atau system mana yang merupakan target sesungguhnya. Contoh honeypots yang disesain untuk hal ini adalah *Deception Toolkit*, salah satu honeypots dengan tingkat interaksi rendah.

Cara ke dua di mana honeypots dapat membantu kita mencegah suatu serangan adalah dengan melalui cara deteksi. Deteksi ini bersifat *critical*, dan tujuannya untuk mengidentifikasi kesalahan pada saat terjadi kegagalan dalam pencegahan. Tidak peduli seaman apapun jaringan kita, pasti memiliki suatu cacat, seandainya tidak ada alasan lain jika kemudian orang ikut masuk dalam proses rumit ini. Dengan cara mendeteksi serangan, maka kita akan dapat segera bereaksi dan mengantisipasinya, menghentikannya, atau mengurangi efek serangan. Jika kita menggunakan system sensor IDS, akan tidak efektif, karena menghasilkan terlalu banyak data yang harus dianalisa, padahal data yang diberikan kebanyakan adalah data yang salah, dan selain itu IDS sulit diimplementasikan pada jaringan IPv6 atau lingkungan terenkripsi. Honeypots mampu dalam menyelesaikan masalah seperti ini. Honeypots dapat mengurangi terjadinya kesalahan dalam mengidentifikasi suatu serangan, karena hanya mengambil beberapa kumpulan data penting yang diperlukan saja, menangkap serangan yang tidak dapat teridentifikasi, seperti *exploits* terbaru, atau *polymorphic shell code*. Honeypots berinteraksi rendah sangat efektif dalam melakukan suatu deteksi pada system jaringan.

Yang terakhir, cara honeypots membantu kita dalam pengamanan jaringan kita adalah pada respon yang dimilikinya. Pada saat jaringan kita mendeteksi adanya kesalahan, biasanya system akan segera merekam informasi tentang siapa penyerang, bagaimana penyerang masuk, dan seberapa besar kerusakan yang diakibatkan oleh serangan tersebut. Sistem yang paling sering diserang adalah mail server. Ketika suatu mail server diserang, tidak mungkin administrator system mematikan mail server untuk melakukan

analisa terhadap serangan yang terjadi, karena akan menghambat trafik komunikasi email, yang dapat berakibat terganggunya proses produksi (misalnya server email di dalam perusahaan, jika server mati dapat berakibat terganggunya proses produksi). Selain itu akan sangat sulit mendeteksi secara langsung dalam membedakan, yang sedang melakukan login ke mail server user yang benar atau penyerang. Honeypots dapat melakukan logging hanya pada aktivitas yang bersifat serangan saja, dan selain itu juga bisa dibuat *offline* (dimatikan) untuk dianalisis, tanpa mengganggu proses produksi.

Sampai pada point ini kita membicarakan honeypots sebagai penunjang pada kebutuhan produksi. Untuk penelitian, honeypots digunakan dalam pengembangan system keamanan jaringan dengan merekam aktivitas-aktivitas terbaru dari para penyerang.

## 6. KESIMPULAN

Ada 2 macam honeypots, yaitu honeypots berinteraksi rendah dan honeypots berinteraksi tinggi. Interaksi mendefinisikan seberapa jauh honeypots mengijinkan masuknya penyerang ke sistem. Solusi honeypots dapat digunakan dalam proses produksi maupun adalam penelitian.

## 7. REFERENSI

Lance Spitzner, Honeypots, Definitions and Value of Honeypots

<http://www.tracking-hackers.com/papers/honeypots.html>

## BIOGRAFI PENULIS



**Wahyu Wijanarko.** Lahir di Kulonprogo, 7 Januari 1984. Menamatkan SMTA di SMU N 1 Bantul pada tahun 2001. Saat ini menjadi mahasiswa di Jurusan Teknik Elektro Fakultas Teknik Universitas Gadjah Mada Yogyakarta, dengan konsentrasi Sistem

Komputer dan Informatika.

Menjadi anggota Open Source Initiative UGM, dan juga ikut serta dalam pengembangan sistem komputer UGM di UPT Pusat Komputer UGM. Menjadi anggota di berbagai milis komputer Indonesia, dan juga beberapa forum webmaster.

Berpengalaman di dalam pengembangan berbagai sistem informasi berbasis web maupun GUI, dan desain database dengan menggunakan software opensource, serta administrasi server berbasis GNU/Linux.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

URL: <http://wahyu.com/>

Email: [wahyu@wahyu.com](mailto:wahyu@wahyu.com)