HoneyPot – Trap for penetrating attacks

* Information system resource
* Used to expressly set up to **attract & trap** people who attempt to penetrate an organization’s network
* **No** **authorized activity**, doesn’t have any **production value**
* **Any traffic to it is likely to be a probe, attack or compromise**
* Honeypot can **log port access** attempts or monitor an **attacker’s keystrokes.**  These could be early warning of an more concerted attack



# Types of Honeypot

## Based on Interaction

### Low interaction Honeypot:

Simulates only **limited services & applications of a target network or systrm**

### Medium-interaction

Simulates **real OS, Applications & services of target n/w**

### High-interaction

Simulate **all services and applications of target n/w**

### Pure Honeypots

Emulate THE real production n/w of a target organization

### Production Honeypots

* Deployed inside production n/w of org. along with other production servers
* They also helps to find out internal flaws & attackers within an org,

### Research Honeypots:

* High-interaction honeypots
* Primarily deployed by research institutes, govts., or military orgs., to gain detailed knowledge about the actions of intruder

## Based on Deception Technology:

### Malware :

To **trap malware campaigns or attempts** over the network infrastructure

### Database

**Employ fake databases that are vulnerable to database attacks like SQL Injection & database enumeration**

### Spam

Specifically target spammers who abuse vulnerable resources such as open mail relay and open proxies

### Email

Fak email addresses that are specifically used to attract fake & malicious emails fpr adversaries

### Spider

To trap web crawlers & spiders

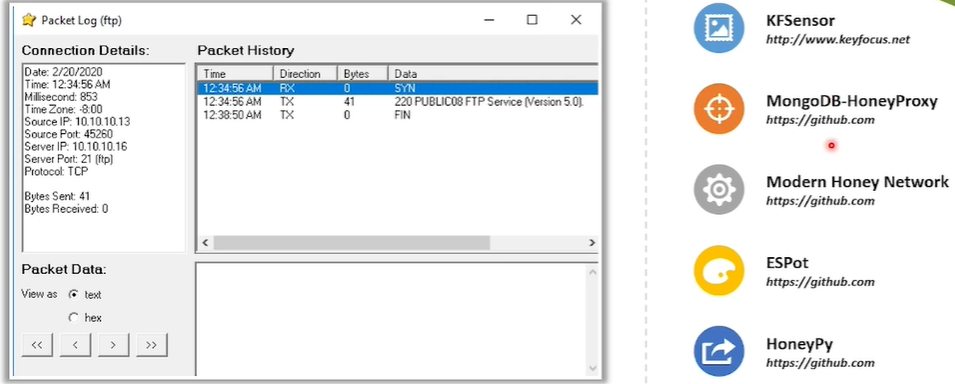
### Honeynets

* Networks pf honeypots
* Effective in determining entire capabilities of adversaries

# Honeypot Tools

## Honey BOT

* It is a medium interaction honeypot for windows
* Easy- to-use solution
* Ideal for network security research



## Top 20+ Honeypots for Identifying Cybersecurity Threats

There are as many honeypots as there are types of software running, so creating a definitive list would be quite difficult. On this list we've included some of the most popular honeypot tools that are, in our own experience, a must for all blue and purple teams.

### [¶](https://securitytrails.com/blog/top-honeypots#content-ssh-honeypots)SSH honeypots

* [**Kippo**](https://github.com/desaster/kippo): This SSH honeypot written in Python has been designed to detect and log brute force attacks and, most importantly, the complete shell history performed by the attacker. It's available for most modern Linux distros, and offers both cli-command management and configuration, as well as web-based interface. Kippo offers a fake file system and the ability to offer fake content to attackers (such as user password files, etc.), as well as a powerful statistics system called Kippo Graph.
* [**Cowrite**](https://github.com/micheloosterhof/cowrie): This medium interaction SSH honeypot works by emulating a shell. It offers a fake file system based on Debian 5.0, letting you add and remove files as you wish. This application also saves all the downloaded and uploaded files in a secure and quarantined area, so you can perform later analysis if needed. Apart from the SSH emulated shell, it can be used as an SSH and Telnet proxy, and allows you to forward SMTP connections to another SMTP honeypot.

### [¶](https://securitytrails.com/blog/top-honeypots#content-http-honeypots)HTTP honeypots

* [**Glastopf**](http://glastopf.org/): This HTTP-based honeypot lets you detect web-application attacks effectively. Written in Python, Glastopf can emulate several types of vulnerabilities, including local and remote file insertion as well as SQL Injection (SQLi) and using a centralized logging system with HPFeeds.
* [**Nodepot**](https://github.com/schmalle/Nodepot): This web-app honeypot is focused on Node.js, and even lets you run it in limited hardware such as Raspberry Pi / Cubietruck. If you're running a Node.js app and are lookingto get valuable information about incoming attacks and discover how vulnerable you are, then this is one of the most relevant honeypots for you. Available on most modern Linux distros, running it depends on only a few requirements.
* [**Google Hack Honeypot**](http://ghh.sourceforge.net/): Commonly known as GHH, this honeypot emulates a vulnerable web app that can be indexed by web crawlers but remains hidden from direct browser requests. The transparent link used for this purpose reduces false positives and prevents the honeypot from being detected. This lets you test your app against ever-so-popular [Google dorks](https://securitytrails.com/blog/google-hacking-techniques). GHH offers an easy configuration file, as well some nice logging capabilities for getting critical attacker information such as IP, user agent and other header details.

### [¶](https://securitytrails.com/blog/top-honeypots#content-wordpress-honeypots)WordPress honeypots

* [**Formidable Honeypot**](https://es.wordpress.org/plugins/formidable-honeypot/): This is one of the most popular honeypots used with Wordpress. It's literally invisible to humans; only bots can fall into its trap, so once an automated attack comes into your form, it will be effectively detected and avoided. It's a non-intrusive way to defend Wordpress against spam. Conveniently, it doesn't require any configuration. Simply activate the plugin and it will be added to all the forms you use in Wordpress, in both free and pro versions.
* [**Blackhole for Bad Bots**](https://es.wordpress.org/plugins/blackhole-bad-bots/): This one created to avoid automated bots from using unnecessary bandwidth and other server resources from your site infrastructure. By setting up this plugin, you can detect and block bad bots, from automated malware attacks to spam and several types of adware attacks. This Wordpress honeypot works by adding a hidden link in the footer of all your pages. This way it isn't detected by humans, and catches only bad bots that are not following the robots.txt rules. Once a bad bot is caught, it will be blocked from accessing your website.
* [**Wordpot**](https://github.com/gbrindisi/wordpot): This is one of the most effective Wordpress honeypots you can use to enhance [Wordpress security](https://securitytrails.com/blog/is-wordpress-secure" \o "Is WordPress Secure?). It helps you detect malicious signs for plugins, themes and other common files used to fingerprint a wordpress installation. Written in Python, it's easy to install, can be handled from the command line smoothly, and includes a wordpot.conf file for easy honeypot configuration. It also allows you to install custom Wordpot plugins so you can emulate popular Wordpress vulnerabilities.

### [¶](https://securitytrails.com/blog/top-honeypots#content-database-honeypots)Database honeypots

* [ElasticHoney](https://github.com/jordan-wright/elastichoney): With Elasticsearch so frequently exploited in the wild, it's never a bad idea to invest in a honeypot specifically created for this type of database. This is a simple yet effective honeypot that will let you catch malicious requests attempting to exploit RCE vulnerabilities. It works by receiving attack requests on several popular endpoints such as /, /\_search and /\_nodes, and then responds serving a JSON response that is identical to the vulnerable Elasticsearch instance. All logs are saved in a file called elastichoney.log. One of the best things about it is that this honeypot tool is available for both Windows and Linux operating systems.
* [HoneyMysql](https://github.com/supriyo-biswas/HoneyMysql): This simple MySQL honeypot is created to protect your SQL-based databases. Written in Python, it works on most platforms and can be installed easily by cloning its GitHub repo.
* [MongoDB-HoneyProxy](https://github.com/Plazmaz/MongoDB-HoneyProxy): One of the most popular MongoDB honeypots, this is specifically a honeypot proxy that can run and log all malicious traffic into a 3rd party MongoDB server. Node.js, npm, GCC, g++ and a MongoDB server are required to get this MongoDB honeypot working properly. It can be run inside a Docker container or any other VM environment.

### [¶](https://securitytrails.com/blog/top-honeypots#content-email-honeypots)Email honeypots

* [**Honeymail**](https://github.com/sec51/honeymail): If you're looking for a way to stop SMTP-based attacks, this is the perfect solution. Written in Golang, this honeypot for email will let you set up numerous features to detect and prevent attacks against your SMTP servers. Its main features include: configuring custom response messages, enabling StartSSL/TLS encryption, storing emails in a BoltDB file and extracting attacker information such as source domain, country, attachments and email parts (HTML or TXT). It also provides simple yet powerful DDoS protection against massive connections.
* [**Mailoney**](https://github.com/awhitehatter/mailoney): This is a great email honeypot written in Python. It can be run in different modes such as open\_relay (logging all emails attempted to be sent), postfix\_creds (used to log credentials from login attempts) and schizo\_open\_relay (which allows you to log everything).
* [**SpamHAT**](https://github.com/miguelraulb/spamhat): This trap is designed to catch and prevent spam from attacking any of your email boxes. To get this working, make sure you have Perl 5.10 or higher installed, as well as some CPAN modules such as IO::Socket, Mail::MboxParser, LWP::Simple, LWP::UserAgent, DBD::mysql, Digest::MD5::File, as well as having a running MySQL server with a database called 'spampot'.

### [¶](https://securitytrails.com/blog/top-honeypots#content-iot-honeypots)IOT honeypots

* [**HoneyThing**](https://github.com/omererdem/honeything): Created for the Internet of TR-069 enabled services, this honeypot works by acting as a full modem/router running the RomPager web server and supports TR-069 (CWMP) protocol. This IOT honeypot is capable of emulating popular vulnerabilities for Rom-0, Misfortune Cookie, RomPager and more. It offers support for TR-069 protocol, including most of its popular CPE commands such as GetRPCMethods, Get/Set parameter values, Download, etc. Unlike others, this honeypot offers an easy and polished web-based interface. Finally, all the critical data is logged in a file called honeything.log
* [**Kako**](https://github.com/darkarnium/kako): The default config will run a number of service simulations in order to capture attacking information from all incoming requests, including the full body. It includes Telnet, HTTP and HTTPS servers. Kako requires the following Python packages to work properly: Click, Boto3, Requests and Cerberus. Once you're covered with the required packages, you can configure this IOT honeypot by using a simple YAML file called kako.yaml. All the data is recorded and is exported into AWS SNS, and flat-file JSON format.

### [¶](https://securitytrails.com/blog/top-honeypots#content-other-types-of-honeypots)Other types of Honeypots

* [**Dionaea**](https://github.com/DinoTools/dionaea): This low-interaction honeypot written in C and Python uses the Libemu library to emulate the execution of Intel x86 instructions and detect shellcodes. In addition, we can say it's a multi-protocol honeypot that offers support for protocols such as FTP, HTTP, Memcache, MSSQL, MySQL, SMB, TFTP, etc. Its logging capabilities offer compatibility with Fail2Ban, hpfeeds, log\_json and log\_sqlite.
* [**Miniprint**](http://miniprint/): With printers being some of the most overlooked devices within computer networks, Miniprint is the perfect ally when you need to detect and collect printer-based attacks. It works by exposing the printer to the Internet using a virtual file system where attackers can read and write simulated data. Miniprint offers a very deep logging mechanism, and saves any postscript or plain text print jobs in an upload directory for later analysis.
* [**Honeypot-ftp**](https://github.com/alexbredo/honeypot-ftp): Written in Python, this FTP honeypot offers full support for plain FTP and FTPS so you can perform a deep track of user and password credentials used in illegal login attempts, as well as uploaded files for every FTP/FTPS session.
* [**HoneyNTP**](https://github.com/fygrave/honeyntp): NTP is one of the most overlooked protocols on the Internet, and that's why it's a good idea to run an NTP Honeypot. This is a Python simulated NTP server that runs without a hitch on both Windows and Linux operating systems. It works by logging all the NTP packs and port numbers into a Redis database so you can perform later analysis.
* [**Thug**](https://github.com/buffer/thug): Thug isn’t a honeypot per se, but rather a honeyclient. Just as honeypot technologies enable research into server-side attacks, honeyclients take on client-side attacks. Acting as a complement to honeypots, Thug is a low-interaction honeyclient tool designed to mimic the behaviour of a web browser to analyze suspicious links and determine if they contain malicious components.
* [**Canarytokens**](https://github.com/thinkst/canarytokens): Canarytokens is a honeytoken tool created to emulate web bugs, the transparent images that track when someone opens an email by embedding a unique URL in the web page’s image tag and monitors GET requests. Canarytokens does the same thing but for file reads, database queries, process executions, patterns in log files and much more. It allows you to set up traps in your systems rather than setting up separate honeypots. In other words, attackers announce themselves as having breached your system by “tripping” over a token.

Extra tip: Don't forget to test [MHN](https://github.com/pwnlandia/mhn), which isn't actually a honeypot, but a centralized server for management and honeypot data collection. It includes a lot of the honeypots we mentioned here such as Glastopf, Dionaea, Cowrie and others.

Also important: remember, if you're setting up a honeypot in your live infrastructure, you're going to be exposed to a high level of incoming attacks—that's the very nature of honeypots. You'll be playing with fire. And it wouldn't be the first time we heard about someone who installed a honeypot on their production servers and then got hacked because smart bad guys were able to spoof and hide behind the legal network traffic.

## [¶](https://securitytrails.com/blog/top-honeypots#content-summary)