Honeynets: a tool for counterintelligence in online security

David Watson, UK Honeynet Project

Sun Tzu, the ancient Chinese general, summed up one of online security's basic principles over two millennia before electricity was invented. "If you know the enemy and know yourself, you need not fear the result of a hundred battles," he said.

Today's organizations, businesses and individuals spend millions each year on the latest defensive technologies, yet few have stepped back and attempted to understand the nature of the threats they face. How many of them know who their attackers are, and why they are being attacked?

The honeynet is a relatively new development in security technology that can help people to understand their attackers and how they work. A honeynet is a network of computer systems designed to attract attackers. When a honeynet system is attacked and compromised by a blackhat, every activity the attacker generates is captured, from keystrokes and downloaded toolkits to IRC messages and outbound emails. At the same time, potentially hostile outbound network activity is controlled.

Honeynet technology enables a system operator to replay details of an incident step by step, at a pace of their choosing. Through analysis of multiple attacks and increased education, operators can begin to understand the actors and motivations behind such activities and ultimately increase their defensive capabilities.

Honeynet origins

Much like antivirus software, firewalls and intrusion detection systems, the basic concepts underpinning modern honeynets appeared during the late 1980s. Seminal publications on honeynet principles include Clifford Stoll's *The Cuckoo's Egg*[1] and Bill Cheswick's paper *An evening with Berford*[2].

Interest from security professionals and researchers picked up in the late 1990s, with the release of a small number of basic commercial honeynet products and formation of the Honeynet Project in 1999. Today, honeynets are generally accepted as flexible and powerful security tools, with events such as Blackhat, CanSecWest and the annual honeynet track of IEEE's Security and Privacy workshop attracting the interest of many academic, governmental and commercial sources.

Honeypots

Honeypots are the core building block in all honeynets. The Honeynet Project defines a honeypot as an information systems resource whose value lies in its unauthorized or illicit use. Unlike traditional IT security systems such as firewalls or intrusion detection systems, which are designed to solve particular problems, honeypots don't typically address specific objectives. Instead, their value lies in how they are used by attackers. Honeypots involve aspects of prevention, detection, information gathering and more, and are powerful and flexible security tools.

Because there are no legitimate reasons to connect to a honeypot, any interaction is probably malicious. Consequently, honeypots dramatically reduce the number of false positive alerts compared with traditional network event-based security products such as intrusion detection systems. This high signal-to-noise ratio is useful for administrators often overwhelmed by false alarms, as the smaller data sets are relatively small and easy to manage and analyse. This results in shorter attack detection and incident response times.

Another advantage of honeypots over more traditional attack detection technologies is that the latter often depend upon signature matching or statistical models to identify attacks. This means that unknown or novel threats may not always be detected.

In contrast, honeypots are designed to capture all known and unknown attacks directed against them. Because any network activity related to the honeypot represents an anomaly, even the stealthiest activity will register on a honeypot's radar. Because honeypots operate at the host level, encrypted or non-IPv4 communications that can often blind traditional network based sensors can still be captured.

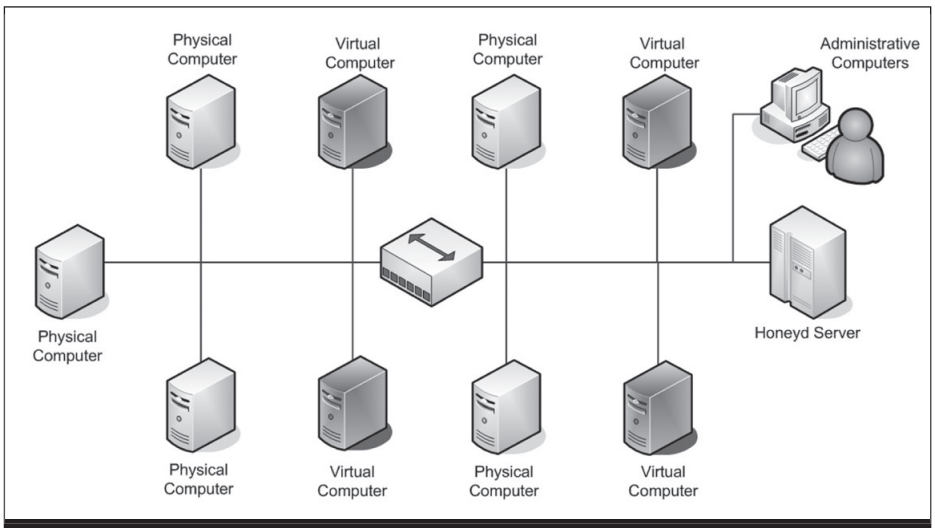
Honeypots are also very flexible, with a wide variety of deployment configurations that respond well to customization and will operate even with minimal resource availability (as real production services need not necessarily be provided).

Types of honeypot

Honeypots are often classified based on the amount of interaction granted to an attacker. Greater interaction allows an attacker freer reign and increases the level of information gathered, but it also increases the associated complexity and risk.

Low interaction honeypots (LIH) generally only emulate network services and host systems. LIH systems are generally limited to known threats, because emulated services don't usually respond correctly to previously unknown attacks. In any case, a determined assailant will often be able to quickly identify an LIH system. Both configuration and logging are relatively simple, and an attacker's options are normally limited to reduce operator risk. Nevertheless, LIH are useful resources for early warning and information gathering purposes.

Examples of LIH are Specter (commercial) and Honeyd (open source).



Sample honey deployment

Both software-only solutions allow a single LIH host to emulate multiple common network services of one or more operating systems and log all detected activity.

Honeyd can generate entire virtual networks of fake hosts, gateways and routers on a single physical machine, complete with thousands of virtual IP stacks that will pass operating system fingerprinting tests such as an nmap scan. A single honeyd server can easily scatter virtual honeypots throughout the unused IP address space of a production network, including support for extensive routing topologies, and can even introduce network latency and apply QoS rules. This makes honeyd a flexible and fairly powerful host simulation, monitoring and alerting tool.

Understanding HIH

Unlike functionally-limited LIH, high interaction honeypots (HIH) provide entire operating systems running real applications for attackers to interact with. Services are no longer emulated. Instead, real computers are deployed for attackers to remotely compromise.

HIH provide great advantages, in that attackers can not only probe the honeypot but can actually break in and run command shells and applications or download new exploits and toolkits. Because all services are real, HIH can detect novel activities. They can gather extensive detail about an attacker's actions, including keystrokes and interactive sessions, which increase the opportunity to analyse potential threats.

Other HIH benefits include optimization at deployment time for subsequent forensic analysis. For example, they can use pre-computed MD5 file hashes to support the easy detection of subsequent file system modification. They generally have shorter, more focused deployment life spans, with the associated reduction in unwanted system activity that can potentially slow down incident analysis.

The downside to HIH is that higher levels of interaction increase the level of risk. Attackers have more potential to use the honeypot for malicious purposes. Constraining attackers and controlling outbound data are the greatest challenges for honeynet operators.

HIH are also more complex. Real computer systems must be deployed and managed, and must be supported by covert monitoring systems required to observe the attacker's actions. Because malicious users will potentially be accessing the honeypot systems, before deploying HIH within your organisation, you should also investigate any relevant local legal issues and address concerns about liability for an attacker's actions.

Production vs research

Honeypots are often further classified into production or research honeypots, depending upon their intended objectives and deployment architecture. Production honeypots are used to detect, respond to and possibly prevent attacks, by mirroring production systems or distracting attackers from higher value targets. Techniques used include slowing down network scanners and DoS attacks using 'sticky' techniques such as LaBrea Tarpits, or confusing attacker with large numbers of low value targets and greatly increasing the amount of noisy scanning required.

Honeypots can easily be taken offline and analysed in situations when real production systems could not, and they also provide excellent preparation and training in incident handling (such as establishing internal incident response plans, chain of custody processes and localised forensics best practices). Honeypots can also be rolled out relatively rapidly to provide additional live analysis and detection capabilities in areas facing raised security threat levels.

Research honeypots are generally deployed in much wider ranges of configurations and are primarily intended to capture and analyse malicious activity, gather information, detect trends and better understand blackhat tools, techniques and motivations.

How honeynets evolved

The term 'honeynet' was coined by the Honeynet Project when trying to describe whole networks of HIH. Honeynets are the most powerful and complex types of HIH, providing entire networks of real computer systems for attackers to fully interact with.

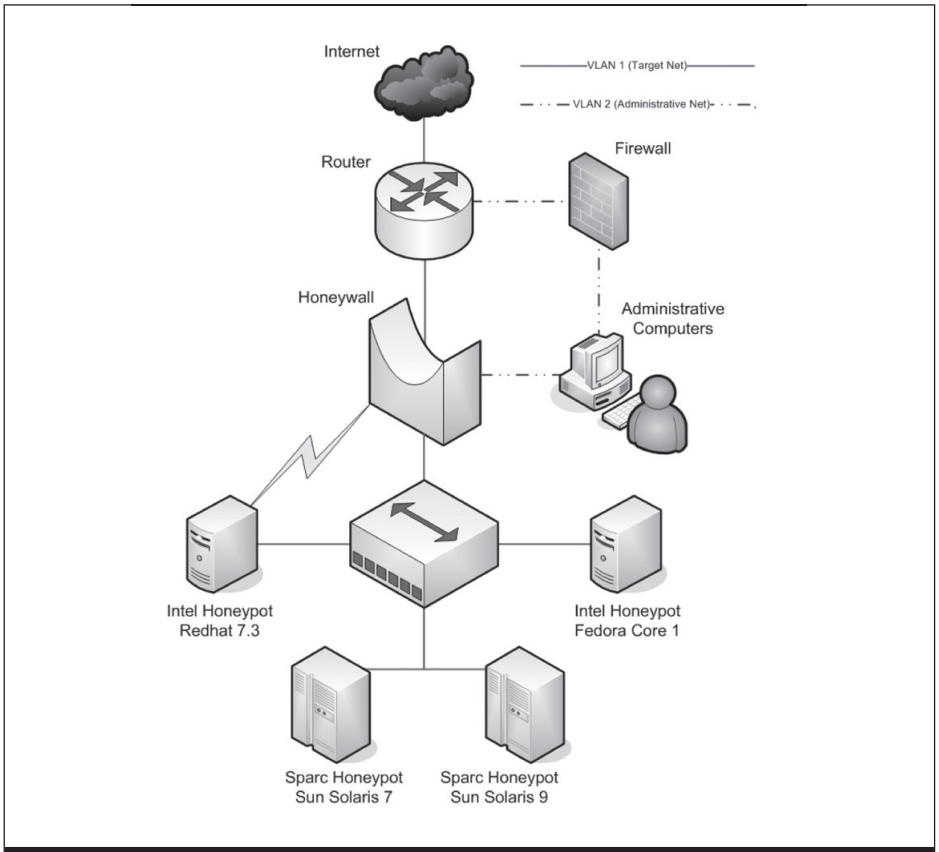
A honeynet covertly observes every action of an attacker in complete detail. It's a computer version of the reality TV programme *Big Brother* house, where the environment appears real but in fact only exists for the purpose of observing the behaviour of its participants. Unlike most reality TV shows, however, the contestants are hopefully unaware of their audience and definitely won't be winning any popularity prizes!

First-generation honeynets were crude affairs comprising separate physical hardware devices running multiple operating systems, and were often difficult to configure and support remotely. They operated at layer 3 of the OSI model and were non-transparent to IP traffic. They

were potentially intrusive and often easy to detect by observing traceroute output and increased time-to live counts. These honeynets were able to capture plain text attacker communications, but were easily defeated by encryption or non-standard network protocols.

Second-generation honeynets provided greater integration with purpose-built tools for key tasks. They were often consolidated onto single devices to reduce hardware requirements, support and management costs. Honeynet network architecture evolved to become stealthier and more intelligent, adding transparency and multiple layers to make detection more difficult.

Solutions to the challenges of encryption and outbound attack mitigation were also developed. The Honeynet Project released Eeyore, its second-generation honeywall gateway, in 2003. Eeyore was a bootable LiveCD running a cut down busybox Linux kernel. This eased deployment pains and increased the number of people deploying honeynets, but it was still quite inflexible and hard to customise or maintain over long periods.



Sample Honeynet Topology

Current honeynet technology

The 2005 release of Roo, the Honeynet Project's next generation honeywall gateway, delivered a full local hard drive installation of the Fedora Core 3 operating system, and added easy updating, online documentation and a number of additional data analysis features. The Roo honeywall is designed for transparent deployment in front of all honeypots, with honeynet and management traffic transmitted over separate physical network interfaces.

For data control, Roo implements layer 2 bridging. This transparently forwards Ethernet frames between an external and internal network interface and allows significant activities to be performed on packets during the transition process. Features include an iptables-based firewall to manage all IP network connections, packet rate limitation capabilities to throttle outbound honeypot network activity, and support for both whitelists and blacklists.

Roo also features snort_inline, an intrusion prevention system developed by the Honeynet Project and designed to inspect outbound network traffic for signs of malicious signatures then covertly block or mangle the suspected traffic. These advances in data control substantially improve the protection now available to honeynet operators.

These features offer multiple layers of data capture. This data is supplemented by full network packet capture, passive operating system detection (p0f) and network flow analysis and aggregation (using Argus and the Honeynet Project's own HFlow application). All honeypot-related network activity is processed and then logged in a MySQL database, which is accessible via the secure web based "Walleye" analysis interface. Real time email or pager alerting plus daily network summary report capabilities are also provided.

Sebek

The Honeynet Project's Sebek tool addresses the challenge of host-based monitoring and capturing encrypted data. It employs a loadable kernel module approach (more often seen in blackhat rootkits) to handle host data capture in the kernel rather than the user space, making it much harder to detect and defeat. Trojaned read() calls capture all user keystrokes and file/socket read activity, which is then written to non-local storage via UDP network packets. This traffic is hidden from system users by bypassing the host's own IP stack and instead writing the data directly to the network card device driver, effectively rendering it invisible to an attacker, even if root access and a network packet sniffer are available. The Sebek client is available for Linux, *BSD, Solaris and Windows operating systems.

Honeynets in the real world

Production honeynets are regularly deployed to provide early warning and attack detection capabilities, or to alert organizations to potential insider threats. Law enforcement, governments and ISPs are actively involved in many forms of honeynet-related activity, such as critical infrastructure protection, international

botnet tracking and establishing global early warning systems.

Honeypots help to defend Internet users against spam, phishing and other forms of cybercrime activity. They harvest new exploits and malicious code for the signature generation programmes of major antivirus vendors, and detect new network activity for inclusion in intrusion detection and prevention system updates. Client-side honeypots continuously process queues of suspect emails and instant messages, or crawl the web looking for malicious content, so that popular email and web clients can then block such material through the latest security updates and watch lists.

Research honeynets are used to gather information about current computer security threats, help detect worm outbreaks and assist in tracking global information security trends. Because the data collected by honeynets are observed blackhat actions and not just the isolated analysis of malicious code, honeynets help to educate security researchers and academics, corporate system administrators and infrastructure protection groups, law enforcement agents, military intelligence and even home computer users. Improving the community's knowledge of blackhat tools, tactics and motivations is an important part of improving our overall defences, and a continually evolving process.

Current areas of research by the Honeynet Project include global distributed honeynet deployments (for early warning and long term trend analysis), improvements in honeynet data analysis (through the establishment of a unified security data analysis framework and visual programming environment), operating high-value honeypots (initially within the financial services industry), detection of insider threats, anti-spam honeypots, real time dynamic analysis of newly detected malware, and detection and mitigation of botnet command and control channels. Semi-automated attack profiling and integration of post compromise forensic databases remains a longer term goal.

Types of honeynet research

Honeynet Project members have previously used honeynets to successfully:

- Analyse attacks against common operating systems and determine average 'time to compromise' models for standard operating system configurations.
- Reverse engineer new zero-day malware, worms and mass scanning tools / autorooters.
- Publish one of the first documented cases of underground credit card trading and organised cybercrime, and repeatedly track online cybercriminals involved in financial scams such as carding.
- Capture and categorise phishing and pharming techniques, along with observing various spam, open proxy operations, identify theft and DDoS extortion rackets.
- Observe attackers setting up their public websites, forums and IRC servers on compromised honeypots, or embarking on political hacktivism campaigns (such as in India, Pakistan and Indonesia).
- Produce one of the first profiling models for blackhat social interaction, based on captured IRC data and real world observation.
- Detect unusual network activity, such as IPv6 traffic tunnelled over IPv4 (a group of Italian hackers breaking into NASA via Solaris servers in Mexico), hacking sessions being tunnelled over Network Voice Protocol and malicious peer-to-peer activity on production networks.
- Analyse malware and botnet propagation mechanisms and then build botnet command and control detection and mitigation solutions.
- Track the market for botnet rental services and other underground pseudo-currencies, such as credit card details and account credentials.

The security arms race

Over the past decade, the main forms of observed malicious activity have regularly changed. PC antivirus software greatly reduced infection rates from physical media, but widespread network communications opened up other avenues of attack to the blackhats. Early manual intrusions on poorly configured systems were countered through establishing better systems management techniques and adding simple access control lists. Mass scanning of network services and the impact of network worms was mitigated through more advanced firewall technologies and new intrusion detection systems, but the growth in broadband internet access exposed many more PC systems to the blackhat threat.

Worms once again became commonplace, but recent security improvements - especially to the MS Windows desktop - reduced the effectiveness of network based attacks and forced attackers to move up the application stack. Client and online applications are now usually their targets, and increasingly social engineering and targeted attacks that use short-lived, custom crafted trojans are becoming the blackhat norm, with financial gains almost always the main goal.

Although honeynet technology continues to evolve, so do blackhat countermeasures. Honeypots running default configurations become less attractive once mass scanning activity is reduced, and more effort is now required to 'sweeten' research honeynets and attract advanced or targeted threats. Additional work is also required to disguise honeypots and prevent their true purpose being exposed, and larger scale honeynet deployments are required to study global rather than local trends.

Honeynet technology is obviously subject to blackhat scrutiny, as all the Honeynet Project's tools and research is given away to the public under open source licenses, so blackhat papers on subjects such as detecting and disabling Sebek, fingerprinting a honeypot, or detecting the presence of honeywall network traffic rate limits by measuring packet throughput regularly require corresponding improvements in honeynet technology and deployment practices.

Virtual honeynets

Virtualization technologies such as VMWare, User Mode Linux (UML) and Xen are attractive as they allow an entire honeynet to be consolidated onto a single machine, reducing infrastructure costs and operational support complexity. Multiple operating system variants can be run in parallel, using only basic hardware, and complex virtual network topologies can quickly be assembled. Less power and space per honeynet are required, meaning self-contained laptop systems can easily be used as portable virtual honeynets for training, demonstration or in-the-field deployment purposes.

Typically the base operating system runs only the virtualization software and then honeywalls and honeypots are installed into individual virtual machines as guest operating systems. Remote graphical or web-based management is often provided, and virtual machines can be completely reinstalled using remote virtual media. Data backups are significantly simplified and accelerated through use of disk snapshot technology, whilst automated virtual machine reconfiguration is often possible through the virtualization technology's APIs.

Downsides to virtual honeynets include a single point of failure, greater chance of honeypot fingerprinting due to standard virtual machine hardware configurations, and the additional risk that if a virtual honeypot is compromised, a back channel attack against the underlying virtualization technology could lead to a compromise of the entire virtual honeynet and host operating system.

Some attempts have been made to mitigate the increased risks posed by virtual honeynets, such as binary obfuscation patches for VMWare. The Honeypot Proc File System (HPPFS), and secure, undetectable keystroke monitoring via TTY logging at the underlying driver layer of the guest operating system. A more secure operating mode called separate kernel address space (SKAS) enables the UML kernel to run in a different host address space from its processes, which further reduces the risk of virtualization technology detection. However, blackhat research in this area also remains active, and it is likely that specific attacks against virtual honeynets will be detected in the coming years, continuing the honeynet arms race.

Advanced honeynets

Operating a single honeypot in one geographic location will often provide interesting and useful information, but a wider perspective is required to study the threats faced by larger organisations and to understand trends on a global level. If enough honeynets are deployed over a large enough network range, honeynet operators can potentially answer questions about the Internet as a whole. Distributed honeynets provide one such tool for casting much larger metaphorical nets.

Physically distributed honeynets are placed in different locations, usually set up in a standalone configuration but configured to additionally submit data to a central location. This can be as simple as the daily transfer of all captured network data to a shared directory structure, or as complex as a real-time distributed solution supporting federated and confederated trust models, hierarchical central data stores and multi-analyst push/pull querying of all global data.

The obvious major downside to physically distributed honeynets is that moving a honeynet node requires the hardware to be physically shipped between locations, configured, and managed remotely. One way to solve this problem is by centrally locating all the honeypots and honeynets in one well-managed location and then transparently tunnelling network traffic destined for a remote honeypot's apparent IP address to the central honeyfarm. With only minimal routing capabilities being required on each remote site, technologies such as GRE tunnelling and multiple VLANs can seamlessly make a honeypot physically located in the central honeyfarm on the other side of the world appear to as an attacker as if it was actually local to the targeted remote network. Honeymole by the Portuguese Honeynet Project is an example of a traffic tunnelling tool for building honeyfarms.

An observant attacker could potentially detect a honeyfarmed honeypot by observing incorrectly incremented TTL counts on returned packets. Luckily network address translation (NAT), iptables packet mangling and policy-based routing can be used to remove this behaviour, although increased network latency over large international distances might still alert a more perceptive attacker. Currently there is no obvious solution to the problem of honeyfarms operating over large international distances. In either case, the main problem raised by distributed honeynets is the increase in captured data volumes and associated significant data analysis challenges.

About the author

Having operated honeynet systems for many years, David Watson leads the UK Honeynet Project (www.ukhoneynet.org) and is also one of five Research Alliance steering committee members. He released the Honeysnap reporting tool, was co-author of the Honeynet Project's "KYE Phishing" white paper and is currently researching blackhat IRC activities, POS and financial system honeypots. He is also the project manager and lead developer for the Honeynet Project's Global Distributed Honeynet initiative and has presented in the past to most major US government, military and law enforcement agencies.

References

[1] Stoll, Clifford (1989) *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage* New York: Pocket Books

[2] Cheswick, W. R.. and Bellovin, S. M, *An Evening with Berford In which a Cracker is Lured, Endured, and Studied*, Chapter 10 in Firewalls and Internet Security, Addision Wesley, Reading, MA, 1994.