The Honeynet Project: Data Collection Tools, Infrastructure, Archives and Analysis

David Watson
The UK Honeynet Project Chapter
david@honeynet.org.uk

Jamie Riden
The UK Honeynet Project Chapter
jamie@honeynet.org.uk

Abstract

We briefly introduce the Honeynet Project, describe the honeynet data collection tools and techniques currently in use by it's members, review the types of data collected and research published, and present some current and proposed infrastructures for capturing and sharing honeypot-derived network attack data.

1. The Honeynet Project

The Honeynet Project[1] is an international volunteer organization dedicated to computer security research. It was founded in 1999 and holds non-profit (501c3) status in Illinois, USA. Membership is drawn from active chapters in over 25 countries, which helps to provide a global research perspective, and the organization is strongly committed to the ideals of the Open Source movement.

The goal of the Honeynet Project is to learn about "the tools, tactics and motives involved in computer network attacks", which is primarily carried out through the use of honeypots and honeynets (networks of honeypots). A honeypot has been defined as "a security resource whose value lies in being probed, attacked, or compromised"[2].

A series of honeypot and honeynet related technologies have been developed to help support the Honeynet Project's research goals, all of which are freely available for download from our public website[3]. Regular member-generated research publications on new threats are released in the form of "Know Your Enemy" (KYE) white papers[4], which are also available from our public web site, as well as conference presentations or academic paper publications by chapters or individual members[5].

2. Data Collection Tools

First we will briefly review some of the data collection tools that are commonly in use by the Honeynet Project. Some of these tools have been developed wholly or partly by the Honeynet Project and others are merely being used in honeynet deployments.

Honeyd[6] is a low-interaction honeypot which emulates the IP stacks of various target Operating Systems and optionally provides basic service emulation, using an efficient design allows for at least 65,000 simulated hosts on a commodity PC. It has support for routed network topologies, including simulated routers and associated latency and bandwidth issues[Chandran03], as well as being able to place real machines within the IP address range honeyd is using.

Nepenthes[7] provides a low-interaction honeypot which emulates common Windows known vulnerabilities and downloads the payloads when an attacker attempts to exploit these issues. Nepenthes is extremely good at capturing autonomously spreading malware such as worms. Some of the data generated when running a nepenthes node is the attack source, the downloaded malware and optionally the hexdumps of shellcode used to leverage the exploit. Nepenthes also provides an optional sample submission to external agents, so for example captured data can be fed directly to an automated sandbox or battery of anti-virus engines for fully automated sample analysis and reporting.

Honeytrap[8] is another low-interaction network services honeypot. It too provides basic service emulation, but differs from Honeyd and Nepenthes by also detecting connection attempts against unbound TCP ports. It uses connection monitors to extract TCP connection attempts from a network stream and alert on unknown attacks, and can also mirror or proxy TCP connections.

Kojoney[9] is a low-interaction honeypot which emulates an SSH server process and records the usernames and passwords of attackers attempting to log in to the honeypot. Since the machine is solely a honeypot it is assumed that zero or a negligible

proportion of the attempts are being made in good faith. Attacker source addresses are logged and later geolocated to provide basic attack reporting.

Sebek[10] is a tool for monitoring high-interaction honeypots; it is basically a rootkit-style kernel module or patch and supports Linux, BSD, Solaris and Win32 platforms. It hooks system read/write calls to capture attacker's keystrokes, file access and other input/output activity. This data is then exported over the network via UDP packets, so another function of Sebek is to hide this monitoring traffic from the attacker as well as hiding its own presence on the machine. Host level data capture capabilities are particularly useful when encryption technologies would otherwise result in attacker activity going undetected by plain text network based IDS devices or packet capture based solutions.

Hflow[11] is a data coalescing tool for honeynet/network analysis. It coalesces data from snort, p0f, and sebekd into a unified cross related data structure for storage in a relational database. The recently released Hflow2[12] offers further improvements over traditional netflow based approaches for high interaction honeynet research and attempts to address some of the potential performance issues with sebek and hflow.

Honeywall[13] is a bootable CDROM which can be used to quickly build a high-interaction honeynet. Honeywall acts as a transparent layer two network bridge and allows for transparent data control, data capture plus online data analysis. Some of the data collected by Honeywall includes iptables logs, keystroke logs (for Sebek enabled high interaction honeypots), file and I/O data, passive OS fingerprints via p0f, snort alerts, argus/netflow/hflow data and full binary packet dumps. Another function of a Honeywall is to mitigate outbound attacks launched from a compromised honeypot onto the public Internet; snort_inline and connection rate limiting are used for this purpose. A classical generation III honeynet is formed from various honeypot hosts located behind a Honeywall bridge. Secure remote management and online data analysis are provided by the WallEye web based console.

Capture-HPC[14] is a high-interaction client honeypot framework . In the recently released version 2.0, support is provided for running arbitrary applications within a Windows virtual machine, although the major use so far has been crawling potentially malicious URLs with Internet Explorer and looking for those sites which attempt to change the state of the virtual machine in undesirable ways. The Capture-HPC client program tracks state changes within the virtual machine using kernel call-back mechanisms and reports these to the server portion of Capture-HPC. In addition to the notification of changed files, the server portion can also collect a packet dump in tcpdump format and the actual contents of files which were changed, modified or deleted after the particular URL was visited. A list of URLs can be obtained via parsing email from spam-traps, manual submission or by searching for various terms on a search engine such as Google or Yahoo.

Honeybow[15] fulfills a similar role to Capture-HPC but uses slightly a different methodology to achieve the same ends; it can also integrate with the Nepenthes/mwcollect GOTEK collection architecture.

Various web application honeypots have been developed by Honeynet Project members, including PHPHoP[16], Google Hack Honeypot[17] and HIHAT[18]. The former two are low-interaction honeypots designed to emulate vulnerable web applications and to capture payloads and monitor basic attacker search engine and mass scanning behavior. Conversely HIHAT provides a high-interaction web application honeypot and data reporting interface. It allows easy conversion of many existing PHP applications to a honeypot and has been successful in monitoring web application attacks such as SQL injection and remote file includes.

SpamPot[19] is a system for collecting and analyzing spam email messages, and is one of a number of similar systems.

PEHunter[20] is a snort dynamic preprocessor that extracts Windows executables from network traffic. It is typically deployed as an inline device in front of high-interaction honeypots.

Tracker[21] facilitates the identification of abnormal DNS activity. It will find domains that are resolving to a large number of IP's in a short period of time then continue to track those hostname->IP mappings until either the hostname no longer responds or the user decides to stop tracking that hostname. Tracker has recently been used for finding fast-flux domains and unusual A-Record rotations.

Honeymole[22] is a tool designed to simplify the deployment of multiple honeypots by tunneling network traffic to a central network of honeypots (a honeyfarm).

Honeystick[23] is another tool for simplifying honeynet deployments. It is a bootable USB based virtual honeynet that includes both a Honeywall and honeypots on a single portable device. Similar

CDROM and DVD based systems have also been developed in the past, and an updated bootable virtual honeynet and Nepenthes sensors will be made available later this year.

3. Data Collection

Different honeypot implementations can give rise to a large amount of data of different types, from packet captures and malicious binaries, to keystroke logs of attacker's interactions with machines and URLs of malicious web sites. By definition, honeypot systems only exist to be probed or attacked, so wherever possible, all potential sources of incident data are recorded.

At a basic level, most high-interaction honeynet tools provide tcpdump data, keystroke logs, file access details and other input/output associated with an attacker's visit to a honeypot. Many of the low-interaction honeypot tools are designed to capture binaries directly, and in the case of nepenthes, also the particular shellcode used in the exploit. Client honeypot solutions such as Capture-HPC additionally provide the contents of all files which have changed during a drive-by exploitation of the client virtual machine, together with a list of files and registry keys that have changed.

From these multiple sources of raw honeynet data we can derive additional information such as extracting downloaded executables, capturing botnet command and control messages or generating textual logs of IRC conversations. Due to the volumes of data involved per incident and high levels of automated network attacks, we generally make use of automated analysis tools to lay the groundwork for a human analyst to efficiently respond to each incident. Without such automation, “needlestack” data overload is the main challenge facing most honeynet analysts today.

4. Analysis Tools

Honeysnap[24] is a tool for processing tcpdump data. It runs in an offline batch mode on a tcpdump format file, or a series of such pcap files, and it produces a textual summary of each pcap file, including a breakdown into different protocols. TCP streams are automatically reassembled and files or messages from common network protocols such as HTTP, FTP and SMTP are automatically identified and extracted. Of particular use in typical incidents is the reconstruction of IRC conversations on arbitrary ports, which can often play a major part in analysis of compromised honeypots. Apache format logs of inbound and outbound web requests are also automatically generated. Another valuable use of honeysnap is to post-process the packet capture data produced by Capture-HPC; this allows us to see not only the files that have been changed on the client honeypot but also to easily view the web pages that caused the compromise of the client machine. Recently a prototype web interface to honeysnap has been under development, which adds experimental support for cross honeynet analysis.

CWSandbox[25] is an automated sandbox for the Win32 family of Operating Systems. It uses hooking of API calls to trace any file and process access, network communication or registry key changes caused by the execution of submitted binary samples. The system includes packet capture for all outbound network traffic and returns a report of the first three minutes of program execution in XML format.

Anubis[26] is another automated Win32 sandbox solution from the Vienna University of Technology and is similar in capabilities to CWSandbox.

VirusTotal[27] offers online malicious binary sample submission to multiple anti-virus engines. Samples are processed using the latest AV signatures from 20+ vendors and an email report is returned, allowing known malware samples to be easily identified and novel sample to be highlighted.

Capture BAT[28] is a behavioral analysis tool of applications for the Win32 operating system family. It monitors state changes on a low kernel level during the execution of applications and processing of documents, providing an analyst with insights on how the software operates even if no source code is available. Known event noise can be excluded by a fine-grained mechanism that allows an analyst to take into account the process that cause the various state changes. As a result, this mechanism even allows Capture to analyze the behavior of documents that execute within the context of an application, for example the behavior of a malicious Microsoft Word document.

Honeynet Project members generally employ the above tools plus more traditional incident response software and network analysis tools to reconstruct the sequence of events during an attack or compromise of a high-interaction honeypot. Typically a human analyst will examine the enhanced data sets, such as reporting from Honeysnap, Honeywall, CWSandbox and Capture-HPC, and they will then describe the incident; this may involve examining full packet captures, querying public DNS records or correlating data with

other honeypots and sources. In this case the analyst is performing part of the usual incident response procedure, but with the help of the more extensive host and network level data being logged by the honeynet.

5. Data Collection Infrastructures

Due to the volunteer based, geographically dispersed nature of the Honeynet Project's membership, many different types of honeynet systems may be in operation at any particular time. This potentially provides Honeynet Project members with access to a rich and widely varied range of international honeynet-derived data sets, but it also raises concerns over the implications for data protection, privacy and the potential risks of widespread data sharing.

Data collection infrastructures can generally be classified into a number of types:

- Ad hoc (local to one individual member or chapter) or centrally co-ordinated
- Low interaction, high interaction or client honeypots
- Physical or virtual honeypots
- Short term or long lived deployments
- Honeypots shut down immediately once compromised, or allowed to continue operations
- Geographically local or remotely tunneled (honeyfarm) deployments
- Collected data for private, limited use or available within shared data sets

Further details of most Honeynet Project data collection infrastructures can be found within the annual reports issued by each member chapter[29], but examples of various types of Honeynet Project deployments include:

5.1 Local Ad Hoc Deployments

Typically operated by an individual or local Chapter, these isolated deployments tend to involve both physical and virtual honeypots (both low and high interaction), and usually employ a subset of the Honeynet Project's technologies. Although sometimes short lived, some Chapters have operated such deployments over extended periods and may have gathered multiple years of full Honeywall data. Full data sets are usually available only to local members and may contain tens of gigabytes of high interaction pcap data, although higher level sanitized summary data and specific pieces of detailed data may also be shared between individual members or Chapters. Examples would include continuous multi-year physical honeypot deployments by the UK[30] and Georgia Tech[31] Honeynet Project Chapters, or malicious web site crawling records from the New Zealand[32] Honeynet Project Chapter.

5.2 Regionally Co-ordinated Deployments

A number of Honeynet Project Chapters or members operate regional research initiatives that focus on security threats within specific geographic locations. Deployments tend to be a set standardized honeypots that are deployed on multiple geographically separate sites, and project durations tend to be multi-year. Data analysis is usually shared with other regional groups, such as CERTs, and although aggregate data is often published, raw data usually remains private.

Examples would include the Brazilian Distributed Honeypot project[33] (using low interaction honeypots running on OpenBSD, which began in 2003, now has nodes in more than 40 Brazilian organizations and publishes regular sanitized summary data to partners), the Brazilian SpamPots project[34] (which emulated open proxy/relays on 10 separate broadband networks and captured almost half a billion spam emails addressed to over four billion recipients by over 200,000 sending hosts during 400 days from 2006-2007) and the Chinese Honeynet Project Chapter's Matrix Distributed Honeynet project[35] (which deployed 40+ honeypots over 17 nodes distributed at 16 provinces for CNCERT/CC).

5.3 Participation in Third Party Research Projects

Honeynet Project members regularly host sensor nodes, contribute malware samples and contribute analysis to a number of third party research projects that have similar goals and objectives. This occurs on both on a large scale, organized manner and also through personal contacts and private channels, and the data sets involved range from full public disclosure, through shared risk/reward peer groups to closed door vetted infosec communities. Examples include:

- Submissions to various IP black list and RBLs
- Phishing and spam submissions to groups such as Castle Cops and SpamHaus
- Industry group contribution such as SANS handlers
- Contribution to bleeding edge snort signatures
- Hosting sensor nodes for honeynet based research projects such as Leurre.com
- Malware sample collection and submission to

groups such as MWCollect and Malfease
- Operating public malware analysis services such as www.cwsandbox.org
- Malware sample submission to AV vendors
- Botnet C&C tracking and mitigation with groups such as Shadowserver

5.4 Centrally Co-ordinated Deployments - Global Distributed Honeynet Phase One

The Honeynet Project's Global Distributed Honeynet (GDH) Phase One[30] was an attempt to standardize, automate and simplify deployment of complete honeynets. GDH Phase One ran from January to June 2007 and used the generation III architecture including a Honeywall and Sebek on one or more high interaction honeypots plus a Nepenthes Sensor, but implemented each honeypot as a virtual rather than physical machine.

GDH Phase One featured eleven nodes around the world, comprising four public IP addresses each, and during the three months between March to May it collected 122 GBytes of high interaction pcap data. 73 million argus flows were logged, composed of 730 million packets from 301,200 unique source IP addresses. 672,800 SSH log in attempts were recorded using Kojoney honeypots and 1680 unique malware samples were collected by Nepenthes sensors (and automatically analyzed by submitting them to CWSandbox and VirusTotal). Human analysts then viewed the events and produced high-level write-ups of each incident, which included over 300 “handlers diary” blog posts and a 250+ page internal status report.

Interesting incidents included observation of simultaneous compromises of geographically separate vulnerable web applications by the same web application botnet attackers, identification of attack traffic detected by the majority of global sensor nodes and measurement of the effectiveness of AV software against localized malware. Raw data was only shared between node hosting Honeynet Project members, although summary details were released to the public in November 2007.

The first phase of the GDH project provided a test bed for next-generation distributed technology and data analysis tools, processes and research for the Honeynet Project. The primary lessons learned were that data volumes quickly became the main challenge and that greater automation was required to improve our data analysis capabilities. A number of improvements will be required in GenIII honeynet tools, such as Sebek, Hflow and the Honeywall. Significant time was spent on making the initial installation work on different pieces of hardware, even with a fairly good minimum specification. Another issue was that the original license agreement to host a GDH node only let participants share with each other, and that to disseminate data more widely, permission had to be sought from each participant, which substantially reduced the overall benefit to the full Honeynet Project membership.

Further details on GDH Phase One can be found in David Watson's November 2007 PacSec presentation, which is available on the Honeynet Project's website[36]. Additional information about the project can be presented to interested parties as required.

5.5 Centrally Co-ordinated Deployments - Global Distributed Honeynet Phase Two

GDH Phase Two will begin in April 2008 and will build on the lessons learned in Phase One. The aim is to maximize deployment efficiency through hardware standardization and continuously operate a global network of both low and high interaction distributed honeynets, based on current honeynet technology. It will include a larger range of honeypots, including a client honeypot component in the form of Capture-HPC, and feature more regular rotation of virtual honeypot images. It will allow rapid deployment and rotation of new honeypots by uploading new virtual machines. GDH phase two will aim to consolidate, integrate and improve our existing distributed data analysis capabilities and add a content-rich data query interface to assist with analysis.

This time the license agreement to host a GDH node will allow the data produced to be made available to all Honeynet Project members (and selected external partner organizations). This will allow the Honeynet Project to maintain a larger incident response team, and hopefully to publish more interesting and timely research. We also aim to substantially increase our sensor deployment installation footprint by evaluating both bootable and embedded Linux nodes for light-weight Nepenthes sensors or OpenVPN gateways to a central honeyfarm. We also plan to work with groups such as Shadowserver to “outsource” some elements of our data processing to organizations with existing significant resources in areas that would otherwise require further internal development.

6. Conclusion

The Honeynet Project has developed a wide

range of honeypot-based data collection technologies and its members regularly deploy these technologies in various data collection infrastructures. Significant volumes of high value attack data is regularly collected, including a number of ongoing multi-year data collection initiatives, and this research data is used to private the public with regular information on the latest security threats in the form of KYE white papers and individual Chapter or member publications. The Honeynet Project appreciates the benefit of increased data sharing and seeks to do more in the future, if concerns over data protection and privacy can be successfully managed and the risk of data leakage is minimized.

7. References

[1] The Honeynet Project: "*About the Honeynet Project*", http://www.honeynet.org/misc/project.html

[2] Spitzner, L. Honeypots: Tracking Hackers, Addison-Wesley, Boston, 2002

[3] The Honeynet Project: "*Tools*", http://www.honeynet.org/tools

[4] The Honeynet Project: "*KYE Whitepapers*", http://www.honeynet.org/papers/kye.html

[5] The Honeynet Project: "*Individual Whitepapers*", http://www.honeynet.org/papers/individual

[6] The Honeynet Project: "*Honeyd*", http://www.honeynet.org/tools/honeyd

[Chandran03] http://www.paladion.net/papers/simulating_networks_with_honeyd.pdf

[7] Baecher, P., Koetter, M., Holz, T., Dornseif, M., Freiling, F.C.: *The Nepenthes platform: An efficient approach to collect malware*. In: Zamboni, D., Kruegel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 165–184. Springer, Heidelberg (2006) http://honeyblog.org/junkyard/paper/collecting-malware-final.pdf

[8] The Giraffe Honeynet Project: "*Honeytrap*" http://honeytrap.mwcollect.org/

[9] Coret, JA: "*Kojoney - A honeypot for the SSH Service*" http://kojoney.sourceforge.net/

[10] The Honeynet Project: "*Sebek*", http://www.honeynet.org/tools/sebek

[11] The Honeynet Project: "*Hflow*", https://projects.honeynet.org/hflow

[12] Viecco, C: "*Improving Honeynet Data Analysis*". In Proceedings of the 2002 IEEE Workshop on Information Assurance and Security, T1B2 1555 United States Military Academy, West Point, NY, 17–19 June 2002 http://www.cs.indiana.edu/~cviecco/papers/hflow2.pdf

[13] The Honeynet Project: "*Honeywall*", https://projects.honeynet.org/honeywall

[14] Seifert, C., Steenson, R. "*Capture-HPC"*, https://projects.honeynet.org/capture-hpc/

[15] The Chinese Honeynet Project: "*Honeybow*", http://honeybow.mwcollect.org/wiki/HoneyBow

[16] The French Honeynet Project: "*PHP HoP*", http://www.rstack.org/phphop/

[17] Chicago Honeynet Project: *"The Google Hack Honeypot*", http://ghh.sourceforge.net/

[18] The German Honeynet Project: "*Web based Honeypot Decoys*", http://honeyblog.org/archives/111-Web-based-Honeypot-Decoys.html

[19] The Brazilian Honeynet Project / CERT.Br: "*Using Honeypots to Monitor Spam and Attack Trends*", http://www.cert.br/docs/palestras/certbr-itu-ap2007.pdf

[20] The Giraffe Honeynet Project: "*PEHunter*" http://honeytrap.mwcollect.org/pehunter.html

[21] The Australian Honeynet Project: "*Fast Flux Tracker*", http://honeynet.org.au/?q=node/10

[22] The Portuguese Honeynet Project: "*Honeymole*", http://www.honeynet-pt.org/index.php/HoneyMole

[23] The UK Honeynet Project: "*Honeystick*", http://www.ukhoneynet.org/research/honeystick-howto/

[24] The UK Honeynet Project: "Honeysnap" https://projects.honeynet.org/honeysnap

[25] Laboratory for Dependable Distributed Systems , University of Mannheim: "*CWSandbox"* http://www.cwsandbox.org

[26] Secure Systems Lab of the Vienna University of Technology: "*Anubis: Analyzing Unknown Binaries*", http://analysis.seclab.tuwien.ac.at/features.php

[27] Hispasec: "*VirusTotal*", http://www.virustotal.com

[28] Seifert, C. : "*Capture BAT",* http://newzealand.honeynet.org/capture-standalone.html

[29] The Honeynet Project: "*Annual Reports*", http://www.honeynet.org/status/sr-200710.html

[30] The UK Honeynet Project: "*Annual Reports*", http://www.ukhoneynet.org/status-reports/

[31] The GA Tech Honeynet Project" "*Annual Report",* http://www.nsa.gatech.edu/honeynet/reports/2007-03.html

[32] The Honeynet Project: "Malicious *Websites*", http://www.honeynet.org/papers/mws/index.html

[33]The Brazilian Honeynet Project: "*Distributed Honeypots*

Alliance", http://www.honeypots-alliance.org.br/
[34] The Brazilian Honeynet Project: "*SpamPots*", http://www.cert.br/docs/whitepapers/spampots/
[35] The Chinese Honeynet Project: "*Status Report*", http://www.icst.pku.edu.cn/honeynetweb/honeyneten/statusreport_200703.htm
[36] Watson, D. : "*Global Distributed Honeynet",* PacSec07, http://www.honeynet.org/speaking/PacSec07_David_Watson_Global_Distributed_Honeynet.pdf