Honey Plotter and the Web of Terror

Mark Withall*, M. Shirantha de Silva*, David Parish*, and Iain Phillips†
*Electronic & Electrical Engineering, Loughborough University, UK
Email: {m.s.withall,m.s.de-silva,d.j.parish}@lboro.ac.uk
†Department of Computer Science, Loughborough University, UK
Email: i.w.phillips@lboro.ac.uk

Abstract— **Honeypots are a useful tool for discovering the distribution of malicious traffic on the Internet and how that traffic evolves over time. In addition, they allow an insight into new attacks appearing. One major problem is analysing the large amounts of data generated by such honeypots and correlating between multiple honeypots. Honey Plotter is a web-based query and visualisation tool to allow investigation into data gathered by a distributed honeypot network. It is built on top of a relational database, which allows great flexibility in the questions that can be asked and has automatic generation of visualisations based on the results of queries. The main focus is on aggregate statistics but individual attacks can also be analysed. Statistical comparison of distributions is also provided to assist with detecting anomalies in the data; helping separate out common malicious traffic from new threats and trends. Two short case studies are presented to give an example of the types of analysis that can be performed.**

I. Introduction

This paper introduces a tool for navigating and visualising data from a distributed honeypot network. The tool allows flexible examination of both aggregate traffic statistics and detailed characteristics, cross-correlation between multiple honeypots and the detection of anomalies via the statistical comparison of traffic distributions.

One of the main problems implementing effective network traffic analysis tools is the large volumes of raw data involved. Most honeypots deal with this by only storing a small amount of ‘per flow’ information. However, in this project, all the packets are recorded to allow more flexibility in examining the malicious activities. The problem is addressed through the use of a relational database and a web-based user interface to interact with it.

A. Motivation

Many of the technologies on which the Internet has been built were designed before the globalisation of the Internet was envisaged. Vulnerabilities in software processes and security systems are frequently discovered and give opportunities for hackers to gain access to remote systems and information.

There is a perception that the motivation behind security attacks is altering [1]. At one time perpetrators were often teenagers, gaining access to systems to demonstrate prowess. Now however, organised criminal groups are exploiting security loop holes to gain access to sensitive information such as credit card numbers. It is estimated that up to 8 million credit card numbers were stolen in one attack alone in 2003 [2]. Networks of compromised hosts are being used to launch Distributed Denial of Service (DDoS) attacks in an attempt to extort money from online businesses [3].

B. Interactive Network Sensors

Network security is maintained using a variety of tools and techniques. Software processes and services are patched regularly, to remove vulnerabilities. Firewalls are configured to restrict access to hosts, only allowing traffic to pass through that matches specified criteria [4]. Network Intrusion Detection Systems (NIDS) are deployed to report on known malicious traffic [5]. These techniques rely heavily on prior knowledge of existing exploits. Although there are systems that incorporate anomaly detection, most systems rely predominantly on pattern matching. Consequently new exploits are still very hard to secure against, especially where they are used to take advantage of previously unidentified vulnerabilities.

Interactive Network Sensors (often referred to as honeypots or honeynets) are another tool in the network security armoury [6]. In essence they are monitoring systems that are used to observe attackers activities. The hosts run standard services but are unadvertised, meaning that there is no legitimate reason for communicating with them. However, illegitimate traffic from, for instance, a hacker scanning for potential targets or a worm exploiting the address space randomly, are detected. Through the analysis of attackers activities, new exploits can be revealed and the motivation behind attacks can be understood. Interactive Network Sensors are a means of reconnaissance, of keeping in touch with the current activities of the hi-tech criminals.

Interactive Network Sensors may be deployed in a number of ways. There are high interaction and low interaction sensors. High Interaction Sensors are designed to allow hackers the ability to interact with the host just as they would any other target. Hosts may be compromised and used at the hacker’s discretion, for reconnaissance, communication or even as a zombie for a further attack. The time span for these activities could be anything from seconds to months. The activity is monitored giving useful information about the tools the hackers use and sometimes revealing the motives behind the attacks. Low Interaction Sensors emulate the responses of a designated server. They are designed to allow some interaction but greatly restrict the scope of the hacker. Specifically, a hacker would not be able to install software nor launch further attacks. Low Interaction Sensors are not truly compromised; they emulate responses.

1-4244-1251-X/07/$25.00 2007 IEEE.

C. Low Interaction Sensors

The tools presented in this paper are aimed primarily at Low Interaction honeypot data.

The honeypots used in this project aim mainly to generate bulk statistics (rather than investigate individual attacks) to give an overall profile of a site and to correlate between multiple sites. It is expected that these aggregated statistics will show seasonal variations, according to the time of day, day of week *etc.*

Low-interaction honeypots are suitable for this task, as they are simple to deploy and don't need complete realism in their emulation. These sensors run emulated services that can interact, to a point, with a user. This enables them to emulate more services and Internet Protocol (IP) addresses per site as they are less resource intensive. This enables large numbers of low Interaction Sensors to be deployed, giving a reasonable sample of activity across the IP address space. The honeypots used in this project capture and store all packets whereas most honeypots only log a small number of statistics per flow (similar to NIDS).

II. Methods

In this section the configuration of the honeypots used is discussed, along with the design of the query and visualisation tool used to analyse the data produced.

A. Honeypots

Each honeypot system comprises a single machine running open source Honeypot software (Honeyd[1]) at a low-interaction level. A set of monitoring, management and security scripts are incorporated. The entire software application works using a 'Live' CD that runs OpenBSD 3.7[2]. The sensor design is focused around a secure and maintainable Honeypot, which provides full packet capture and storage of Internet traffic directed to it. A daily log, consisting of 24 hours worth of Internet traffic, is captured in a raw binary format (pcap); starting from midnight of each day. Whilst the complete operating system and scripts reside in memory (a feature of the 'Live' CD) the actual captured traffic is stored in the form of logs on the hard disk. These logs are transferred to central computers for post-processing and analysis. The Honeypot contains two network interfaces. One interface captures Honeypot traffic from emulated machines; each with a unique IP address. The other interface provides a management interface to a single access computer. This management interface is secured by the native OpenBSD firewall "Packet Filter". The Honeypot software (Honeyd) runs scripts that emulates (to a limited extent) several common services such as web servers (IIS and APACHE), FTP servers, SSH servers and TELNET. It also emulates the operating systems in various UNIX and Microsoft flavours. In addition to the emulated services provided by honeyd, the software implements the correct ARP response mechanism for emulated IP addresses (as and when required) via custom scripts.

1) Sites: There are currently four active sites located in various locations around the UK. These honeypots capture data on 8, 12, 28 and 1024 IP address. The number and types of services emulated in each honeypot vary with the number of addresses on the honeypot.

B. Honey Plotter

To assist in the navigation of the data produced by the honeypots, a web-based query and visualisation tool was developed; called Honey Plotter. This tool is in two halves: a database for storing the packet headers and meta data, and a web-based user interface for querying the database and producing visualisations.

1) Data Storage: By storing the information about the traffic from the honeypots in a database, it allows the information to be easily queried in a flexible manner. The database of choice, for this application, was PostgreSQL[3]. PostgreSQL is a free, high-performance database, which conveniently supports network data types, such as IP and MAC addresses, natively. The main issue is which data should be stored in the database.

The obvious choice is to store the packet headers in the database, as they already conform to a structure that can be easily used in this context and they allow many useful questions to be asked about the data. For example, using the packet header information we can construct the distribution of Transmission Control Protocol (TCP) ports, for a given time window, which gives a good indication of the types of malicious traffic on the network. As the whole packets are already stored, in the pcap files, we can refer back to them for any information not in the database, such as the packet content.

The particular header information stored is the standard fields from the following layer 2, 3 and 4 protocols:

- Ethernet
- IP
- TCP
- UDP
- ICMP

In addition to the protocol headers listed above, various meta-data is stored. Information about each pcap file is stored; the filename, the honeypot location, the first and last packet timestamps and the total number of packets and bytes. This allows some more general queries to be performed, such as the number of bytes per month per site, without having to query the individual packet information. Also, each packet has the file it came from, the timestamp and its length stored.

Finally, several auxiliary tables are stored in the database to allow port numbers, protocols and the location of IP addresses to be converted to a more useful, human-readable form.

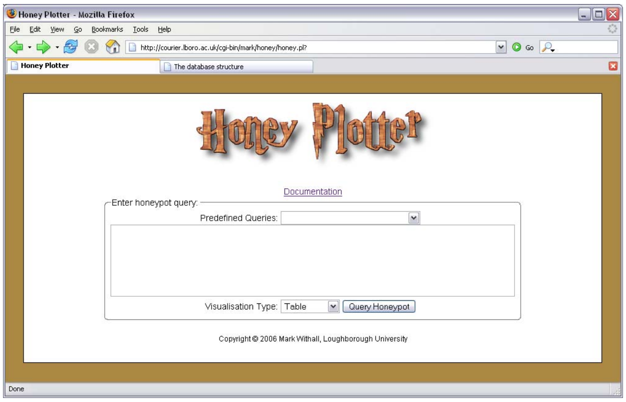
2) User Interface: In addition to the database, a web-based user interface was created to provide an easy way to construct visualisations from queries of the data.

The tools used to develop this interface were Perl/CGI, AJAX (Google Maps^{TM}API) and gnuplot. These tools were

[1] http://www.honeyd.org/

[2] http://www.openbsd.org/37.html

[3] http://www.postgresql.org/



http://courier.lboro.ac.uk/cgi-bin/mark/honey/honey.pl?

Enter honeypot query:

Predefined Queries:

Visualisation Type: Table

Copyright © 2006 Mark Withall, Loughborough University

Fig. 1. Main query interface for Honey Plotter

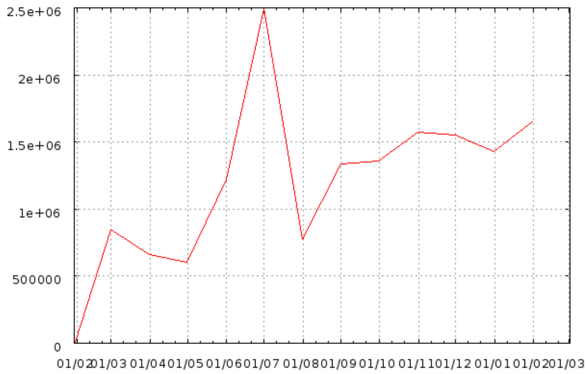


Fig. 2. Change over time of number of packets sent to the honeypot at site 1

chosen because they allowed rapid development of the system (due both to their power for this type of application and the familiarity the developers have with them).

The main query interface is shown in Figure 1.

For this initial version of Honey Plotter, a fairly simple (low-level) interface was decided upon. This consists mainly of a textbox for the user to input an Structured Query Language (SQL) query. The main work behind-the-scenes was to map the results of this query to the selected visualisation type. There are four types of visualisation:

Table This is just a nicely formated version of the table returned by the database for the given query.

Line graph This takes the first two columns of the returned table and uses the first column as the x-axis and the second as the y-axis. There is an option in the interface to specify that the x-axis should be treated as a timestamp.

Histogram This works in much the same way as the line graph, only this time the first column is used as the labels for each column in the histogram.

Map This requires the returned table to have four columns labelled: count, countryname, latitude and longitude. These are then used via the Google Maps^{TM}API, to produce an interactive map of the data. Each location is given a coloured flag (based on the value of count), which, when clicked on, will show the country name and count value.

Examples of the different visualisation can be seen in the case studies presented in the Results section.

3) Statistical Analysis: A challenge facing the community is that of automated data analysis. As honeypots have no legitimate use, all traffic observed to and from a sensor must, by definition, either be misdirected or illegitimate. However, of the large volume of illegitimate traffic on today's Internet, much is of little direct interest from the perspective of learning about new and emerging attacks. Port scans for instance are constantly being attempted across the IP address space, but there is little to gain through the detailed analysis of such data; although knowledge of new ports being scanning may be useful. The packets are usually generated automatically by, for example, compromised hosts and yield little new information. The value of analysing this background of automated tools, searching for poorly secured systems to infect, is at a sharp contrast to the value of analysing the activities of human beings, perhaps using a new exploit, determinedly targeting a specific host. Consequently, tools are needed that are able to distinguish between the melee of illegitimate but uninteresting background noise, and the illegitimate and genuinely interesting new exploits or targeted attacks.

In addition to the manual interaction, via the web-based interface, the Honey Plotter tools include statistical analysis scripts can be run on the underlying database to find anomalies in the data based on 'significant' changes in distribution. For example, a script was developed to construct Kolmogorov-Smirnov (KS) statistic values for TCP port distribution and time-to-live, with various time windows, for the the comparative distributions. This information was then plotted using gnuplot.

III. Results

Table I shows a summary of the data gathered from all of the honeypots, up until the end of February 2007. The table gives the total number of bytes captured by the honeypot and the average number of bytes per IP address per month.

Honey Plotter is used to conduct the following two short case studies. For the case studies, only Site 1 is analysed. Figure 2 shows how the traffic rate has changed over the lifetime of the honeypot.

A. Case Study 1: Geographic source change over time

This case study looks at the changes in geographic distribution of malicious traffic over time.

Figure 3 shows the geographic distribution from the first whole month (March 2006)[4]. The countries sending the most malicious traffic are Romania and Ireland. There is also quite

[4] See the PDF version of this paper for colour images

TABLE I

DATA CAPTURE SUMMARY FOR ALL SITES. SITE 2 EMULATES 1024 IP ADDRESSES BUT ONLY 12 ARE STORED IN THIS VERSION OF THE DATABASE. SITE 4 HAD NOT COLLECTED ITS FIRST WHOLE MONTH OF DATA AT THE TIME OF WRITING.

Site	IPs	Months collecting	Total Traffic (Bytes)	Average Traffic per IP per Month
1	12	13	1,473,087,125	9,442,866
2	1024 (12)	5	53,127,805	885,463
3	28	4	1,141,985,521	10,196,299
4	8	0	0	0

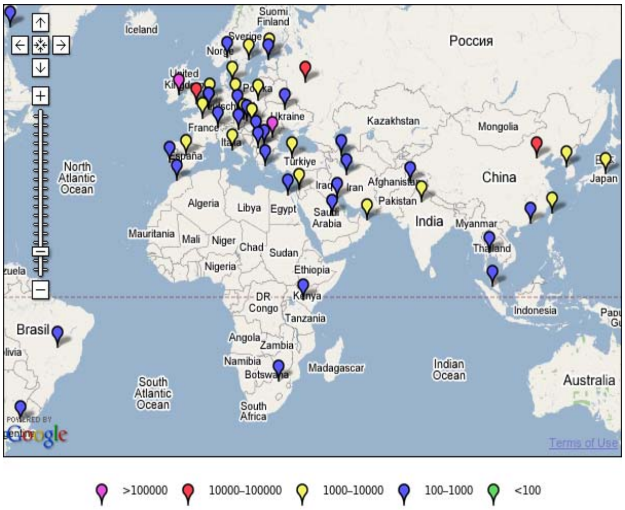


Fig. 3. Geographic distribution of malicious traffic (number of packets sent) from March 2006

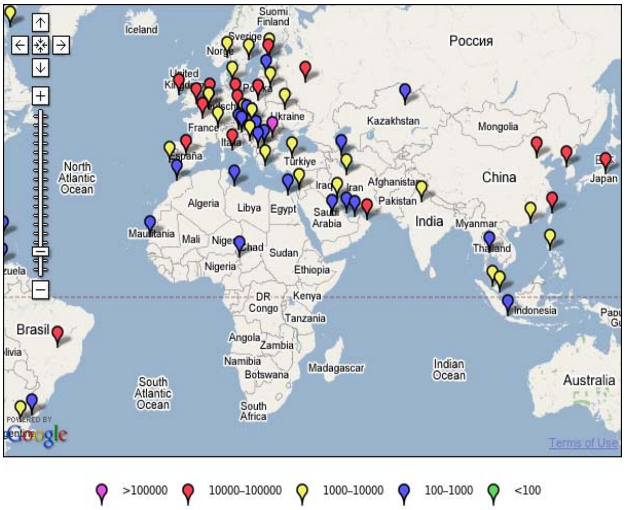


Fig. 4. Geographic distribution of malicious traffic (number of packets sent) from December 2006

a lot of traffic from Russia, China and the United States (not shown in the figure).

As time goes on, there is an increase in malicious traffic from Europe and Southeast Asia. Figure 4 shows the geographic distribution from December 2006 with many more European and Southeast Asian countries sending large numbers of packets.

Another interesting thing to note is the large number of different countries that are sending traffic.

B. Case Study 2: TCP port distribution over time

The second case study focuses on the TCP port distribution over time.

The TCP port distribution is fairly consistent over the 13 month lifetime of the honeypot. The most common TCP ports, in most months, are 1433, 139 and 445. The official use of these ports are Microsoft SQL Server (1433), NetBIOS (139) and Active Directory (445). However, as this is a honeypot, no user is likely to be attempting to uses these services legitimately. Each of these ports also corresponds to common worm attacks. SQL Server has a known buffer overflow vulnerability on port 1433 that is attacked by, for example, the SQL Snake worm. Port 445 is used by worms like Sasser and Korgo, exploiting the LSASS vulnerability and port 139 is used as an alternative port for Sasser (suggesting this is the most likely cause of the traffic on port 445, as both ports have heavy traffic).

Figure 5 shows a typical month, with the above ports being the most commonly used.

At one point, port 1433 becomes very dominant (which correlated to an increase in traffic from Russia); beginning in May 2006 and eclipsing most other traffic in June and July, before returning to more normal levels. Figure 6 shows the TCP port distribution for July 2006, with virtually all traffic being on port 1433.

The next two most common ports are 5900 and 135. 135 is fairly consistently in the top 4 or 5 ports over the whole data. It is officially the port for the Microsoft RPC Locator Service but is most likely the Nachi or MSBlast worms.

Port 5900 is used by the VNC remote desktop protocol but in around May 2006 an exploit was released[5], which corresponds to its first appearance in the top 10. Over the next few months the port usage became much more common. See Figure 5 for its position in November 2006.

One final interesting event was a sudden spike in the usage of port 80, in August 2006.

[5] http://www.securityfocus.com/bid/17978

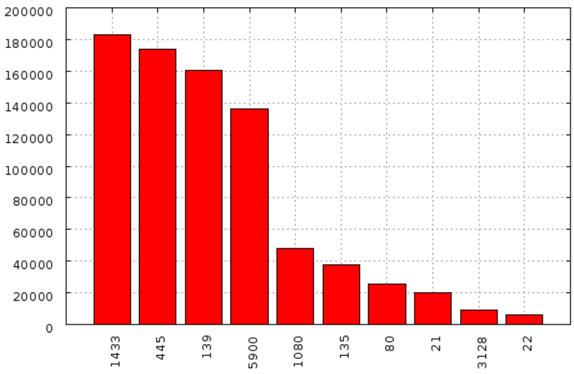


Fig. 5. The TCP port distribution for November 2006.

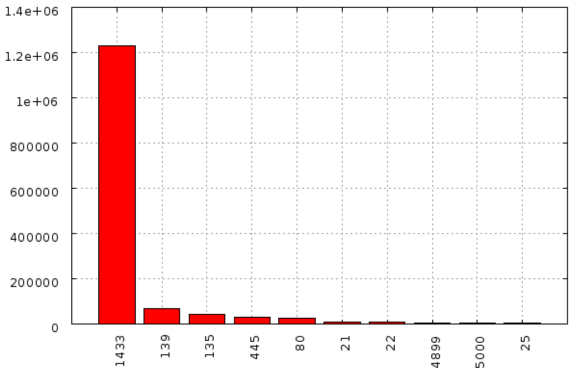


Fig. 6. The TCP port distribution for July 2006.

IV. Conclusions

A web-based query and visualisation tool for analysing data from low-interaction honeypots has been presented. The tool allows a flexible approach to analysing malicious traffic gathered by multiple honeypots along with accompanying scripts to assist in the detection of anomalies in the data. As the underlying database is created from pcap files, it can be generalised to any network monitoring data. Two small case studies were also presented to show the types of information that can be gained from using Honey Plotter.

Future work on Honey Plotter will be focused in two main areas. The first area is optimising the database structure to minimise the query time for the most common types of queries, by precalculating summaries of the data at various resolutions; for example, by day, by week and by month. In addition, the inclusion of NIDS output (such as snort[6]) may be a useful addition. The second area is improving the user interface and graphing capabilities. The current version requires queries to be input as SQL but it could be much improved by selecting date ranges from calendars, drop-down menus for field selection, *etc*. In addition, the graphs currently only support one line or set of bars. By improving this, the system will have much greater flexibility. Also, the addition of new types of visualisations, based on previous research [7], will be considered.

Acknowledgements

The authors would like to thank John Whitley and Scott Culcheth for their assistance with the titling of the paper.

References

[1] "Know your enemy: Motives." Whitepaper available from `http://www.honeynet.org/papers/motives/`, June 2000.

[2] "Hacker hits up to 8m credit cards." Story from CNN `http://money.cnn.com/2003/02/18/technology/creditcards/`, February 2003.

[3] "E-commerce targeted by blackmailers." Story from BBC News `http://news.bbc.co.uk/go/pr/fr/-/1/hi/technology/3238230.stm`, November 2003.

[4] K. Ingham and S. Forrest, "A history and survey of network firewalls," Tech. Rep. TR-CS-2002-37, University of New Mexico, 2002.

[5] J. McHugh, A. Christie, and J. Allen, "Defending yourself: The role of intrusion detection systems," *IEEE Software*, vol. 17, pp. 42–51, September/October 2000.

[6] J. G. Levine, J. B. Grizzard, and H. L. Owen, "Using honeynets to protect large enterprise networks," *IEEE Security & Privacy*, vol. 2, pp. 73–75, November/December 2004. GA Tech Honeynet Project.

[7] M. S. Withall, I. W. Phillips, and D. J. Parish, "Network visualisation," *IET Communications*, 2007. To appear.

[6] http://www.snort.org/

1-4244-1251-X/07/$25.00 2007 IEEE.