Internet Sensor Grid: Experiences with Passive and Active Instruments

Peter Komisarczuk, Ian Welch

Internet Sensor Grid: Experiences with Passive and Active Instruments

Peter Komisarczuk[1+2], Ian Welch[2]

[1] School of Computing and Technology, Thames Valley University, Ealing, London, UK
{peter.komisarczuk} @tvu.ac.uk

[2] School of Engineering and Computer Science, Victoria University of Wellington,
PO Box 600, Wellington 6140 New Zealand
{ian.welch, peter.komisarczuk} @ecs.vuw.ac.nz

Abstract. The Internet is constantly evolving with new emergent behaviours arising; some of them malicious. This paper discusses opportunities and research direction in an Internet sensor grid for malicious behaviour detection, analysis and countermeasures. We use two example sensors as a basis; firstly the honeyclient for malicious server and content identification (i.e. drive-by-downloads, the most prevalent attack vector for client systems) and secondly the network telescope for Internet Background Radiation detection (IBR - which is classified as unsolicited, non-productive traffic that traverses the Internet, often malicious in nature or origin). Large amounts of security data can be collected from such sensors for analysis and federating honeyclient and telescope data provides a worldwide picture of attacks that could enable the provision of countermeasures. In this paper we outline some experiences with these sensors and analyzing network telescope data through Grid computing as part of an "intelligence layer" within the Internet.

Keywords: Internet Background Radiation, Drive-by-downloads, honeyclient, Network Telescope, Grid computing.

1 Introduction and Background

The Internet is constantly evolving with new emergent behaviours arising [1, 2], some of these new behaviours are benign but unfortunately many are malicious. For example complex distributed entities such as software robot armies (botnets) that make it hazardous to use or be connected to the Internet [3, 4]. The US NSF project GENI (Global Environment for Network Innovation) identifies security and robustness as the most compelling reasons to redesign the Internet [5]. Understanding how to design the network of the future to be more resilient to attack requires an understanding of the current malicious activity in the Internet. In this paper we provide an experience report and review around several tools developed to detect, study and analyse drive-by-downloads for client side attacks and a network telescope for Internet Background Radiation (IBR) detection.

Web-based exploits are currently the fastest growing attack on the Internet with around 1 in 150 web servers said to be compromised in early 2010 [from Kaspersky 2010]. Compromised or malicious web servers deliver "drive by downloads" [6,7] which are exploits that occur when your web browser visits a compromised server; your browser receives the requested web page but also receives targeted and usually obfuscated malicious content often causing a system to be compromised. For example, a keylogger program installed without the users permission in order to gather user name and password data. To discover these exploits instruments called honeyclients [8, 9, 10, 11, 12] have been developed of which there are two main classifications. A high-interaction honeyclient [8] is an instrument consisting of a complete operating system running a web browser where we classify a web server as either malicious or benign based on changes observed to the state of the operating system after visiting the web site. This is a relatively slow process but is highly accurate. On the other hand, low-interaction honeyclients [9] are faster but less accurate because they classify a web page for example by matching structures in the document against previously generated signatures of malicious content. They are prone to give false positive results and would miss new attacks for which there are no signatures.

Internet Background Radiation (IBR) is defined as unsolicited, non-productive traffic that traverses the Internet and is often malicious in nature or origin. An IBR sensor consists of a darknet (or network telescope) [13], which is an IP address range that is advertised but contains no real responders/systems. Instead a darknet passively records all packets. Additionally active responder systems can be deployed which respond to IBR to gain further information on the attacks.

Blocking IBR traffic may be possible if an Internet user, business or ISP can use control data derived from such Internet sensors to identify IBR traffic destined for their network and enable countermeasures. Unfortunately one technique used by attackers to avoid detection is to forge the IP source address of their attack packets. The effect of this is that a victim detects packets from a number of different innocent source addresses. The responses the victims make to these packets are sent back to the forged IP addresses and this traffic is commonly known as "backscatter". There are also benign sources of unsolicited traffic to an address space; such sources include the effects of network mis-configuration and network mapping studies [14]. Pang, el al. [15] showed that TCP traffic accounts for the majority of IBR traffic, ranging from 56.5% to 95.0% of traffic across individual monitors, with TCP ports 135 and 445 (loc-srv and Microsoft-ds respectively) being popular exploits. Likewise Yegneswaran, Barford, and Ullrich [16] showed that worm activity contributed between 20% to 60% of any one day's IBR traffic arriving at a network sensor.

Drive-by Downloads and IBR are just two parts of the overall malicious behaviour seen on the Internet. IBR forms part of the early phase of attacks where the Internet is probed for vulnerable hosts, or it shows the spread of malware through the Internet. IBR detection is based on passive measurement techniques, whereas Drive-by-Download detection is an active measurement that provides: maps of compromised web servers, redirection and exploit servers, detects malware delivery mechanisms, malware packers, malicious payloads and can lead onto detection of botnet command & control infrastructure and attack commands based on honeypot techniques. These, and other instruments can form part of a barometer – a weather map for the Internet.

In this paper we discuss the Internet Sensor Grid. Review some of the developments in active and passive sensors and measurements and discuss some of the Grid based analysis that can be undertaken on network telescope data.

2. The Internet Sensor Grid

We identify an Internet sensor grid [17] comprising active and passive measurement systems that encompasses a wide range of malicious activity measurements, allowing potential correlation between different attack components to be determined. Political, business, social, economic and technical difficulties, make a wide scale federated Internet sensor grid a difficult system to build and run. It requires careful dissemination of information; based on trust and reputation, with guarantees around the provenance of the data, the analysis undertaken and the control data or feedback disseminated.

Some service providers in this space have evolved in recent years, often as not for profit organisations, such as the Honeynet Project [10], ShadowServer [18] and Team Cymru [19] that have capability for large-scale data collection and analysis. Trust is provided through recommendation and vetting of individuals or organizations. These service providers are creating large-scale cooperative sensor and notification services. For example Team Cymru launched the Dragon Research Group in 2009 to deploy a large set of network telescopes to gather data and a community to develop tools [19]. Team Cymru disseminate a bogon routing reference, IP-ASN mapping, malware hash registry as well as Darknet monitoring data. The Honeynet Project are deploying the second phase of their Global Distributed Honeynet (GDH-II) which will incorporate thousands of active honeypot sensors across the world [10] allowing research and dissemination into host attack trends, techniques, malware and botnet command and control.

Networks of the future could develop more sophisticated systems, allowing federated security sensor data to be used to reduce malicious activity. Such systems may use Complex Adaptive Systems (CAS) concepts [1, 2, 20, 21] because of the combination of large numbers of software, hardware and human agents involved. The Internet Sensor Grid would be a CAS in itself as they would be collaborative systems of many components that detect, share, initiate protection or launch countermeasures [22]. Such a system may be based on biologically inspired immune system concepts [23], virally spreading the key information to enable reaction and countermeasures [24] like an immune system using antibodies [25]. Existing work on collective intelligence [26] may also be incorporated but needs to be evaluated to fit with this problem space. Key areas to be developed cover the creation of collaborative sensors and actuators [25, 17, 28, 29], detection and classification of emergent behaviour [3,4], inventing and testing system response and counter-measures. Current basic systems are proving successful, e.g. filtering and warning services currently allow ISPs or Telcos to effectively filter bogon space. Data from darknets, honeypots and honeyclients can be used to provide countermeasures, e.g. by CERTs, to warn typically innocent Internet users that their system is being compromised and used for illegal activities, such as spamming and distributed denial of service attacks.

3. Review of Developments in Active And Passive Sensors

There are a wide variety of sensors that can be employed in the detection of malicious behaviour, including active and passive measurements techniques – integrating network telescopes and active devices such as honeypots, honeyclients, DNS tools for fast flux detection etc. Data is gathered and analysed ranging from low-level protocol interactions through to geopolitical, network topology and network/service provider level. This paper specifically looks at the evolution and applications of the data and intelligence that can be extracted from network telescopes and honeyclients, and describes our experience and development of tools based on Grid computing used to scale systems or analyse the data collected. Below, we briefly review honeypot/honeynet, network telescopes and honeyclient developments.

A honeypot/honeynet is a computer system deployed as a "sacrificial lamb": "A security resource whose value lies in being probed, attacked or compromised" [30]. It has no production value; so anything going to/from the honeypot is likely to be a probe, attack or a compromise. It is used for monitoring, detecting and analyzing attacks and consists of the sacrificial system and a firewall to capture the packets sent and received. A honeypot is labour/skill intensive and has a limited field of view (scalability) and does not directly protect vulnerable end systems. Detection mechanisms tend to fall into one of two broad categories: Active Responders and Passive Monitors [16]. An Active Responder replies to incoming IBR traffic to solicit further traffic so as to more precisely detect its nature and intent. Examples include Honeypots [30] which are characterized as high interaction or low interaction host systems e.g. HoneyTrap [31], HoneyD [32], and simpler higher capacity "Stateless Active Responders" such as iSink [33] and the Internet Motion Sensor [34]. These systems are attached to the Internet in order to be probed and compromised and their value is in the intelligence gathered. A honeyclient on the other hand scans the Internet and is looking to be compromised by malicious servers in order to detect compromised or malicious servers and their exploit mechanisms.

A high interaction honeypot [30] is secured with a reasonable password on the common system userid's and is monitored by software and network hardware. The system is typically accessed through a Honeywall – a firewall that allows attackers in but limits the connectivity from the honeypot so that it cannot effectively be used to launch attacks, but does allow the attacker to incorporate the honeypot into a botnet for example. The return on investment or value obtained from such honeypots includes knowledge of botnet command and control infrastructure, attack plans, malicious actions etc.

Low interaction honeypot systems provide an emulated computer system, services and applications. These systems have limited capabilities, which make them more scalable and can emulate a large network address space and provide services that to some extent looks like the real thing. Such devices are relatively easy to deploy, with lower risk because they are not complete computer systems, however they capture more limited information. A number of low interaction systems exist, for example the HoneyTrap [31], often run in a virtual machine, accepts TCP connections to gathers information including at least the first data packet. It is designed to keep TCP connections open long enough to receive useful information on the exploit and it also provides a 'mirror mode' option where the attack can be reflected to the originator.

HoneyD [32] is a dynamic tool that fakes a number of IP addresses and emulates the fingerprints of various operating systems, making it harder for an attacker to detect it is talking to a honeypot. It can be deployed within an Enterprise using any un-used address space, so an attacker scanning for devices to compromise could pick an address used by HoneyD rather than a real system. HoneyD runs services on specific ports using scripts, which can show known security flaws in order to coax the attacker to launch an attack. Default operations can be applied, such as sending TCP reset, or accept commands for a given port. Data from honeypots is being aggregated and shared through the mwcollect alliance, see http://alliance.mwcollect.org.

The honeyclient has several forms like the honeypot. The high interaction honeyclient is a complete system driven to browse the web and uses system state changes to detect malicious behaviour e.g. that captures API calls, shown in Figure 1. The honeyclient system developed incorporates a server component (Capture-HPC) [8] and a Microsoft Windows behavioural analysis tool (Capture-BAT), running in a Virtual Machine environment controlled from the coordinating server. The latest developments have been to provide network API monitoring and extensions to the capture server to incorporate a database and checkpoints to optimize operations [35] and has been used in a variety of studies, including a long term scan of the .nz domain.

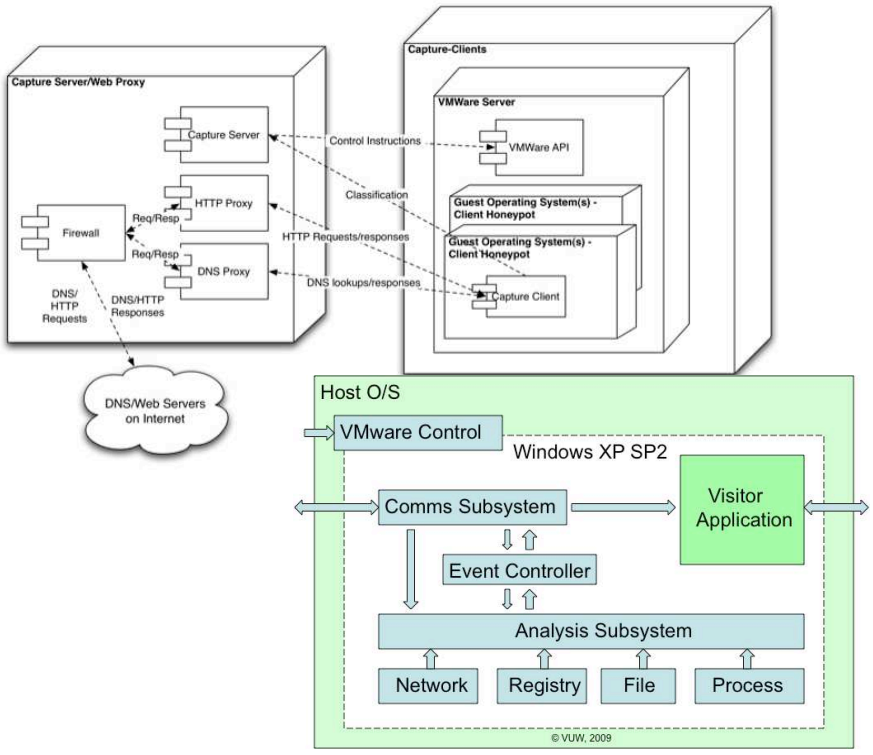

Figure 1 Capture-HPC honeyclient architecture [8, 22]

Developments to scale the honeyclient system have been trialed using Grid computing which encapsulated the system for the Grid using the gRAVI toolkit [36] and using workflow engines to control Grid execution [28]. The use of workflow engines proved less effective than hoped at scaling honeyclient infrastructure. Alternatively low interaction honeyclients [9] can be employed, typically using fewer compute resources to classify a web page as potentially malicious. Static analysis of web page features can be used to provide a classification, e.g. the use of obfuscated JavaScript or small iframes [37] and it can use DNS data to form server maps [27] and detect fast-flux attacks to classify interactions that are likely to be malicious. Other low interaction systems, such as PhoneyC [38], provide a virtual HTTP honeyclient combining a web crawler and analysis engine. The input is provided through a call to Curl, which mimics a legitimate browser's behavior. The page is then evaluated, all content is downloaded, scripts are evaluated and a classification of the webpage made. The malicious web pages from the low interaction system are then checked using a high interaction honeyclient. Combining low and high interaction honeyclients into hybrid systems can optimise system performance and thus costs and helps develop the honeyclient business case [39]. These systems employ feedback loops for learning and automated signature generation [40].

Unlike low interaction honeypots or honeyclients, a darknet or network telescope is a passive monitoring system that records IBR traffic destined for a range of addresses. The darknet does not respond to any incoming packets, thus there are no scalability issues with respect to responding to incoming packets, nor is there a need to provide real or emulated services. The bottleneck is the speed at which we can record packets for analysis. Network telescopes require significantly less computing and bandwidth resources than active responder systems as no reply packets are generated [13, 15, 19, 41, 42, 43]. The network telescope does detect a wide range of malicious activity, but the data gathered is limited to the first packet in an exchange, so it may not be possible to determine the exact nature of the attack.

The size of the IP address range being monitored by a Network Telescope has a major impact on its ability to detect large-scale network events [13]. As it is purely passive the time to detect an event is based on the probability of an attacker selecting the darknet address range, so the larger a darknet the more likely it is to detect malicious activity, or be picked for scanning, or by a propagating worm. Detection time ranges from an average 1.82 hours for a /16 network address pool down to 19.4 days for a sensor using a /24 network [13]. Harder, Johnson, Bradley, and Knottenbelt [43] outline the difficulty of accurately detecting worm infection rates using a /24 network telescope. Unfortunately, increasing the size of a Network Telescope increases the packet rate that the sensor is expected to deal with. In circumstances where not all traffic can be captured, traffic sampling can be employed, or sensors can be segmented. Studies from Yegneswaran, Barford, and Ullrich [16]; and Pang, Yegneswaran, Barford, Paxson, and Peterson [15] have pioneered the use of various sampling methods for network telescope data. Nonetheless, there is a loss in accuracy because the entire population of IBR is not being captured.

The data captured by these sensors ranges from probes to attacks and to the exploitation of the attacked systems. The data captured includes protocols (packets), addresses of exploited systems, malware samples, command & control interactions, misuse and attacks. The data is stored in a number of formats, from PCAP and binary

files through to logs of keystrokes, system and API calls, including temporal data. This data can be employed for example to identify the attackers and classify attacks [44, 37]. The IEEE Computer Security Group (ICSG) is beginning standardization of data exchange for the security industry using an XML schema for meta-data exchange [45], covering malware sample exchange, URLs, and events such as Conficker.

4. Network Telescope and Analysis Engine

The network telescope setup used at Victoria University is shown in Figure 2 consisting of the advertising router, providing routing advertisements of the address space used, a VLAN trunk to the capture server for data collection, which can be split up using multiple VLANs. This host captures any packets that are destined for this address space, packs them into PCAP files and these packets are sent for analysis periodically, reducing the real time processing overhead of the system. The trade off for the telescopes lower resource overhead is that no packets are ever sent back to the IBR source, which limits the ability to ascertain the source's true intent. With a darknet, for example, all that can be determined is that there is an incoming TCP connection to a specific port e.g. 135, a Blaster worm attack, but without a response to the source we are unable to see the confirming connection on port 4444 [34].

The capture server integrity is maintained by hardening the system, we use tcpdump to capture all packets to disk (in standard PCAP format) and use netfilter rules to identify if any traffic originates from the telescope indicating a breach of the system. The tcpdump process is monitored by the init process to ensure that the tcpdump process is kept alive. All data was captured and stored in 100MB files, which were compressed along with identifying metadata. Data processing was accomplished using a tool developed called pcapstat written in C using the libtrace library [46]. Pcapstat was deployed on a GT2 Grid and the resulting output analysed with the open source statistical package R [47]. Details can be found in the thesis from Dean Pemberton [48].

The network telescope deployed used a /16 address range located in New Zealand and was unused at that time; it has since been re-used by an organization for their IP needs. It was deployed continuously from 16th December 2004 to the 12th March 2006. The dataset collected by the network telescope consists of approximately 225.6 GBytes and the anonymised dataset is available at DATCAT (http://www.datcat.org, the Internet measurement data catalog), where the destination network address has been replaced with the loopback address, but all other data, including all the data contained within UDP or TCP packets has been preserved except for any destination address information. Anonymisation of the data was completed using a modified version of the libtrace anonymisation tool from the WAND group at Waikato University (http://research.wand.net.nz/software/libtrace.php). The PCAP data was anonymised by changing the first two octets of the destination address to 127.0 respectively, but preserving the lower 16 bits of the address. The libtrace library has a tool to anonymise packet headers and adjust the header checksum accordingly, the modified version of the tool also anonymised occurrences of the source address in binary or text form, or in reverse order as used for example in a reverse DNS lookups in the data field of packets.

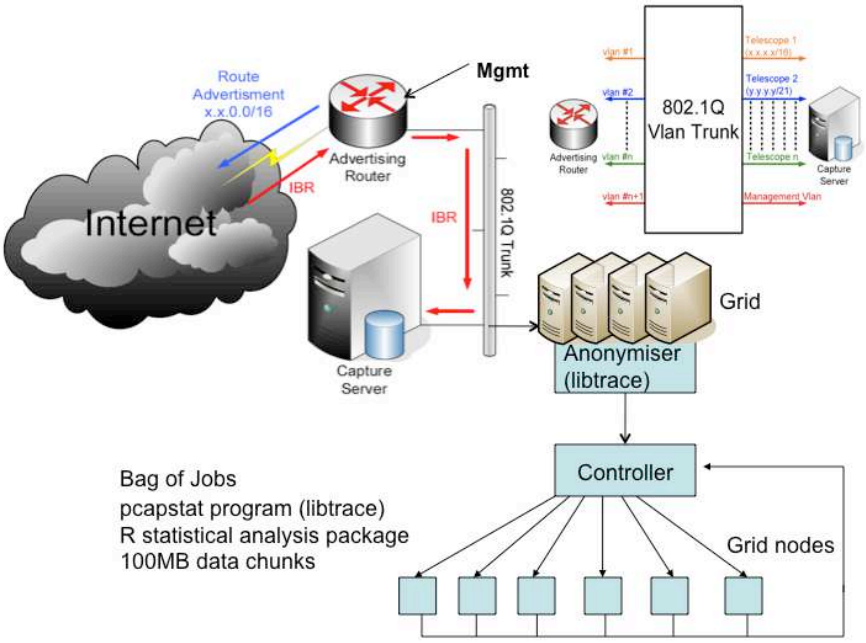

Figure 2 Network Telescope Setup and Grid Analysis Engine

On average approximately 499Mbytes of data was collected per day over the 452 days of the experiment. Aggregating data from even a reasonably small number of /16 network telescopes, such as envisaged in medium to large scale experiments, would require processing of many 10's of GBytes of data per day in order to produce attack data, trend analysis, IP geo-location, black listing etc. Some data sources are required to anonymise data before sharing it and all data needs to be processed to provide base statistics and trend analysis, e.g. using the statistical package R. Our experiment developed some Grid based solutions that distribute data amongst a set of nodes using the "bag of jobs" paradigm, shown in Figure 2. Data is stored in the pcap format and our base analysis software (pacpstat), extracts core data [48]. Data is concatenated into 100MByte files to be processed and pcapstat was statically compiled to overcome issues with deployment on the Grid (availability of libtrace library).

The Grid setup for the analysis used a distributed computing architecture based on the Sun Grid Engine (Sun Microsystems). This Grid cluster has 232 execution nodes made up from desktop computers used by staff and students connected through a fast Ethernet network. Jobs can be submitted to this cluster and are assigned a node on which to run. A single Grid user is limited to submitting up to 90 concurrent jobs. All of the data file archives on which analysis was performed were stored on a shared file system. This network file system is made available to each of the computing nodes via NFS. This setup allowed up to 90 nodes to process data from the archive files, which initially caused a bottleneck at the file server. To overcome this bottleneck the scripts used to run these jobs copied compressed data files from the file server to local hard disc and to minimize the impact of file server congestion and network latency

the extracted data was written to local disc and then the files were archived and transmitted in compressed form back to the NFS share once all analysis was complete. The distribution of Pentium IV processor speeds used were: 42% - 2600MHz, 37% - 2800MHz, 21% - 3200MHz. These computing nodes were required to analyse 2256 individual data files of 100Mbytes each. The distribution of node execution times for pcapstat on the data files is shown in Figure 3. The sum of the execution times is 504,264 seconds (almost 6 days) so parallel processing is essential for rapid analysis. Because the actual analysis runs were performed in a number of chunks to allow other users to make use of the computing grid, it is difficult to obtain a precise overall execution time. For an Internet control loop we may need more responsiveness – 10's of seconds rather than 100's. This raises a number of engineering questions: What is a good sample size from network telescopes? How do we aggregate large data sets? How do we detect changes in probe patterns, and categorise anomalous behaviour (anything captured by a network telescope is potentially malicious except for a few cases)?

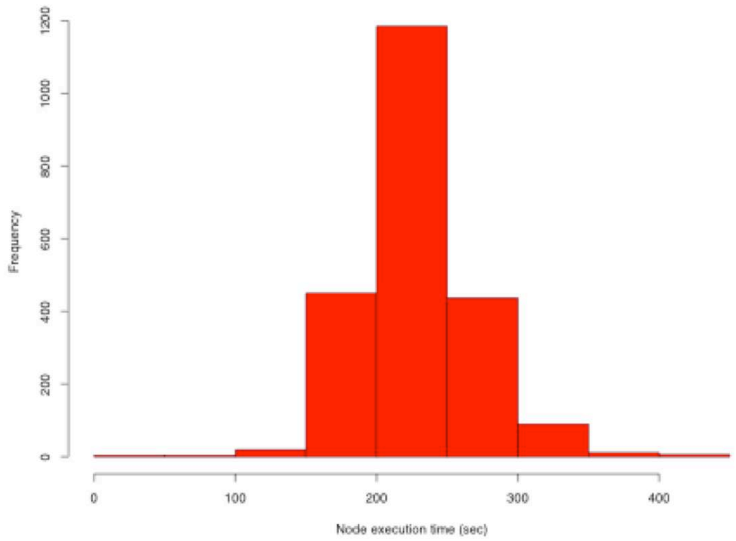

Figure 3 Grid node processing time (seconds) for 100MB PCAP traces

Previously we identified the non uniformity of data hitting the network telescope from an address range perspective and identified the data sampling strategy that would be best employed to estimate the IBR hitting a larger address space [48, 49]. Here we outline other aspects of the dataset such as typical attack trends detected. The network telescope captured a set of TCP (60.7%), UDP (36.6%) and other traffic (2.7%), mainly ICMP, ARP and IGMP. In terms of IBR arrival rate the network telescope saw a wide range of arrival rates, the majority of activity seen related to port scans. Figures 4 and 5 show typical features, including the repetitive nature of scans on a weekly basis (UDP port 1434), the decay in attacks (e.g. TCP port 1023), step changes (e.g. TCP port 139) and ad hoc attacks (e.g. UDP port 135).

Geo-IP location and ASN determination are a useful component of IBR analysis. Typical results for each attack address are shown in Figure 6, using Google maps and the free Geolite database [50], which provides latitude and longitude of an IP address. Some country data is sparse, however it gives an indication of attack location.

Figure 4 Top 4 TCP port activity (packets/day)

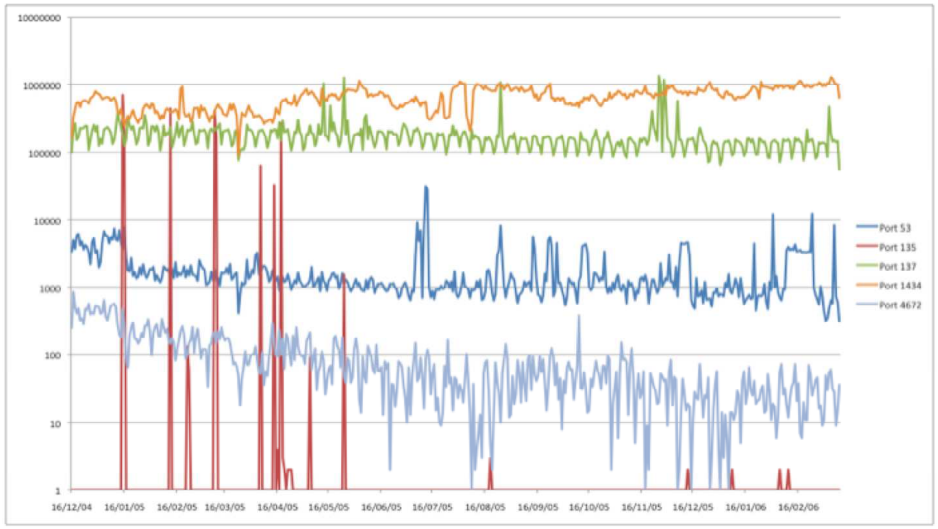

Figure 5 Top UDP port activity (packets/day) and occasional port 135 attacks

The intensity of each attack location is shown using a colour scale. Red indicates a 50% attack intensity (China), green areas indicate a 1% attack intensity, blue areas indicates a low attack intensity (< 0.1%). For geo-IP analysis the dataset task was split into 1800 jobs, which were sent to Grid nodes sequentially using Ruby and the

DRMAA framework (http://drmaa4ruby.sunsource.net/). Jobs downloaded a file containing 10,000 IP addresses and output a file for a second job summarising and collating information for mapping. Each Grid job finished within an hour.

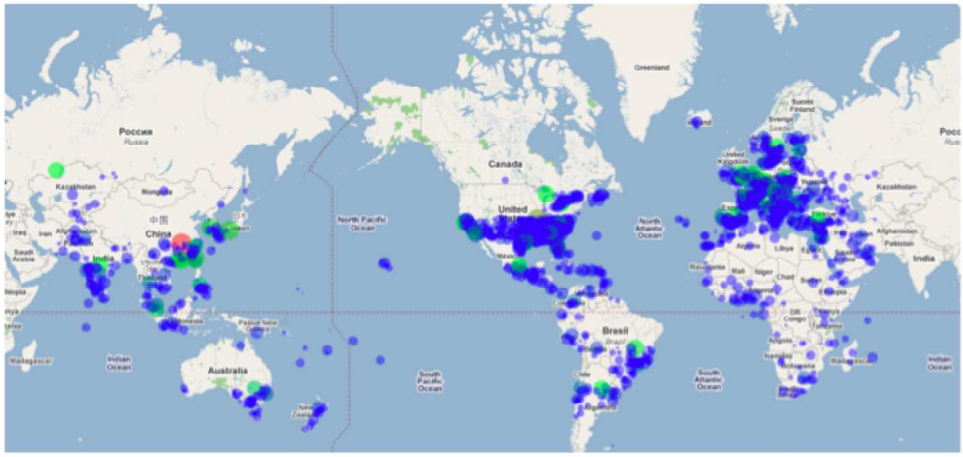

Figure 6. Geo-heatmap, locating attacks using Google Maps and Geolite data

Using IBR data for traffic filtering was tested on a simplistic testbed. An analysis of a single 100MB sample from near the start of the telescope data indicated two prevalent attack types, the SQL slammer worm and Microsoft-ds from a total of 7,287 attacking hosts. Applying exclusion lists matching these source addresses and TCP characteristics over a sampled set of logged data files showed between 94.3% to 21.4% of all IBR traffic hitting the network telescope would be stopped through the filtering specifically for these addresses and exploits. This indicated the long-term activity of the exploited hosts and attackers prevalent throughout the experiment.

5 Summary

This paper has provided a review of work related to malicious activity detection on the Internet using active and passive measurement devices. We highlight work that is extending these measurement devices and discuss some of the key areas of investigation in developing honeyclients and network telescopes, including a discussion of some experiences with grid computing for scaling instruments and analysis. Internet Sensor Grids and network control infrastructure are becoming feasible, potentially using techniques from complex adaptive system theory, AI and biologically inspired system behaviour for deploying countermeasures. Even simple countermeasures could provide benefits through the application of filtering based on analysed IBR data.

Acknowledgments. The authors would like to acknowledge the work of research assistants Ben Palmer and Wayne Thomson, Tim Best, internee Jerome Selles, MSc

students Dean Pemberton and David Stirling, PhD students Christian Seifert and Van Lam Le in the development of tools, techniques and analysis for this paper.

References

[1] Park, K.: The Internet as a Complex System, Purdue University, Technical Report (2002)

[2] Park K., Willinger W.: The Internet as a Large-scale Complex System, Journal of the Royal Statistical Society: Series A (Statistics in Society), Volume 170, Number 1, pp. 260-260(1), Blackwell Publishing (2007)

[3] Navarez J., Seifert C., Endicott-Popovsky B., Welch I., Komisarczuk P.: Drive-By-Downloads, Victoria University of Wellington, New Zealand, Technical Report (2008).

[4] CERT, CERT/CC statistics 1998 to 2008. http://www.cert.org/stats/cert_stats.html.

[5] Clark, D., Shenker, S., Falk, A.: GENI Research Plan. GDD-06-28, Version 4.5 (2007)

[6] Seifert C.: Know Your Enemy: Behind The Scenes Of Malicious Web Servers. The Honeynet Project, http://www.honeynet.org/papers/wek/KYE-Behind_the_Scenes_of_Malicious_Web_Servers.pdf, (2008)

[7] Seifert C., Steenson R., Holz T., Yuan B., Davis MA.: Know Your Enemy: Malicious Web Servers. The Honeynet Project http://www.honeynet.org/papers/mws/KYE-Malicious_Web_Servers.pdf, (2008).

[8] Seifert C., Steenson R., Welch I., Komisarczuk P.: 'Capture - A Tool for Behavioural Analysis of Applications and Documents'. In: The Digital Forensic Research Workshop, Pittsburgh, PA, (2007).

[9] Seifert C., Welch I., Komisarczuk P.: HoneyC - The Low-Interaction Client Honeypot. In: the Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, (2007).

[10] The Honeynet Project, http://www.honeynet.org/misc/project.html

[11] Seifert, C: Improving Detection Accuracy and Speed with Hybrid Client Honeypots. PhD Proposal, Victoria University of Wellington, New Zealand, (2007).

[12] Komisarczuk, P., Seifert, C., Aval, C., Abbasi, F.: New Zealand Chapter Status Report For 2008. The Honeynet Project, Technical Report, (2009).

[13] Moore M., Shannon C., Voelker G.M., Savage S.: Network telescopes. Technical report, CAIDA, (2003).

[14] Plonka D. Flawed routers flood university of wisconsin internet time server. http://pages.cs.wisc.edu/~plonka/netgear-sntp/

[15] Pang R., Yegneswaran V., Barford P., Paxson V., Peterson L.: Characteristics of Internet Background Radiation. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pages 27–40. ACM Press, ISBN 1-58113-821-0 (2004).

[16] Yegneswaran V., Barford P., Ullrich J.: Internet intrusions: global characteristics and prevalence. In: Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 138–147. ACM Press, ISBN 1-58113-664-1 (2003).

[17] Komisarczuk P., Welch I., Pemberton D., Seifert C.: Grid Enabled Internet Instruments In: proceedings of IEEE Globecom (2007).

[18] Shadowserver, http://www.shadowserver.org/wiki/

[19] Team Cymru. The Team Cymru Darknet Project. http://www.cymru.com/Darknet/index.html and Dragon Research Group, http://drg.team-cymru.org/

[20] Dooley K.: Complex adaptive systems: a nominal definition, Arizona State University, Technical Report (1996)

[21] Foukia N., Hassas S.: Managing computer networks security through self-organisation: a complex system perspective. In: Engineering self-organising systems, Berlin, Springer, pp124-38 (2004).

[22] Green I., Raz, T., Zviran M.: Analysis of Active Intrusion Prevention Data for Predicting Hostile Activity in Computer Networks. In: Communications of the ACM, volume 50 Number 4, pp. 63-68 (2007).

[23] Glickman M., Balthrop J., Forrest S.: A machine learning evaluation of an artificial immune system. Evolutionary Computation, vol. 13, no. 2, pp. 179-212, (2005)

[24] Farrow R., Geer D.: Workshop Summary. In: HotBots '07, The First Workshop on Hot Topics in Understanding Botnets, USENIX, Cambridge MA. http://www.usenix.org/publications/login/2007-08/openpdfs/hotbots07sums.pdf, (2007).

[25] Shafi K., Abbass, H.A.: Biologically-inspired Complex Adaptive Systems approaches to Network Intrusion Detection. Inf. Secur. Tech. Rep. 12, 4, 209-217. (2007)

[26] Wolpert DH., Tumer K., Frank J.: Using collective intelligence to route internet traffic. In: Advances in Neural Information Processing Systems, (1999)

[27] Seifert, C., Komisarczuk, P., Welch, I., Aval, C. U., Endicott-Popovsky, B.: Identification of Malicious Web Pages Through Analysis of Underlying DNS and Web Server Relationships. In 4th IEEE LCN Workshop on Network Security (WNS 2008), Montreal, (2008).

[28] Stirling D., Welch I., Komisarczuk P.: Designing Workflows for Grid Enabled Internet Instruments. In: the 8th IEEE ccGrid 2008 Conference, Lyon, France, (2008).

[29] Bagnasco A., Poggi A., Scapolla A.M.: A Grid-Based Architecture for the Composition and the Execution of Remote Interactive Measurements. In: the Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), Amsterdam, Netherlands, (2006).

[30] Honeynet Project. Know your Enemy: Honeynets. http://prokect.honeynet.org/papers/honeynet/

[31] HoneyTrap, http://sourceforge.net/projects/honeytrap/

[32] HoneyD, http://www.honeyd.org/

[33] Yegneswaran Y., Barford P., Plonka D.: On the design and use of internet sinks for network abuse monitoring. In: Proceedings of Symposium on Recent Advances in Intrusion Detection, (2004).

[34] Cooke E., Bailey M., Watson D., Jahanian F., Nazario J.: The Internet Motion Sensor: A distributed global scoped Internet threat monitoring system. Technical report, Univerisity of Michigan, Electrical Engieering and Computer Science, (2004).

[35] Capture-HPC, https://projects.honeynet.org/capture-hpc

[36] Chard K., Tan J., Boverhof R., Madduri R., Foster I.: "Wrap Scientific Applications as WSRF Grid Services using gRAVI". In: the IEEE 7th International Conference on Web Services (ICWS), Los Angeles, USA, (2009).

[37] Seifert C., Komisarczuk P., Welch I.: Identification of Malicious Web Pages with Static Heuristics. In: the Australasian Telecommunication Networks and Applications Conference, (ATNAC), Adelaide, Australia, (2008).

[38] Nazario J.: PhoneyC: A Virtual Client Honeypot. In: USENIX, LEET Workshop, http://usenix.org/events/leet09/tech/ full_papers/nazario/nazario.pdf (2010)

[39] Seifert, C., Komisarczuk, P., Welch, I.: True Positive Cost Curve: A Cost-Based Evaluation Method for High-Interaction Client Honeypots. In the SECUREWARE conference, Athens, (2009).

[40] Le V.L., Komisarczuk P., Gao X,: Applying AI to Improve the Performance of Client Honeypots. In the Passive and Active Measurements Conference 2009, Seoul Korea, (2009)

[41] Moore D., Shannon C., Brown D.J., Voelker G.M., Savage S.: Inferring Internet denial-of-service activity. ACM Trans. Comput. Syst., 24(2):115–139. ISSN 0734-2071. (2006)

[42] Cooke E., Bailey M., Morley Z. Mao, Watson D., Jahanian F., McPherson D.: Toward understanding distributed blackhole placement. In: Proceedings of the 2004 ACM workshop on Rapid Malcode Analysis, pages 54–64. ACM Press, ISBN 1-58113-970-5. (2004)

[43] Harder U., Johnson M.W., Bradley J.T., Knottenbelt W.J.: Observing Internet Worm and Virus Attacks with a Small Network Telescope. In: Proceedings of the 2nd Workshop on Practical Applications of Stochastic Modelling, pp. 113–126, (2005).

[44] Man V.K: Clustering Malicious Networking Attacks. MSc Thesis, Victoria of Wellington, Wellington, New Zealand, (2008)

[45] IEEE, Computer Security Group (ICSG), Industry Connections Activity Initiation Document Version: 1.0 (Approved), http://standards.ieee.org/prod-serv/indconn/icsg/ICSG_ICAID.pdf

[46] WAND Network Research Group. The libtrace packet library. http://research.wand.net.nz/software/libtrace.php

[47] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org

[48] Pemberton D.: An Empirical Study of Internet Background Radiation Arrival Density and Network Telescope Sampling Strategies. MSc Thesis, School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, New Zealand (2007).

[49] Pemberton D., Komisarczuk P., Welch I.: Internet Background Radiation Arrival Density and Network Telescope Sampling Strategies. In the proceedings of the Australasian telecommunication Network Application Conference, 2007, Christchurch, New Zealand, (2007).

[50] MINDMAX. Maxmind - geolite city free geolocation database, http://www.maxmind.com/app/geolitecity