SSH – Somewhat Secure Host

Craig Valli

Security Research Institute, Edith Cowan University, Australia
c.valli@ecu.edu.au

Abstract. Honeypots are a proven technology for network defence and forensics. This paper focuses on attacks directed to network devices that utilise SSH services. The research uses the SSH honeypot Kippo to gather data about attacks on the SSH service. Kippo uses python and SSL to generate mock SSH services and also provides a filesystem honeypot for attackers to interact with. The preliminary research has found that attacks of this type are manifest, have a variety of profiles and may be launched from a variety of platforms.

Keywords: kippo, ssh, honeypot, python, cyber attack.

1 Introduction

Secure Shell (SSH) is intended to be a secure replacement for insecure plaintext methods and services including telnet, rsh and ftp by using point to point encrypted tunnels. There has been a noticeable increase in network borne threat against SSH services in recent years. The attacks essentially use automated approaches to attempt to login in to the service or daemon. Once logged in the attacker may chose to do an automated compromise of the account using various exploitation methods or tools such as Metasploit to deliver the payload.

The intelligence that a successful attack leaves behind is invaluable to the cyber security professional in discovering new methods of compromise or gathering of malicious code for forensic examination. By logging these types of attacks it also lets the potential victim gain insights into what information assets are potentially being targeted by the malcode. This paper will provide an overview of the systems utilised and an analysis of the data that these systems have provided.

2 The Development of the SSH Honeypot System

The systems utilised were based on a default install of Ubuntu Linux 11.04 as the base operating system. Various SSH honeypots were in fact trialled against simulated brute force attacks using a variety of tools. After this initial testing it was determined for the research that the kippo SSH honeypot system would be utilised.

The basic Ubuntu install then had the latest repository code for Kippo installed upon it as per the instructions on the kippo.googlecode.com Wiki. The mysql database suite was suitably configured, secured and used to record all interactions from the kippo honeypot. The mysql database structure is expressed in Table 1.

Y. Xiang et al. (Eds.): CSS 2012, LNCS 7672, pp. 227–235, 2012.

Table 1. MySQL database stucture for kippo honeypot

TABLE auth id int(11) PK, session char(32) NOT NULL, success tinyint(1) NOT NULL, username varchar(100) NOT NULL, password varchar(100) NOT NULL, timestamp datetime NOT NULL,	TABLE input id int(11)NOT NULL PK session char(32) NOT NULL, timestamp datetime NOT NULL, realm varchar(50) default NULL, success tinyint(1) default NULL, input text NOT NULL, KEY session (session,timestamp,realm)
TABLE clients id int(4) PK version varchar(50) NOT NULL	TABLE sensors id int(11) NOT NULL (PK) ip varchar(15) NOT NULL
TABLE sessions id char(32) NOT NULL PK starttime datetime NOT NULL, endtime datetime default NULL, sensor int(4) NOT NULL, ip varchar(15) NOT NULL default '',	TABLE ttylog id int(11) NOT NULL PK session char(32) NOT NULL ttylog mediumblob NOT NULL
termsize varchar(7) default NULL, client int(4) default NULL, KEY starttime (starttime,sensor)	

The database structure allows for the complete logging of all activity on the honeypot that relates to activity generated by an attacker.

A trial system was used to remove operational issues and develop a stable production system for use in the honeypot research. The data presented here was collected over a 4 month period. This was undertaken from February 2011 through to June 2011 which forms the data for the paper.

The current active systems utilise three home ADSL accounts located in Western Australia. Two ADSL accounts are from the same Internet service provider. In addition to these ADSL systems there are three Virtual Private Servers(VPS) one each in Singapore, Germany and USA. The active systems were progressively started over three months in the 2nd quarter of 2012 and as such no extensive geographical pattern analysis can occur at this point however the data collection is ongoing. It should be noted that the VPS servers and one of the home ADSL accounts do not serve anything to the Internet and the only outbound traffic is that which the researcher generates in shell interactions, extraction of attack data or when the system generates DNS query-all other traffic is malicious. Most of the active systems also have full capture of network traffic to disk using tcpdump with ports 222 and 443 being ignored as these are used for system administration. The DNS traffic is not excluded to enable the detection of any anomalous activity in DNS that may be generated by attacker interactions.

The kippo honeypot is intended to be a low interaction honeypot. It has a dictionary of both default and commonly used passwords that it uses to present a weakly configured system to the attacker. The system emulates a SSH session via the use of the python based twisted libraries to emulate cryptographic functionality and allows an attacking entity to attempt a login to the system believing it is entering into a legitimate SSH session. It should be noted that the Metasploit suite had a module that did reliably detect a kippo session due to issues in initiation of the faked encrypted session. Upon successful guessing of the password the attacker is then moved into a fake filesystem with which they can interact with.

In this fake system all interactions with the shell are monitored and recorded. The system also allows the use of wget and other commands commonly used to fetch or download code, and manipulate it on the “compromised host”. In essence through effective mimicry it is able to allow an attacker to login and interact with what they think is a real compromised host. It should also be noted that there are inconsistencies in how the fake system is presented and that an intelligent agent or human actor should quickly resolve that they are in a honeypot.

3 Known Attack Methods

The prevailing modus operandi is the use of brute force and utilisation of dictionary based methods. The dictionary or list method is the use of words or wordlists that are tried one after the other blindly against the victim account. These lists will typically use a dictionary word, known default password (such as admin) or common password strings from a keyboard pattern such as qwerty123456 or combinations thereof. Dictionary based methods are highly effective at compromising default installations on any number of network enabled devices and systems or systems that utilise poor passwords that fit this pattern.

Detection of automated dictionary based attacks can be a relatively simple task. Words are sometimes sorted sequentially A to Z or Z to A and attempted in this fashion. Chronology and magnitude of the attempts to compromise the account are indicators of brute force attack. A strong indicator is the intervals between retry of password for example timing that is not humanly possible to achieve or intervals of retry that are chronologically consistent from the same host.

The level of interactive human sessions is minimal compared to automated attacks. The honeypot system recorded activity it predicted as human in less than X% in the initial trial. Further desktop analysis confirmed these predictions to be accurate.

4 Attack Outcomes and Intelligence Gathered from Trial Systems

The initial test system data is used in this section to illustrate the types of data that can be garnered from the honeypot. The following data and graphs are generated from the test system. Note this traffic is the actual Internet based activity not traffic generated in initial laboratory experimentation to benchmark and test the system.

One of the best ways of determining the type of attack is determining the type of SSH client that is being used to connect and hence determine the operating system used in the attack. It should be pointed out this is not an absolute determinant of the type of SSH client that is used but it is often indicative of the platform. It is indicative because client strings can be faked or modified by the attacker to mask the true identity of the attacking platform. In some cases no banner is sent at all. Detected clients and frequency are displayed in Figure 1 below.

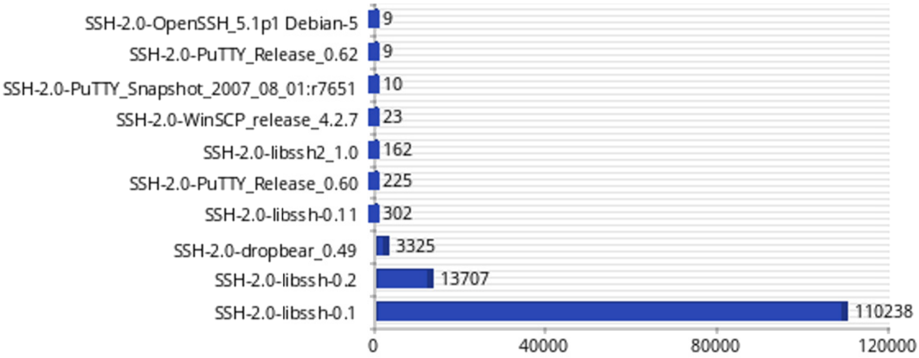

Fig. 1. Top SSH Clients by Client string connecting to the honeypot

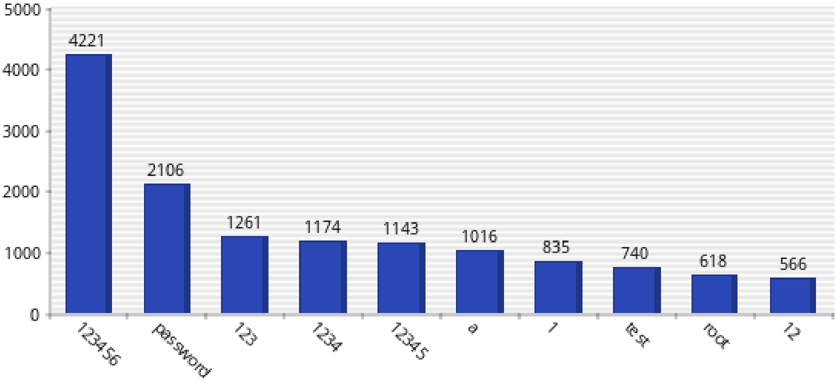

Fig. 2. Top 10 Passwords attempted on the honeypot

Over 97% of the attack profile of detected SSH clients are SSH 2.0 libssh either 0.1 or 0.2. The other identified clients would typically seem inconclusive, however, there is something to be gleaned here. The libssh – X.X , dropbear, OpenSSH (Debian) are well known Linux based libraries for SSH, the remainder are Windows based clients. Also 23 of the contacts with the honeypot were made by the secure copy (scp) WinSCP client which is not normally used for shell based interactions on SSH based servers.

A successful attack in the context of this research is being able to login as a user on the honeypot system, this then allows the user to interact with the fake shell account. The honeypot uses a dictionary of commonly attempted passwords. The honeypot administrator can extend the dictionary at anytime, however, the research so far has used the defaults for the honeypot as supplied with the source code for kippo.

From the test system the success of attacks was not high comparative to the attempts. The 10 most commonly attempted passwords are in Figure 2.

The top 10 passwords and username combinations that were successfully presented to the honeypot are presented in Figure 3 below.

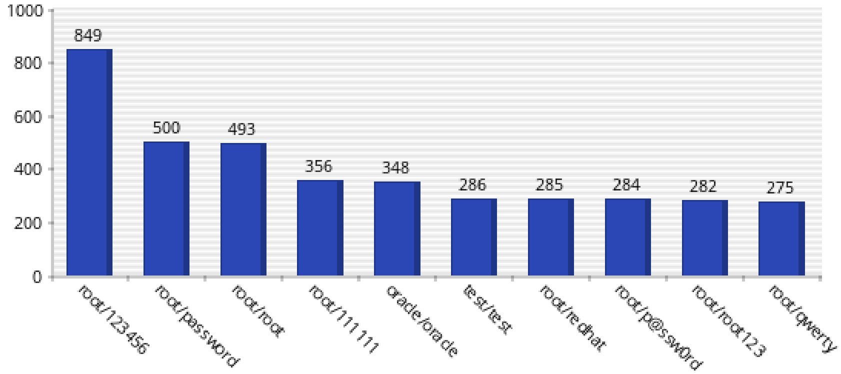

Fig. 3. top 10 password and user combinations

Both Figure 2 and 3 present some interesting information. For example the password 123456 was used 4221 times yet only 849 of these were used against the root account, which gained access to the shell account. Likewise root as a password was used in the root/root combination 493 times, yet the word root was used a further 125 times with other account names.

Default password lists are widely distributed on hacking sites. These default passwords are for default settings on operating systems, services run by the operating systems, default user accounts on operating systems and network devices e.g. ADSL routers. The success of some network borne malware can be readily attributed to poor security i.e. not removing default user name and passwords from default install. One of the worrying trends is the compromise of ADSL modems and routers that are in default configuration as highlighted by (Szewczyk 2009; Szewczyk 2011). The initial research here also shows that ADSL routers appear to be the victims of this. There are many contributing factors for this but one of the standout ones is the lack appropriate security documentation provided with these modems and routers (Andersson and Szewczyk 2011).

From a honeypot intelligence and design perspective the researchers will be running one system with updates to the dictionary of passwords drawn from attempts by intruders. The objective behind this is to allow successful incursion into the system to entrap attack automata, human attack and reconnaissance, and allow capture of malcode for analysis of characteristics. The existing systems will be left as they are to trap longitudinal data on the exploit of SSH. It will enable the research to identify patterns of rescan and also intelligent, self learning systems that do not retry previously failed passwords but instead tries new passwords.

After successful passwords are entered the honeypot system then puts the attacker into a virtual shell environment where the user can access and execute commands. Figure 4 shows the ten most successful inputs to the shell from the attackers.

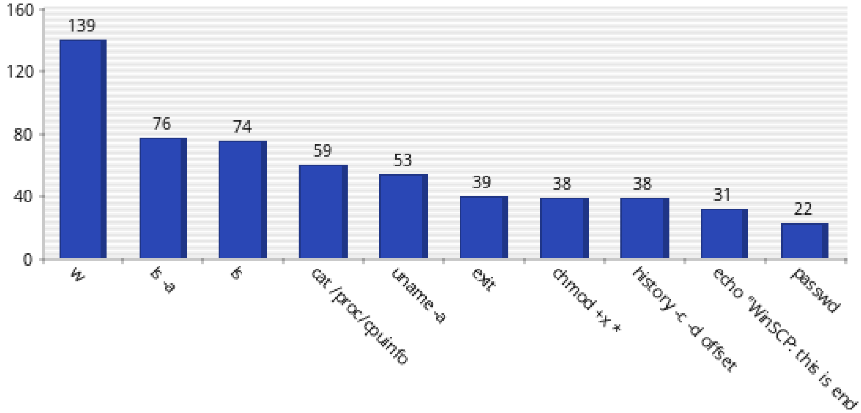

Fig. 4. Top 10 executed commands by the attacking entities

The w command is the most frequently used indicating that the attacking entity is gathering basic system intelligence. An example follows

```
08:59:17 up 18 min, 2 users,  load average: 0.02, 0.10, 0.20
USER      TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
adminuser tty1                      08:42   16:25   2.06s  1.82s -bash
adminuser pts/0    192.168.207.1     08:43    0.00s  1.78s  0.03s w
```

It describes the load, uptime, users logged in, users resource usage and programs or scripts the user is executing at that time. This has a number of uses from an attack perspective i.e. it demonstrates normal system patterns or user patterns of utilisation of the server.

The next two most commonly used commands are ls –a and ls. The command cat /proc/cpuinfo which extracts details about the CPU running the machine an example follows.

As demonstrated above the command gives a comprehensive overview of the processor running the server. This information can be used to identify server platform or in some cases that it is a virtual machine or even a honeypot system. The next command is uname which is UNIX program that prints system information the –a switch is to print all information. This information is kernel name, node name, kernel-release, kernel-version, machine hardware name, processor type, hardware platform, operating system an example follows:

```
  Linux ssh-hp 2.6.35-22-generic-pae #33-Ubuntu SMP Sun Sep 19 22:14:14
UTC 2010 i686 GNU/Linux
```

Table 2. /proc/cpuinfo

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 15
model name      : Intel(R) Core(TM)2 Duo CPU     L7500  @ 1.60GHz
stepping        : 11
cpu MHz         : 1601.000
cache size      : 4096 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 10
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss nx lm constant_tsc up arch_perfm
on pebs bts tsc_reliable aperfmperf pni ssse3 cx16 lahf_lm ida
bogomips        : 3202.00
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:
```

As can be clearly seen by the example the information provided is extensive and then allows an attacker to start refining attack parameters. As shown in Table 2 above it would be useless to try to attempt a compromise based on Linux 2.4.X kernel on a Sparc CPU. Likewise this information can be used to determine if the service is potentially running on server, desktop PC, router, virtual machine or honeypot.

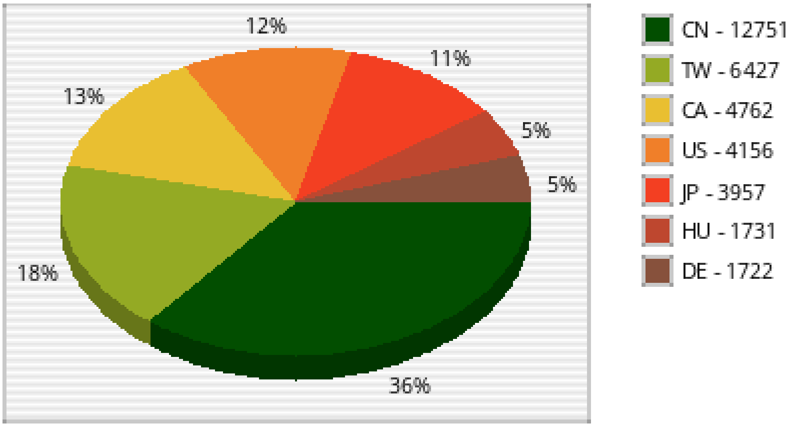

Fig. 5. Countries by number of connections

The history command is an attempt to see what standard users are doing or running on the system (i.e. what applications they are using the account for).

The detail in figure 5 below outlines the number of connections per country. The origin of the attacks utilises the geo-ip functionality in Linux platforms which is sourced from Maxmind to determine location. This uses a combination of known knowns in the form of known IP within a known assigned routing range.

The data indicates that there is a predominance of traffic being generated in China (CN) and Taiwan (TW) approximately 54% of all traffic. It should be noted that China and Taiwan do not have 54% of the worlds population or even Internet users . This data set indicates at 54% there is a disproportionate amount of attacks being generated from China and Taiwan.

5 Future Work

The project has several avenues of exploration beyond mere collection of SSH connection and actor interaction statistics. Once sufficient comparative data is gathered from the multiple sources it will be possible to do pattern and trend analysis on high frequency actors who interact with the honeypots. It should then be possible to build up methods for identifying particular *modus operandi* for a variety of perspectives.

As the attackers interact with the single hosts themselves and then the wider network of hosts. This will allow the researcher to identify behavioural patterns of attack that could lead to methods to identify more reliably the attacking entity be it automated, semi-automated or manual in attack profile. Certain pieces of code will potentially effectively fingerprint due to their interactions with the honeypot. This fingerprinting will require deep packet inspection or similar close analysis of every atomic attack chain which in turn may in of itself be capable of some automated analysis.

Methods to ensure greater deception are needed as kippo honeypot detection modules have been incorporated into the widely used Metasploit framework. The existing known detection has been thwarted through better implementation of the mimicked SSH interactions in the python twisted libraries with code updates. There are still some weaknesses in the primary mimicry of the system as it stands but it is hoped that this research will contribute to amelioration of these. The development of an expanded filesystem mimic will also add credibility to the deception in the honeypot. Expanded development and use of tools, where possible making the interactions again seem more credible. As previously mentioned the use of an expanded password list to allow increased gathering of attack intelligence and malcode for analysis. Finally, a gradual expansion of the entire honeypot network across various time zones and geographical locations. In addition this also within the same time zones and network backbones but with increased geographical and network topographical diversity.

References

Andersson, K., Szewczyk, P.: Insecurity By Obscurity Continues: Are ADSL Router Manuals Putting End-Users At Risk. In: Williams, T., Valli, C. (eds.) The 9th Australian Information Security Management Conference, Citigate Hotel, Perth, Western Australia, Secau - Security Research Centre, Edith Cowan University, Perth, Western Australia 19-24 (2011)

Szewczyk, P.: ADSL Router Forensics Part 2: Acquiring Evidence. In: The 7th Australian Digital Forensics Conference, Kings Hotel, Perth, Western Australia, Secau - Security Research Centre, School of Computer and Security Science, Edith Cowan University, Perth, Western Australia (2009)

Szewczyk, P.: Analysis of Data Remaining on Second Hand ADSL Routers. Journal of Digital Forensics, Security and Law 6(3), 17–30 (2011)