2013

Patterns and Patter - An Investigation into SSH Activity Using Kippo Honeypots

Craig Valli
Edith Cowan University, c.valli@ecu.edu.au

Priya Rabadia
Edith Cowan University, prabadia@our.ecu.edu.au

Andrew Woodward
Edith Cowan University, a.woodward@ecu.edu.au

DOI: 10.4225/75/57b3dbc8fb877
11th Australian Digital Forensics Conference. Held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

PATTERNS AND PATTER - AN INVESTIGATION INTO SSH ACTIVITY USING KIPPO HONEYPOTS

Craig Valli, Priya Rabadia and Andrew Woodward
Edith Cowan University, Security Research Institute
Perth, Australia
c.valli@ecu.edu.au, prabadia@our.ecu.edu.au,a.woodward@ecu.edu.au

Abstract
This is an investigation of the activity detected on three honeypots that utilise the Kippo SSH honeypot system on VPS servers all on the same C class address. The systems ran on identical software bases and hardware configurations. The results are over the period 21st March 2013 until Tuesday 04 June 2013. The initial analysis covered in this paper examines behaviours and patterns detected of the attacking entities. The attack patterns were not consistent and there was large disparity in numbers and magnitude of attacks on all hosts. Some of these issues are explored in the paper.

Keywords SSH, honeypot, Kippo

INTRODUCTION
This is an investigation of the activity detected on three honeypots that utilise the Kippo SSH honeypot system(Tamminen 2013) on VPS servers. The three honeypots were all configured identically all using Ubuntu 11 server as the operating system, identical hardware in terms of CPU memory and hard disk. The three honeypots were all deployed across the same IP version 4 /24 subnetwork space a /24 on virtual private servers (VPS). The three honeypots are referred to as Mopoke (.13), Quokka (.234) and Lair (.78); they are each configured on a VPS. This paper is part of an ongoing investigation into SSH honeypots and builds on experimentation and initial work done in 2012(Valli 2012).

The aim of this investigation was to seek an understanding of the threats against SSH services.

OVERVIEW OF THE SSH HONEYPOT DESIGN AND SYSTEM DESIGN
A kippo SSH honeypot is a medium interaction honeypot . This implies that the honeypot imitates some functions that are exhibited by a 'real' system (Stevens and Pohl 2004, Tamminen 2013). An example of a functionality that is imitated is SSH (Secure Shell). SSH is designed to securely transmit data using a point to point encryption tunnel(Ciampa 2010). Kippo honeypots collect data from SSH service attacks(Code.google.com 2012). An open-source, python based event-driven program called Twisted libraries(TwistedMatrixLabs 2013) is deployed to duplicate a SSH session.

The basic Ubuntu install then had the latest repository code for Kippo installed from kippo.googlecode.com Wiki. The servers were setup according to the guide from BruteForce Labs utilising the authbind daemon(Koniaris 2011).

The mysql database suite was suitably configured, secured and used to record all interactions from the kippo honeypot on each of the servers, locally. The mysql database structure is expressed in Table 1.

TABLE auth	TABLE input
id int(11) PK,	id int(11)NOT NULL PK
session char(32) NOT NULL,	session char(32) NOT NULL,
success tinyint(1) NOT NULL,	timestamp datetime NOT NULL,
username varchar(100) NOT NULL,	realm varchar(50) default NULL,
password varchar(100) NOT NULL,	success tinyint(1) default NULL,

timestamp datetime NOT 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 '', termsize varchar(7) default NULL, client int(4) default NULL, KEY starttime (starttime,sensor)	TABLE ttylog id int(11) NOT NULL PK session char(32) NOT NULL ttylog mediumblob NOT NULL

Table 2 - Mysql Database Stucture For Kippo Honeypot

The database structure allows for the complete logging of all activity on the honeypot that relates to activity generated by an attacker. In addition the Postgresql SQL server python modules were added to the Kippo SSH honeypot and all of the servers pushed their data to a centralised server. The centralised server is utilising the SURFnet IDS(Surfnet 2013) logging server SQL database schema for storing the data from the honeypots.

The Kippo honeypot is intended to be a low interaction SSH based honeypot. It has a dictionary of both default and commonly used system login 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 that would be found in a normal SSH session initiation.

Kippo allows an attacking entity to attempt a login to the system believing it is entering into a legitimate SSH session with the server. Upon successful guessing of the password the attacker is then moved into a fake system 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 files, and manipulate it on the "compromised host" as well as a base set of utilities. In essence through effective mimicry it is able to allow an attacker to login and interact with what they think is 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.

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 using the twisted module in python (Code.google.com, 2012). This problem has now been fixed in the current version of Kippo. This issue when active would have alerted automated attackers to the fact that they had probed a honeypot.

KNOWN ATTACK METHODS

The prevailing *modus operandi* is the use of both brute force and dictionary based methods to try and guess the login and password for the servers. Brute force attacks attempt to find a password by starting with a single alphanumeric character and cycling through every possible character, number and non alphanumeric character sequentially, increasing by one additional character as each

previous password length is exhausted. Such an attack is significantly time consuming, and is limited by factors such as bandwidth and the length and complexity of the password will significantly increase the time required. Unless the password is quite a short, and only uses lower case letters, for example, a brute force attack would simply be unviable. For these reasons, this type of attack is now frequently limited to offline attacks, and employs high speed cracking software and hardware such as GPGPU based systems.

The dictionary based or list based methods use words or wordlists that simply try one password after the other blindly against the victim account on the target system. The speed of these methods is typically restricted by the available network bandwidth to the target system. In addition to network speed the target system's ability to cope with multiple connections before a denial of service occurs as a result of CPU exhaustion is also a constraint. On some systems where countermeasures exist the system may react and then shutdown further connections.

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. Customisation of these wordlists is common, and there are numerate customised wordlists available for download on the Internet. Dictionary based methods are highly effective at compromising default installations on any number of network enabled devices and systems or system that utilise poor passwords that fit this pattern.

Detection of automated dictionary based attacks can be a relatively simple task. Firstly, 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 it is not humanly possible to achieve. Another indicator is the intervals of a login retry that are chronologically consistent from the same host or multiple hosts as a group connecting.

The other attack against an SSH server of course is one that utilises vulnerability whether that be known or unknown (zero day) to cause either privilege escalation or denial of service. It should be noted that while Kippo is a honeypot it is not meant to provide emulation at this level. However, the honeypots are configured with full packet capture allowing post incident analysis for this type of attack. Analysis via full packet capture is not a current focus of this research however.

ATTACK OUTCOMES AND INTELLIGENCE GATHERED FROM THREE HONEYPOTS

The period of coverage for this research was from 21st March 2013 until Tuesday 04 Jun 2013, a period of 75 days. The three hosts have all produced three very different sets of data. This outcome is somewhat incongruous given that all of the hosts are from the same C class network. The hosts are Mopoke XXX.XXX.XXX.13, Lair XXX.XXX.XXX.78 and Quokka XXX.XXX.XXX.234. All of the hosts run the same revisions of software for the base systems and respective operating systems. They are all on identical network links in terms of latency and other critical factors. As best can occur in a networked IT system they are identical.

Overall during the period examined the hosts sustained the following login attempts Mopoke 210,586, Lair 68,417, Quokka 12,770 to their respective honeypots. The numbers of distinct IPs per host are for Quokka 174, Lair 326 and Mopoke had 999.

The honeypots are front loaded with passwords from various known bad databases that will allow login to the honeypot system to interact with the fake command shell. Successful logins achieved per host over the period were Mopoke 663 or 0.314% of all login attempts, Lair 172 or 0.251% of all login attempts and Quokka 78 or 0.610%.

Top 20 Attackers

The following table represents the attacking IPs detected by the three hosts (Table 2). One of the clearly identifiable anomalies is that the top scanning hosts IP 176.31.85.59 for Lair (28565) and Mopoke (25927) which would be indicative of hard scanning of a network class. However for Quokka there were no scans from this IP address at all recorded during the period covered. It should be also noted that these scans were spread over the time period in which data was collected.

Mopoke			Lair			Quokka	
176.31.85.59	25927		176.31.85.59	28565		94.127.2.85	1605
223.4.94.69	13322		173.224.221.197	5310		211.191.168.180	1604
223.4.241.4	10771		46.105.189.201	4703		188.132.206.233	974
74.63.238.68	9507		5.9.200.90	3282		121.9.221.102	974
223.4.171.195	8434		203.93.215.101	2238		211.110.44.113	771
223.4.175.77	6616		94.127.2.85	1605		101.44.1.135	725
59.125.208.244	6097		188.132.206.233	1461		174.140.167.238	603
220.172.107.211	6048		121.9.221.102	974		218.206.117.57	439
201.116.36.180	5777		58.240.17.250	961		74.84.89.106	286
122.158.235.33	4848		114.112.21.15	953		114.80.202.30	275
116.229.239.242	4772		218.206.117.57	854		84.22.32.2	249
223.4.182.71	3535		91.102.16.156	798		31.40.76.183	222
200.214.143.4	3520		166.78.27.149	725		220.161.148.178	216
174.140.167.238	3442		208.115.207.140	627		112.231.23.68	201
188.125.103.177	3211		128.140.1.19	597		124.160.194.27	174
180.168.83.54	3094		220.161.148.178	593		192.157.220.84	169
5.9.200.90	3075		120.101.5.209	546		62.193.192.167	163
223.4.145.38	2888		218.77.178.3	545		63.137.151.184	163
203.69.139.179	2664		59.151.5.236	511		173.208.218.70	161
218.213.234.232	2519		211.191.168.180	449		125.39.8.142	141

Table 2 - Top 20 Connecting IPs for Hosts

Figure 1 shows the spread of hosts probed by the top 20 attacking IPs for each host. Purposeful brute force scanning is indicated by the scanning of hosts by the attacking IP with termination of the number of actual login attempts at equal count, this represents 11% of all scanned IPs. This pattern would be indicative of simplistic brute force scripts. Further, analysis into these login attempts support this hypothesis as the password sets used were consistent across the brute forced attempts on the attacked hosts.

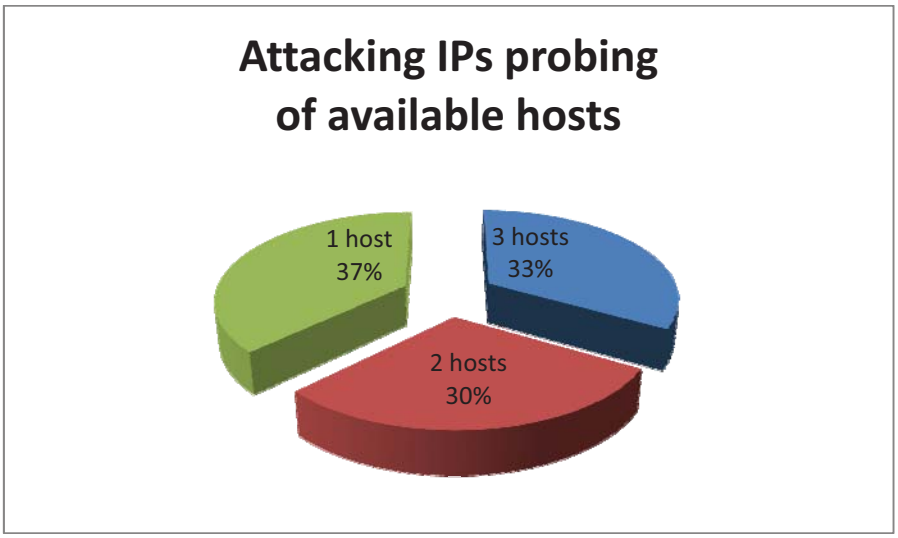

Figure 1 – Honeypot hosts probed by attacking IPs

Geolocation of the attacks presents interesting patterns of attack in this particular research, some of which is in contrast to some of the rhetoric, fear, uncertainty and doubt around cyber security threats. The general assumption touted in the tabloid press is that attacks are numerous and predominantly from Chinese based hosts. Whilst there is some truth in these statements, this research presents evidence which contradicts this viewpoint (Table 3).

1	France	30%		6	USA	4%		11	China	3%		16	Unknown	2%
2	China	7%		7	Poland	4%		12	Mexico	3%		17	Korea	2%
3	China	6%		8	China	4%		13	China	3%		18	USA	2%
4	USA	5%		9	Germany	4%		14	China	3%		19	Turkey	2%
5	China	5%		10	Taiwan	3%		15	China	3%		20	China	2%

Table 3 – Geolocation of Top 20 attacking IPs for all honeypots

The biggest attacking single source IP was in fact France with 30% of attacks on this infrastructure emanating from the top 20 attackers. China however, did represent 8 of the top 20 attackers representing 25% of the volume, but note this is still less than France overall. There were 9 nations involved in the overall attack profile from across the globe, pointing to the global nature of the problem of cyber security.

Attacks associated with password vulnerabilities

The primary and often only defensive barrier on systems is that of a password. For a password to be secure it should typically be over 12 characters long and be complex in its construction using a random combination of letters, digits and punctuation. This complexity would also preclude dictionary words or simple patterns for example based on a keyboard sequence e.gqwerty or a sequence of numbers and letters e.g abc123. There is also publicity about passwords that are insecure and problematic with lists of insecure passwords posted on reputable websites and news services(Anonymous 2012, Harley 2012, King 2012) in attempts to educate users.

The Kippo honeypot is optimised to use these lists of passwords, so it is no surprise in the following table that these words contained in the aforementioned bad password lists appear here numerate times.

Mopoke		Lair		Quokka	
123456	7350	123456	1632	123456	597
password	6204	password	803	password	157
1234	3481	changeme	539	1234	125
changeme	2757	test	421	12345	101
abc123	2592	1234	369	1	90
12345	2414	123	361	123	89
test	2300	12345	311	test	84
123	1904	qwerty	291	root	80
qwerty	1342	admin	279	abc123	70
P@ssw0rd	1102	test123	276	qwerty	68

Table 4 – Top 10 passwords attempted on the honeypots

The top 10 passwords from each of the three hosts display some similarity in the attack approach. Passwords attempted were obvious keyboard patterns or patterns e.g 123456, qwerty and abc123. Other passwords were account names that would have privileged high level access such as root and administrator. It should be noted that many unsecured default systems will have for instance root, root or admin, admin combinations for username and password respectively. The remaining passwords detected in the top 10 were bad choices either because of length or obvious nature for instance changeme.

Interactions with hosts

Once attacking entities had gained access to the honeypot a variety of access was evinced from clearly automated interactions to obvious manual probing. The automated interactions tended to use scripts that endeavoured to download payloads from servers and deploy malicious code.

One of the features of the Kippo honeypot is the false system that allows the attacker to interact with the honeypot system as they would expect to when they have compromised a real machine on the Internet. The fake system provides a limited set of commands within the false machine. Also false machine information is fed via static files representing a system. For instance /proc/cpuinfo will output realistic details to the attacker. One of the commands available is *wget* that's allows the attacker for instance to download files to the compromised host, for one example. The downloading files for compromised host allows an attacker to further compromise a system and try to maintain permanent access to the assets they have now compromised.

The following table represents commands other than file fetch commands such as *wget* and *ftp* that were detected once an attacker had successfully logged in to the false machine.

Mopoke		Lair		Quokka	
ls	110	exit	3	ls	31
w	70	ls -al	2	cd ..	14
exit	54	ls	1	exit	11
ls -a	43	cat /proc/cpuinfo	1	ls -al	6
cd ..	25	cat /etc/issue	1	cd /	4
uname -a	24			ls -l	4
rm -rf .bash_history	24			ls -lah	4
echo "WinSCP: this is end-of-file:0"	23			clear	3
ls -al	21			ls -all	3
chmod +x *	21			ps	3

Table 5 - Top 10 commands executed on the honeypots once logged in

Across the three hosts most of the commands were simple commands gathering basic intelligence such as directory listings to search for interesting directories that may contain files of interest. In

Mopoke there was some malicious command usage detected firstly attempts were made to remove the *.bash_history* file, which is a common technique to hide malicious activity of an attacker. Typically all interactions on a Linux/UNIX based server dialogue to the *.bash_history* file. It is interesting to note that the command given *rf –r* simply deletes the file. This is of note because deleting and not erasing would allowing for forensic recovery of the file.

The obvious second attempt was when attacking entities tried to modify permissions on files by making every file executable with the *chmod +x ** command. Interestingly, on Lair the attacking entity attempted to gain advanced information through the enumerating */proc/cpuinfo* to elicit machine specification with respect to CPU in use.

Malicious software

As previously mentioned the system allows for the use of *wget* and *ftp* commands to download files typically malicious code to the honeypot. This feature enables researchers to capture some of the latest malcode available on the Internet. The system downloads file into a sandbox environment and also logs the interactions. In this research so far Mopoke had 53 files downloaded of which 42 were unique. During the period monitored both Lair and Quokka were not compromised by an attacking agent that attempted to download files. The following table (Table 6) is a list of all downloaded files to the host, note if no count is specified it is a single attempt.

Download	Count	
http://www.uniw.com/flood.tar	5	http://rapk1d.webs.com/key.pl
http://download.microsoft.com/download/win2000platform/SP/SP3/NT5/EN-US/W2Ksp3.exe	2	http://root-arhive.clan.su/scanner/debian.jpg
http://iceagewar.hi2.ro/psyBETA.tgz	2	http://sacele.ucoz.ro/Mech/Mech.tgz
mariusica.altervista.org/lib.tt	2	http://system.comule.com/bnc/psybnc-linux-ro.tgz
www.xdutzzux.altervista.org/udp.pl	2	http://tehgame.altervista.org/rk2012.tgz
dinte.webng.com/unic.tgz		http://w0rmer.altervista.org/bot.txt
doiaru.clan.su/clean.tgz		http://wmbro.webs.com/x.tgz
dracdeinger.altervista.org/mm.tgz		linuxtrade.us/
dracdeinger.altervista.org/rk.tgz		LinuxTrade.Us/Arhive/Others/pico.tgz
ftp://mestahost:mestahost2013@188.120.237.29/Smith.tgz		linuxtrade.us/authorized_keys
http://183.60.202.196:521/hello.exe		linuxtrade.us/mech.tgz
http://198.2.192.204:22/disknyp		linuxtrade.us/mh.tgz
http://31.170.164.152/abc/fast.jpg		linuxtrade.us/x.tgz
http://abc.hol.es/abc/fast.pdf		marius.altervista.org/scanner/HaitaTeam.jpg
http://cachefly.cachefly.net/100mb.test		smithboy.webs.com/emech/psybnc.gz
http://d4ng3r.byethost6.com/stuff/FDSK.zip		w0rmer.altervista.org/arhiva.tgz
http://geocities.ws/map/perl.pdf		wbo.clan.su/chx/ssh.gz
http://hacker-linux-mi.clan.su/gosh.tgz		www.parazit.eu/p/psy.tgz
http://pinky.clan.su/sniff/rk.jpg		www.unix.com/flood.tar
http://psybnc.at/download/beta/psyBNC-2.3.2-7.tar.gz		xx.ucoz.com/xx.pdf
http://rapk1d.webs.com/100mb.tgz		zuby.go.ro/boti.tgz

Table 6 – List of files downloaded to the honeypot hosts

It should be noted that the multiple downloads as listed were from actually different attacking hosts. Many of the files were variously: rootkits (rk.tgz), denial of service tools (flood.tar), IRC tools (psyBNC) and compromise replacement binaries (ssh.gz). There was also PDF and JPG files that contained malicious payload principally targeted at the Microsoft Windows platform. The predominance of file transfers were also attempted by http with only one specifically targeting *ftp* as the transport protocol ftp://mestahost:mestahost2013@188.120.237.29/Smith.tgz

DISCUSSION AND CONCLUSION

The honeypot systems have detected a wide range of malfeasance by attacking entities. However, there are significant incongruities with some of the commonly held views or opinions which are frequently stated as fact in relation to scanning and probing of internet facing hosts. The results of this research have provided evidence to refute some of these claims.

The first misconception is that there is widespread automated scanning of the Internet by bots or automated scripts typically initiated by a human being across entire network blocks. The evidence acquired by this research to this point would indicate that this is not the case for the majority of attacks. These three identical hosts were on the same subnetwork with varying levels of probing and attack. If there was widespread scanning one would postulate that the top 10 or even top 20 attacking IPs for such a longitudinal study would be the same or exhibit high similarity, this has not been the case. However, there are only 4 or 11% of hosts that did scan all three hosts that terminated their connection once a particular limit had been reached, which does indicate some automated scanning did occur.

The second misconception is that many of the techniques are advanced and that the malicious code is likewise advanced. Some of the behaviours exhibited in this research would indicate that the attacks are at least persistent but certainly calls into question their advanced nature. There is however, a word of caution in that these machines were Linux based and not Windows-based machines which are attacked more often. In the downloaded code there was numerous payloads detected that specifically targeted Windows based systems however, using Rumsfeldian logic these were “known knowns”. The “known knowns” would have been readily detected by any base level antivirus software with an up-to-date signature base. This finding also adds weight to the argument that attacks are not sophisticated in orientation.

In conclusion, this is a preliminary analysis of an extensive dataset produced by three honeypots in over just 75 days. The analysis of the data at this stage has been limited to analysing high level logging data recorded by the honeypots with some in-depth analysis where required. There has not been an extensive forensic review of log files and other data that was collected in the honeypot systems at this stage. This data will need further research and development of suitable analysis techniques and methods. Finally, the honeypots are still collecting data from attackers and at time of writing had a combined database approaching 1 million attempts, which will require novel techniques for longitudinal analysis.

REFERENCES

Anonymous (2012). "Password' Is The Most Popular Password Of 2012." Retrieved 23 October, 2013, from http://www.huffingtonpost.com/2012/10/25/most-popular-worst-passwords-of-2012-splashdata_n_2018587.html.

Ciampa, M. D. (2010). Security Awareness: applying partical security in your world. Boston, Course Technology.

Code.google.com (2012). "Kippo shows up in Metasploit." SSH Honeypot Retrieved 23.09.2013, from https://code.google.com/p/kippo/issues/detail?id=48.

Harley, D. (2012). "Passwords and PINs: the worst choices." Retrieved 23 October, 2013, from http://www.welivesecurity.com/2012/06/07/passwords-and-pins-the-worst-choices/.

King, R. (2012). "25 most-used passwords revealed: Is yours one of them?" June 8, 2012. from http://www.zdnet.com/blog/security/25-most-used-passwords-revealed-is-yours-one-of-them/12427.

Koniaris, I. (2011). "Installing Kippo SSH Honeypot on Ubuntu." Retrieved July 10th, 2013, from http://bruteforce.gr/installing-kippo-ssh-honeypot-on-ubuntu.html.

Stevens, R. and H. Pohl (2004). "Honeypots und Honeynets." Informatik-Spektrum**27**(3): 260-264.

Surfnet (2013). "SURFnet IDS." from http://ids.surfnet.nl/wiki/doku.php.

Tamminen, U. (2013). "Kippo SSH Honeypot." Retrieved 09.10.2013, from http://code.google.com/p/kippo/.

TwistedMatrixLabs (2013). "What is Twisted?". Retrieved 23.09.2013, from http://twistedmatrix.com/trac/.

Valli, C. (2012). SSH: somewhat secure host. Proceedings of the 4th international conference on Cyberspace Safety and Security. Melbourne, Australia, Springer-Verlag**:** 227-235.