Getting back at Trudy

SSH Botnet Member Credential Collection using Connect Back Honeypots

University of Amsterdam

Tobias Fiebig
tobias.fiebig@os3.nl

February 11, 2013

Abstract

This paper introduces and tests a novel technique for gathering the credentials of systems used in SSH bruteforce attempts by echoing the credentials send to a honeypot back to the attacking system. The technique is implemented and tested in a real-world scenario. The drawn conclusions allow new insights into the modus operandi of groups conducting SSH bruteforce operations.

Keywords: SSH; Offensive Technologies; Botnets; Honeypots; Security;

1 Introduction

Bruteforce break in attempts are a constant annoyance on the internet [11, p. 6], and the idea of breaking password-based authentication mechanisms by probing plausible and weak passwords is nearly as old as these mechanisms themselves. One of the first descriptions of the concept of password guessing based bruteforce attacks can be found in a paper by Morris and Thompson published as early as 1979 [12, p. 595]

SSH, the Secure SHell, is a popular network protocol for secure data communication with a variety of systems [1, p. 2]. The base protocol has been specified in RFC4251 [21].

Previous research on SSH bruteforce Systems and Botnets has been concerned with different non-offensive techniques for getting greater insights into the modus operandi of the attackers. This includes purely passive techniques as implemented by e.g., Owens [13], who gathered bruteforce attemps in order to identify the wordlists used by the SSH bruteforcers. Other attempts include honeypots that actually allow an attacker to penetrate the system, in order to observe the attackers actions on the infected systems. This has already been implemented by Owens in 2008 [13], although he did not utilize the SSH bruteforce attack vector as entry point for the attacker.

More recent techniques in this direction include the Kojoney [2] software as well as the Kippo [19] software. The first one aims at a general overview of the inbound attacks on a network, simultaneously providing an attacker with the impression of a successful penetration, whereafter the commands issued by the attacker can be analyzed. The latter one provides a full sandbox environment, in which the attackers actions can be thoroughly analyzed.

There is, however, no indication in the literature for active mechanisms that allow the penetration of the attackers system.

1.1 SSH Bruteforcing Nodes

The systems used by attackers are scattered over all parts of the internet [17]. Owens already established that leaving a system vulnerable may lead to an unknown attacker utilizing the system for SSH bruteforcing after successfully penetrating it.

This leads to the hypothesis that systems penetrated by SSH bruteforcing may be used to execute the same technique they have been penetrated with. This theory is backed up by research done by Ramsbrock, Berthier and Cukier, who discovered that attackers first download and then install

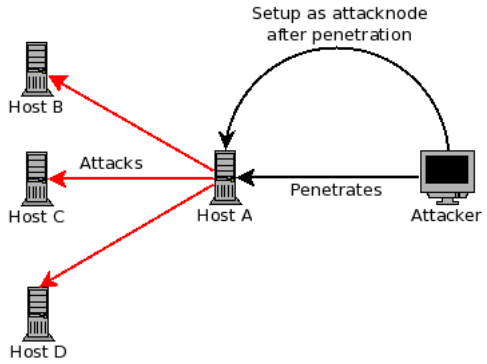

Figure 1: After successfully penetrating a new host the attacker configures it to launch additional bruteforce attacks towards other hosts.

rogue software after the successful compromisation of a system [16]. See Figure 1 for a graphical representation.

As soon as an attacker penetrated a node, subsequent detection would lead to the box being cleaned up and being unaccessible to the attacker. If the attacker would change the weak password that granted access to the system, the legitimate owner would notice that he is unable to log in. Although the research done by Ramsbrock, Berthier and Cukier denies this theory, stating that the majority of attackers changes the password [16, p. 6], this paper assumes that the majority of those attackers is either detected fairly quickly or the passwords are changed back to the original state by the authorized user of that account, without detecting the compromisation.

The last assumption is, that an attacker uses only one wordlist and does not remove the password with which the system was compromised from the wordlist when he starts bruteforcing from that system.

1.2 Research Question

It is therefore plausible to assume that the credentials for a significant fraction of all SSH bruteforcers currently active on the internet can be determined by echoing their login attempts on a honeypot back to them. A diagram of this process can be found in Figure 2.

This work hence aims at collecting data supporting the previously mentioned hypothesis. It will furthermore attempt to provide the reader with any conclusions on the modus operandi of SSH bruteforcers.

2 Ethical and Legal Considerations

This research touches various legal and ethical areas. An in-depth discussion would exceed the boundaries of this paper. Hence, only a short evaluation of the most critical problems is provided, including a brief description on how these problems have been addressed during the research.

2.1 Ethical Implications

During the course of this research no actual logins have been performed. All connections were aborted directly after the authentication succeded, but prior to the opening of a session. All subjects have been informed of their participation in this research. After the subjects have been informed, all data that is directly related to a host has been anonymized. The data presented in this research is reduced to sets containing the first 32bit of a salted SHA-512 hash of the IPv4 address, username, password and the timestamp of the connection. This sufficiently protects the privacy of those third parties originally owning the compromised systems.

2.2 Legal Implications

The legal implications of this project can not be fully determined by the author, as it would require a deep legal background and this required legal background would not be limited to one jurisdiction. By now there is nearly no country without at least one online host. This means that nearly all jurisdictions are concerned. Hence the author decided, that all connect-back sessions would be terminated directly after the result of the authentication attempt is returned, right before a session is opened. This way, the systems are never actually accessed, only the credentials previously sent by the target are verified.

3 Connect Back Software

The first step in testing the proposed hypothesis is the development of software that allows the wire-tapping of inbound SSH connection attempts to harvest the credentials and the inbound host. This data then has to be timestamped and recorded. The second step is adding a feature to that software that attempts a connection on the inbound host. The software then has to record the result of that authentication attempt. As previously mentioned, it has to be ensured that no session is opened after the authentication attempt was successful.

Naturally there is no software available which provides the features needed for this experiment. This means that one has to be developed.

The python libary paramiko [15] provides a quick way of implementing client and server services for the SSH protocol in python.

The libary comes with a demo implementation for a simple SSH server. This demo implementation was extended to support the feature set needed for the research project at hand.

3.1 The SSH-CB Software

To allow the reader to reproduce the results discussed later on, a full copy of the python source code for the patched version as well as the vanilla version of the paramiko SSH server demo code have been attached to this document. The vanilla version can be found in Appendix M and the patched code can be found in Appenfix L.

The original paramiko demo code neither supports multiple concurrent connections, nor does it support re-listening after a connection has been dropped. These features were easily implemented by following the python documentation on socket handling [6].

The connect back feature relevant for this research was added after the patching for the previously mentioned base features was done. The first adjustment beyond code re-arrangement can be found in line 99 of the patched code.

The paramiko implementation is configured to present the banner of the OpenSSH server delivered with Ubuntu[1] 12.04 Precise Pangolin in January 2013. This measure has been taken as a pure

[1] `http://www.ubuntu.com/`

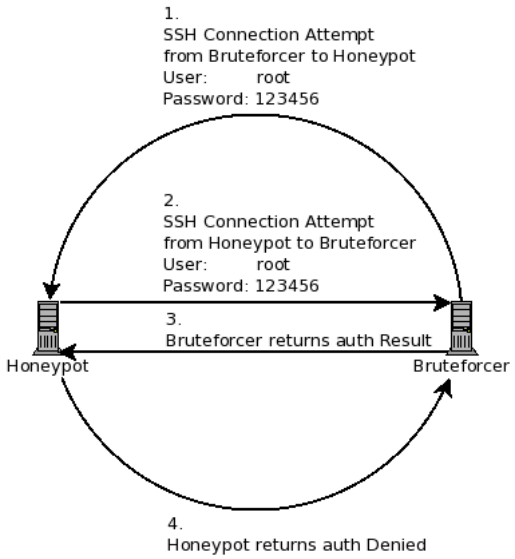

Figure 2: A graphical representation of the proposed technique.

precaution, in case SSH bruteforcers pre-grab the banner of remote systems, for instance to exclude targets that do not look like systems providing a base system suitable for further use like routers, switches or other limited appliances. Research on this is sparse, but at least Kenna [9] suggested that attackers utilize a two-phased scheme in which a list of targets is compiled in the first step and the targets are then bruteforced in a second step.

The second addition can be found in lines 43-48 of the patched code. The server class was extended with a class variable "clientAddr". Its value is set during the instantiation of an object from that class by the constructor. The instantiation can be found in line 106. There the remote address of the socket for that connection is passed as an argument to the constructor of the server class.

The last relevant addition can be found in the "check_auth_password" method of the server between lines 55 and 78 of the patched code. The original method of the parent class is overwritten with a custom authorization function. This custom function executes a connection attempt to the remote host of that connection with the username and password supplied by that host. The "ssh.connect" statement in line 59 of the patched code will throw an exception if the authentication of that connection is not successfull. This is caught by enclos-

ing the whole statement in a try-except block. If the authentication is not successful, an exception is thrown and the data relevant to that connection will be recorded in a file listing failed connect-back attempts by the except block. If no exception is thrown, the authentication attempt was successfull and the try block continues. The relevant data is then stored in a file listing successful connect-back attempts. In both cases the honeypot SSH server returns authorization denied to the client. Relevant data means in both cases the connecting host, the supplied credentials and the date of the connection attempt.

It is important to note that the paramiko SSH implementation specifically requires the code to open a session after the connection has been successfully authenticated [14]. This is not done by the implementation at hand.

As the authorization function is called for each authentication attempt to the honeypot, it is ensured that each connection is processed as described in lines $59ff.$ of the sourcecode in Appendix L.

4 Experimental Design

In order to gather a large sample, two experiments with different settings have been conducted. The first utilized single hosts in different physical and network logical locations, so that probes from various very distinct networks and regions could be taken. A full list on the used hosts can be found in Appendix K. The ssh-cb software was set up to listen on TCP-Port 22, the default SSH-Port[22, p. 3], on each of those systems.

The second approach focused more on measuring distributed attacks, where one wordlist is scattered over several hosts, alternating their pieces of the wordlists over a larger network. For this purpose a set of six /24[2] was requested from RIPE NCC[3]. A copy of the request can be found in Appendix N. Those six networks were supplemented by two /24 contributed by other parties. Documentation on these two networks can be found in Appendix O.

In this case, each /24 was dNATed[4] to a single address, where one instance of the ssh-cb software listened on port 22. That way distinct datasets were created for each /24. The initial target IP in each /24 was not recorded.

5 Results

The results between the two experiments largly varied. Tables 1 and 3 in Appendix K and O provide an overview of the results for both experiments.

5.1 Single Host Results

In the single host experiment, 69,386 connections from 320 different systems were observed. The experiment ran for 299 hours[5] [6].

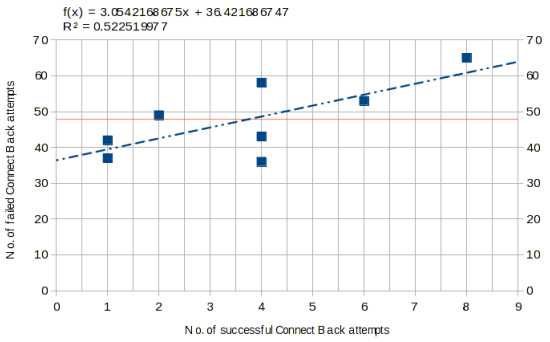

Figure 3: Plot for p2o1 - p2o8, successful vs. failed connect-back attempts.

During this time 29 different sets of username, password and host combinations have been obtained by successfuly connecting back to an attacking node, resulting in an average success rate of 9.375% on all hosts. Connections from single remote hosts have been seen on multiple honeypots. This results in an increased value of 30 non-unique sets of credentials recovered, and 413 non-unique sets of hosts connecting to honeypots.

A correlation between the total amount of inbound hosts and the amount of successful connect back attempts per host seems to exist as shown in Figure 3. The Pearson product-moment correlation coefficient was determined as $\rho_{X,Y} = 0.811$.

[2] CIDR Subnetwork according to RFC4632 [7]

[3] The authority for assigning internet resources within Europe. `http://www.ripe.net/`

[4] According to RFC3022 [18]

[5] The node p2o7 and p2o8 did not, see Appendix K for details.

[6] Time between first and last connection to a honeypot node. Rounded up.

This yields a strong correlation between these two variables. For a qualified statement on a possible causal relation more data would have to be gathered in further research.

A frequency analysis of the collected data supports the previously stated observation of a correlation between the pure number of unique hosts connecting to a honeypot and the rate of successful connect-back attempts. The corresponding histogram can be found in Figure 4.

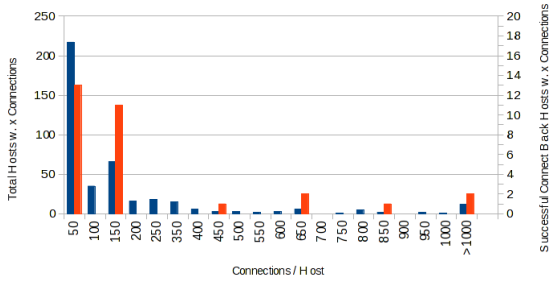

Figure 4: Connections from each inbound host, sorted in classes of stepsize 50, Blue: Amount of hosts. Orange: Amount of successful connect-back attempts

The total amount of successful connect-back attempts per inbound host shows high levels of spiking. Four sections of connection attempts stand out. Of these four only one shows a large amount of different hosts connecting. The group of hosts with 100 to 150 connections shows a high rate of connecting hosts, associated with a high rate of successful connect-back attempts. See Figure 5 for a B-Splined plot of that data.

The creation and comparison of the complements of the set transformed credentials used by these hosts suggest that most hosts in that category use the same wordlist with minor variations. One example for this wordlist can be found in Appendix J.

An interesting aspect of these wordlists can be found in the relatively complex password "7hur@y@t3am$#@!(*(" found in the word list. Sadly, no previous publications on that password could be found. Instead two blog posts turned up, which indicate that there were at least two incidents of remote compromisation by a "Team Thuraya" back in 2009 and 2010 [10, 8].

A further search for passwords in the gathered data, which break the pattern of simple passwords for bruteforce attempts already described by Owens [13] turned up multiple of those passwords. One of those, "spargeosu#^%*&138cucapulinpicior", even accounted for three successful connect-back attempts on different machines. The full list of these passwords can be found in Appendix I. It is assumed by the author that these passwords can be attributed to "groups" running SSH bruteforcing and were leaked to competing "groups".

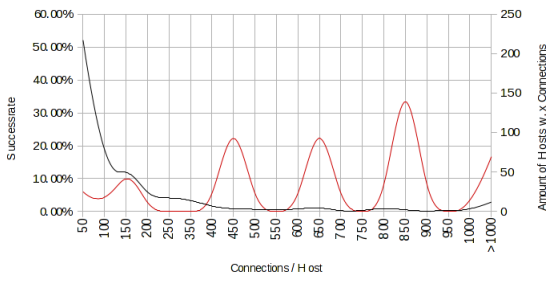

Figure 5: B-Splined plot of successrate vs. amount of hosts per class. Red: Successrate. Black: Amount of hosts per class.

5.2 Multi Network Results

The additionally conducted network-based study produced highly different results. The experiment ran for 333 hours[7]. During that timeframe 632 unique hosts were observed, but only credentials for 36 (5.38%) of these were obtained.

The six /24 networks from mostly consecutive /16 created very similiar results. Not only did they provide a low success rate ranging between 3.81% and 5.76%, they also exhibited a huge spike in the number of hosts connecting per timeslice as show in Figure 6. This effect could also be observed on 195.191.197.0/24. The only network that did not show this effect is 145.100.109.0/24. 145.100.109.0/24 also shows a very high success rate of 15.91%.

A comparison of the average amount of connecting hosts per day between the single host study and the results of the whole network study exposed two spikes in the dataset of the network study. The

[7]Time between first and last connection to a honeypot node. Rounded up.

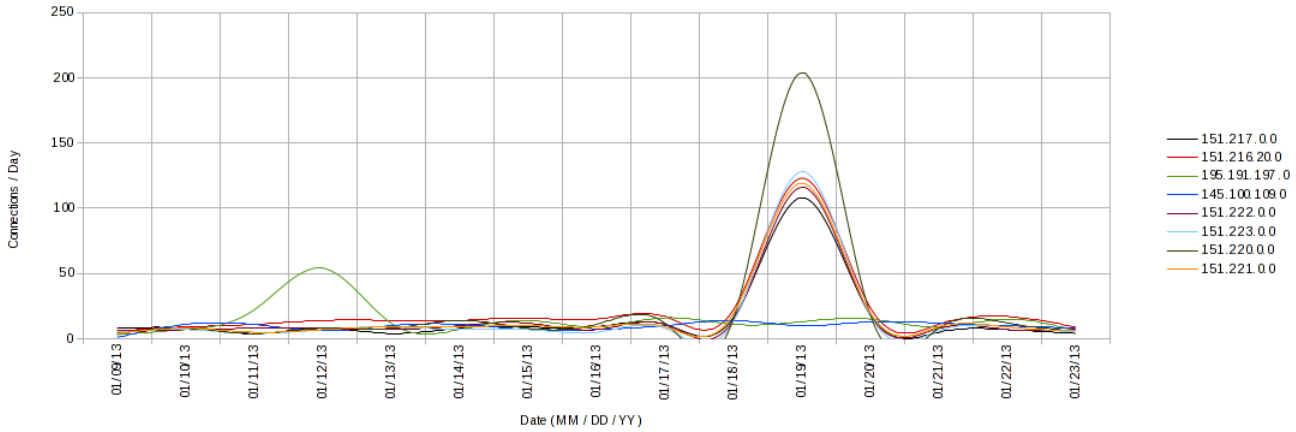

Figure 6: Plot of daily unique hosts connecting for each network during the network study.

six /24 from more or less consecutive /16 showed a spike around 01/19/13, while 195.191.197.0/24 shows a similar spike on 01/12/13, Between 100 and 200 hosts have been observed on 01/19/13 for the six mentioned networks, and 195.191.197.0/24 saw over 50 networks on 01/12/13. As Figure 6 shows, these values largely exceed the average amount of hosts per day observed on other dates.

The gathered data for the multi-network study has been filtered to exclude datapoints for those dates. This leads to the changed results shown in Table 6. The total success rate increased to 11.86%, and the value of total hosts seen decreased to 295, less than half of the unfiltered data. The amount of penetrated hosts however only decreased by one to 35. The full tables for the filtered dataset can be found in Appendix C.

6 Conclusion

The gathered data certainly allows the conclusion that the initial research hypothesis is correct. Connecting back with the same credentials that have been sent by an attacking SSH bruteforce system can lead to a successful penetration of the attacker in a significant number of cases.

A comparison between the data gathered in the single host study and in the whole network study leads to the conclusion that whole networks, especially from the same larger netblock do not promise better results. The outliers detected in the whole network study also suggest the existence of more professional attackers, launching attacks with hundreds of systems at the same time, while each system only attempts a limited set of authorization attempts.

Another side-effect of this study was the detection of various passwords that can be attributed to so far unidentified groups involved with SSH bruteforce operations. The existence of those passwords in wordlists allows the conclusion of the existence of multiple, independently operating groups. It also explains why the theory proposed for this paper holds up against the claims of Ramsbrock, Berthier and Cukier mentioned earlier [16]. The changed passwords leaked to other groups, eventually ending up in those groups wordlists. Those competing groups then penetrate the same systems previously penetrated by the first group, possibly on a different account, start SSH bruteforcing from that account as well, and thereby expose the password of the initially compromised account.

7 Further Work

Although providing various new insights into the world of SSH bruteforcers, the results of this study allow for more future research objectives than conclusions. Various aspects of the proposed technique require further research.

7.1 Generalisation of the Method

The proposed method is currently focused on a single attack vector. It may be possible to extend it to other exploitation techniques. This could include other means of remote access e.g., the common RDP protocol [3] but also services for protocols that are not necessarily related to authorizing remote access to a system like HTTP [5].

7.2 Ethical and Legal Challenges

The proposed technique allows not only the gathering of credentials for compromised systems. It would also be possible to use the credentials to clean up the infected systems and gather more information on the modus operandi of SSH bruteforcing groups.

This paper does not take the ethical and legal implications that arise from the availability of this technique into account. Although the legal implications may be left aside, if this technique is used by a government organisation to actively reduce malicious actions on the internet, the author of this paper already claimed in 2012, that the use of unauthorized remote access for remote forensic purposes by the authorities is not acceptable [4].

That work however did not take cases into account, where the authorities are restricted in the way they may use information gathered on those systems. If the use of data and information of any legitimate user in a criminal investigation or court of law would be prohibited following an idea similiar to the "fruit of the poisonous tree" doctrin in the United States and the individuals executing the procedure are bound to a secrecy agreement similiar to "doctor-patient confidentiality", the final conclusion on the ethical feasability may differ.

The author intends to follow up on these thoughts in future publications.

7.3 Further Analysis of Gathered Data

The data that has been obtained during this study will be anonymized and published at `http://sshcb.wybt.net/`. Further analysis of this data is advised, especially if such an analysis would focus on other aspects of the obtained wordlists.

Acknowledgments

Pieter Lexis - Told me to stop talking and rather test the theory.
Dr. Hans Dijkman - Gave huge support in solving the ethical and legal issues of this work.
Nadine Donaldson, BSc - Gave helpful advise on the data analysis.
Kay Rechthien - Assisted in setting up resources and networks.
Stefan Wahl - Supported the project by providing LIR services for the RIPE networks.
Niels Sijm, MSc - Assisted in setting up resources and networks.
Theodor Reppe - Provided systems for the single host study.
Elmo Todurov - Who independently came up with the same theory during the finalization of this research [20].

References

[1] D. Barrett, R. Silverman, and R. Byrnes. *SSH, The Secure Shell: The Definitive Guide: The Definitive Guide*. O'Reilly Media, 2011.

[2] Jose Antonio Coret. Kojoney - A honeypot for the SSH Service. http://kojoney.sourceforge.net/, Fri Feb 1 16:16:33 CET 2013, 2006.

[3] Microsoft Corp. Understanding the Remote Desktop Protocol (RDP). http://support.microsoft.com/kb/186607/en-us, Fri Feb 1 17:14:46 CET 2013, 2007.

[4] T. Fiebig. Ethical implications of remote forensic software in the context of the extended mind theory. BSc Thesis, University of Osnabrück, 2012.

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785, 6266, 6585.

[6] Python Software Foundation. 17.2. socket — Low-level networking interface. http://docs.python.org/2.6/library/socket.html, Fri Feb 1 16:48:04 CET 2013, 2012.

[7] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632 (Best Current Practice), August 2006.

[8] "jcombs_31". Forum Post. http://www.howtoforge.com/forums/showthread.php?t=48956, Fri Feb 1 17:10:20 CET 2013, 2010.

[9] C. Kenna. Analysis of and response to ssh brute force attacks. The College of William & Mary, 2010.

[10] "Matthew". Blog Post. http://project-2501.net/index.php/2009/09/hacked/, Fri Feb 1 17:09:43 CET 2013, 2009.

[11] J. Mirkovic and P. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[12] R. Morris and K. Thompson. Password security: A case history. *Communications of the ACM*, 22(11):594–597, 1979.

[13] J.P. Owens Jr. A study of passwords and methods used in brute-force ssh attacks. MSc Thesis, Clarkson University, 2008.

[14] Robey Pointer. paramiko 1.7.7.1 API Documentation - Package paramiko :: Class SSHClient. http://www.lag.net/paramiko/docs/paramiko.SSHClient-class.html, Fri Feb 1 16:54:33 CET 2013, 2011.

[15] Robey Pointer. paramiko 1.7.7.1 "George". http://www.lag.net/paramiko/, Fri Feb 1 16:47:17 CET 2013, 2011.

[16] D. Ramsbrock, R. Berthier, and M. Cukier. Profiling attacker behavior following ssh compromises. In *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on*, pages 119–124. IEEE, 2007.

[17] G. Salles-Loustau, R. Berthier, E. Collange, B. Sobesto, and M. Cukier. Characterizing attackers and attacks: An empirical study. In *Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on*, pages 174–183. IEEE, 2011.

[18] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022 (Informational), January 2001.

[19] Upi Tamminen. Kippo - SSH Honeypot. http://code.google.com/p/kippo/, Fri Feb 1 16:19:46 CET 2013, 2009.

[20] Elmo Todurov. A stupidly easy way to hack into computers. http://theorylunch.wordpress.com/2013/01/24/ssh-mitm/, Sun Feb 10 23:46:37 CET 2013, 2013.

[21] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC 4251 (Proposed Standard), January 2006.

[22] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. RFC 4253 (Proposed Standard), January 2006. Updated by RFC 6668.

A Data Summary Single Host Study

A.1 Base Properties

Host	Avg. Connections/h	Max Connections/h	Total Connections
All	232.06	3063	69386
p2o1	26.96	1136	8062
p2o2	18.46	746	5519
p2o3	24.97	1219	7467
p2o4	19.68	645	5886
p2o5	25.81	793	7716
p2o6	41.40	1560	12379
p2o7	35.11	717	10497
p2o8	39.67	3042	11860

Table 1: Base Data for Single Host Study, runtime 299 hours

A.2 Success / Fail Rate

Host	Penetrated Hosts	Non Penetrated Hosts	Successrate
All	30	290	9.38%
p2o1	2	49	3.92%
p2o2	8	65	10.96%
p2o3	1	42	2.33%
p2o4	1	37	2.63%
p2o5	4	43	8.51%
p2o6	6	53	10.17%
p2o7	4	58	6.45%
p2o8	4	36	10.00%

Table 2: Success Rate for Single Host Study

B Data Summary Network Study

B.1 Base Properties

Net	Avg. Connections/h	Max Connections/h	Total Connections
All	1993.72	33027	663912
145.100.109.0/24	668.87	25202	222736
151.216.20.0/24	182.19	3598	60670
151.217.0.0/24	173.47	8294	57767
151.220.0.0/24	211.29	8186	70361
151.221.0.0/24	192.38	8218	64064
151.222.0.0/24	175.58	3740	58470
151.223.0.0/24	196.59	8296	65466
195.191.197.0/24	193.32	3468	64378

Table 3: Base Data for Network Study, runtime 333 hours

B.2 Success / Fail Rate

Net	Penetrated Hosts	Non Penetrated Hosts	Successrate
All	36	632	5.38%
145.100.109.0/24	14	74	15.91%
151.216.20.0/24	13	257	4.81%
151.217.0.0/24	11	180	5.76%
151.220.0.0/24	12	287	4.01%
151.221.0.0/24	8	202	3.81%
151.222.0.0/24	9	193	4.46%
151.223.0.0/24	8	201	3.83%
195.191.197.0/24	4	158	2.47%

Table 4: Success Rate for Network Study

C Data Summary Network Study - Filtered

C.1 Base Properties

Net	Avg. Connections/h	Max Connections/h	Total Connections
All	1732.44	33027	576901
145.100.109.0/24	668.88	25202	222736
151.216.20.0/24	140.88	3598	46913
151.217.0.0/24	136.90	8294	45587
151.220.0.0/24	176.31	8186	58710
151.221.0.0/24	161.26	8218	53698
151.222.0.0/24	135.40	3696	45089
151.223.0.0/24	156.77	8296	52204
195.191.197.0/24	156.05	3468	51964

Table 5: Base Data for Network Study, runtime 333 hours - outliers filtered

C.2 Success / Fail Rate

Net	Penetrated Hosts	Non Penetrated Hosts	Successrate
All	35	260	11.86%
145.100.109.0/24	14	74	15.91%
151.216.20.0/24	12	148	7.50%
151.217.0.0/24	10	83	10.75%
151.220.0.0/24	11	93	10.58%
151.221.0.0/24	7	93	7.00%
151.222.0.0/24	8	89	8.25%
151.223.0.0/24	7	85	7.61%
195.191.197.0/24	4	113	3.42%

Table 6: Success Rate for Network Study - outliers filtered

D Graph: Hosts per Day Single Host Study

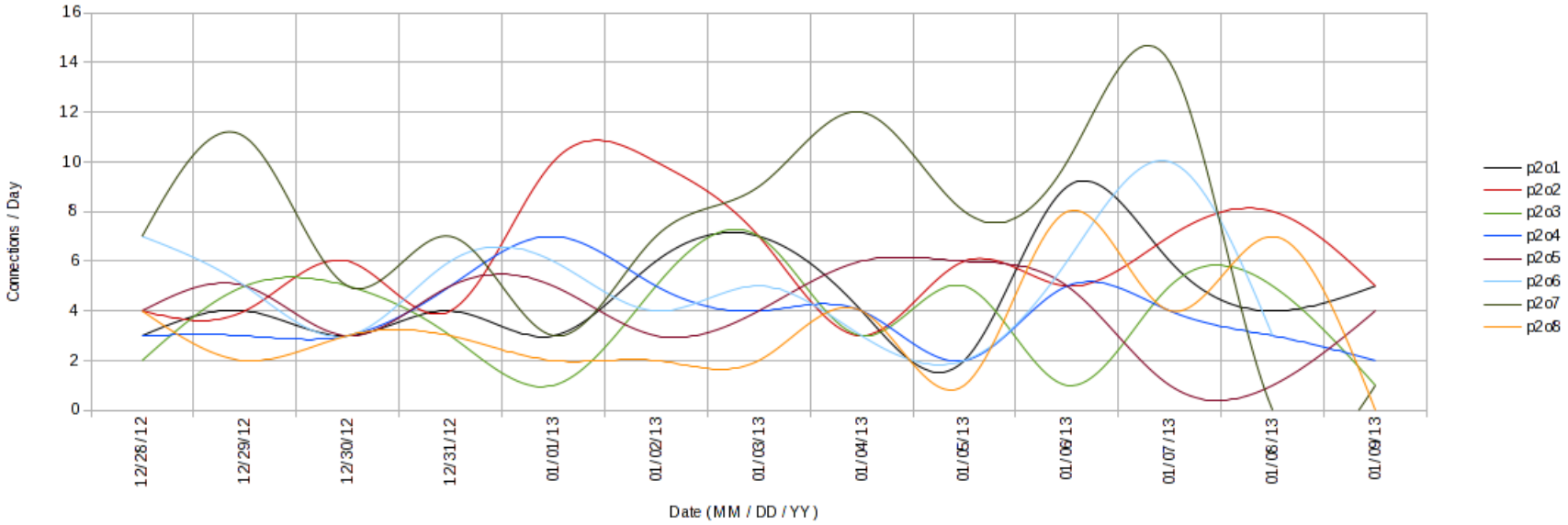

Figure 7: Plot of daily unique hosts connecting for each honeypot during the single host study.

E Graph: Hosts per Day Network Study

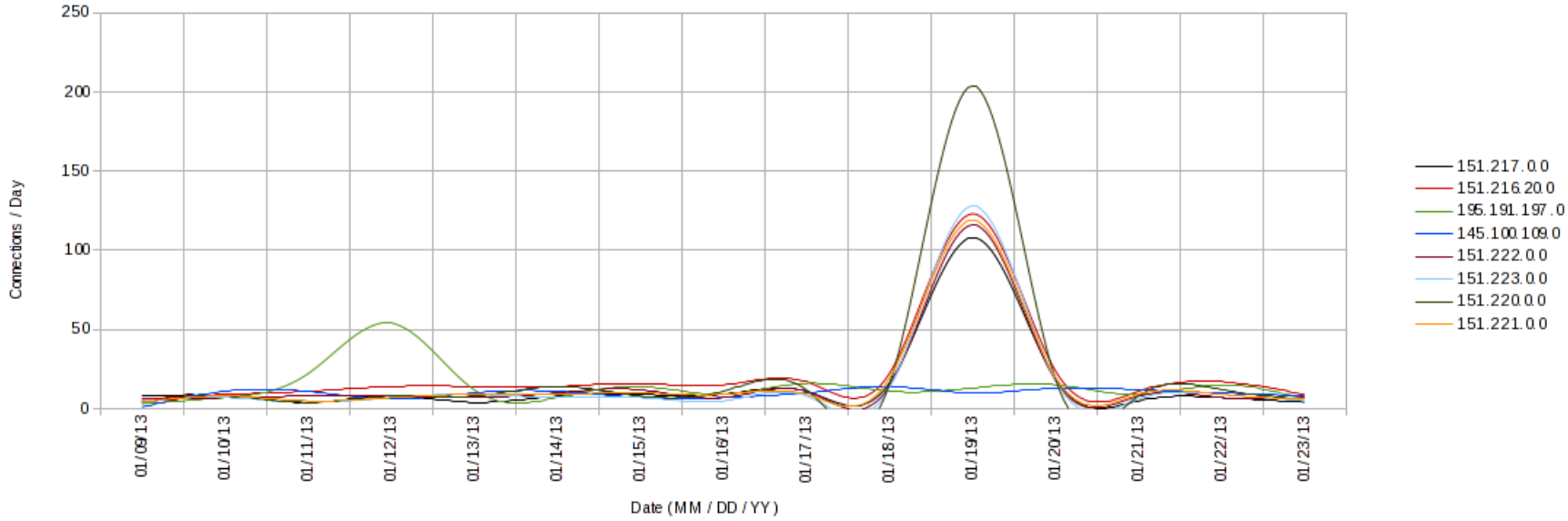

Figure 8: Plot of daily unique hosts connecting for each network during the network study.

F Graphs: Single Host Successrate Graphs

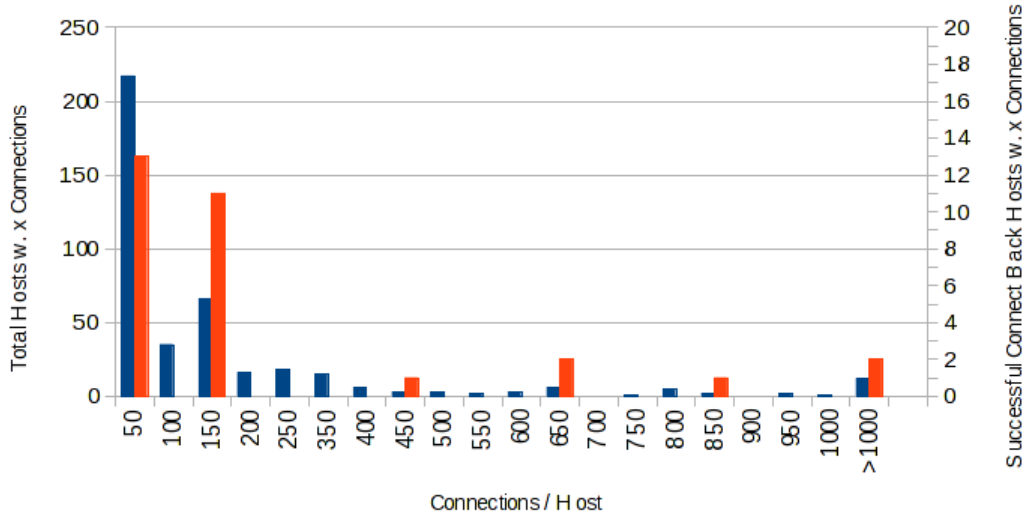

Figure 9: Connections from each inbound host, sorted in classes of stepsize 50, Blue: Amount of hosts. Orange: Amount of successful Connect Back attempts

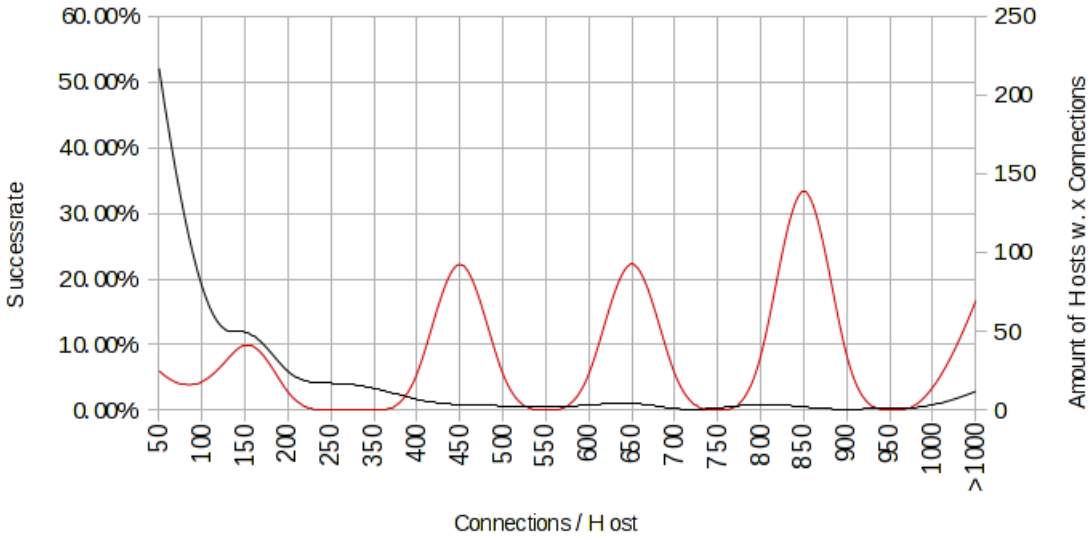

Figure 10: B-Splined plot of successrate vs. amount of hosts per class. Red: Successrate. Black: Amount of hosts per class.

G Graphs: Network Successrate Graphs

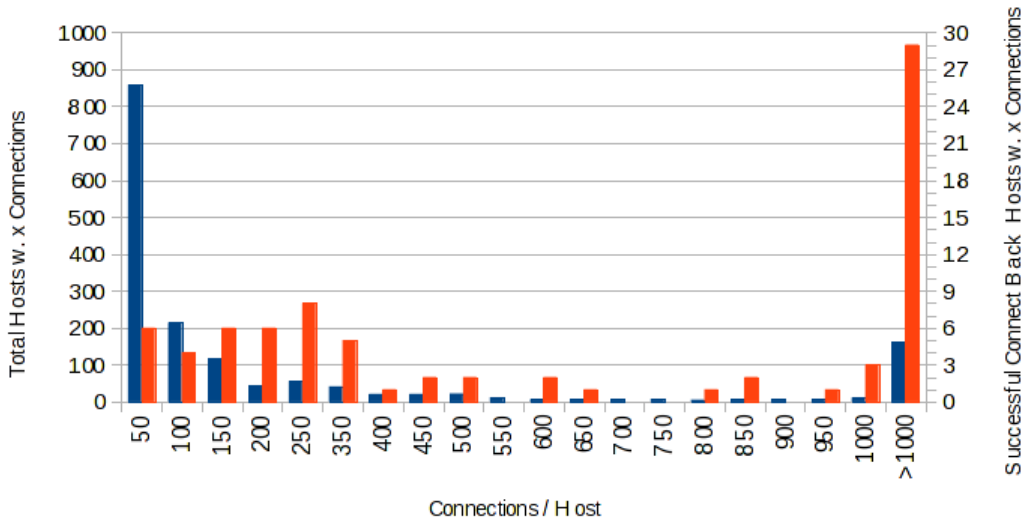

Figure 11: Connections from each inbound host, sorted in classes of stepsize 50, Blue: Amount of hosts. Orange: Amount of successful Connect Back attempts

Figure 12: B-Splined plot of successrate vs. amount of hosts per class. Red: Successrate. Black: Amount of hosts per class.

H Graph: Successfull vs. Failed Connect Back Attempts Single Host Study

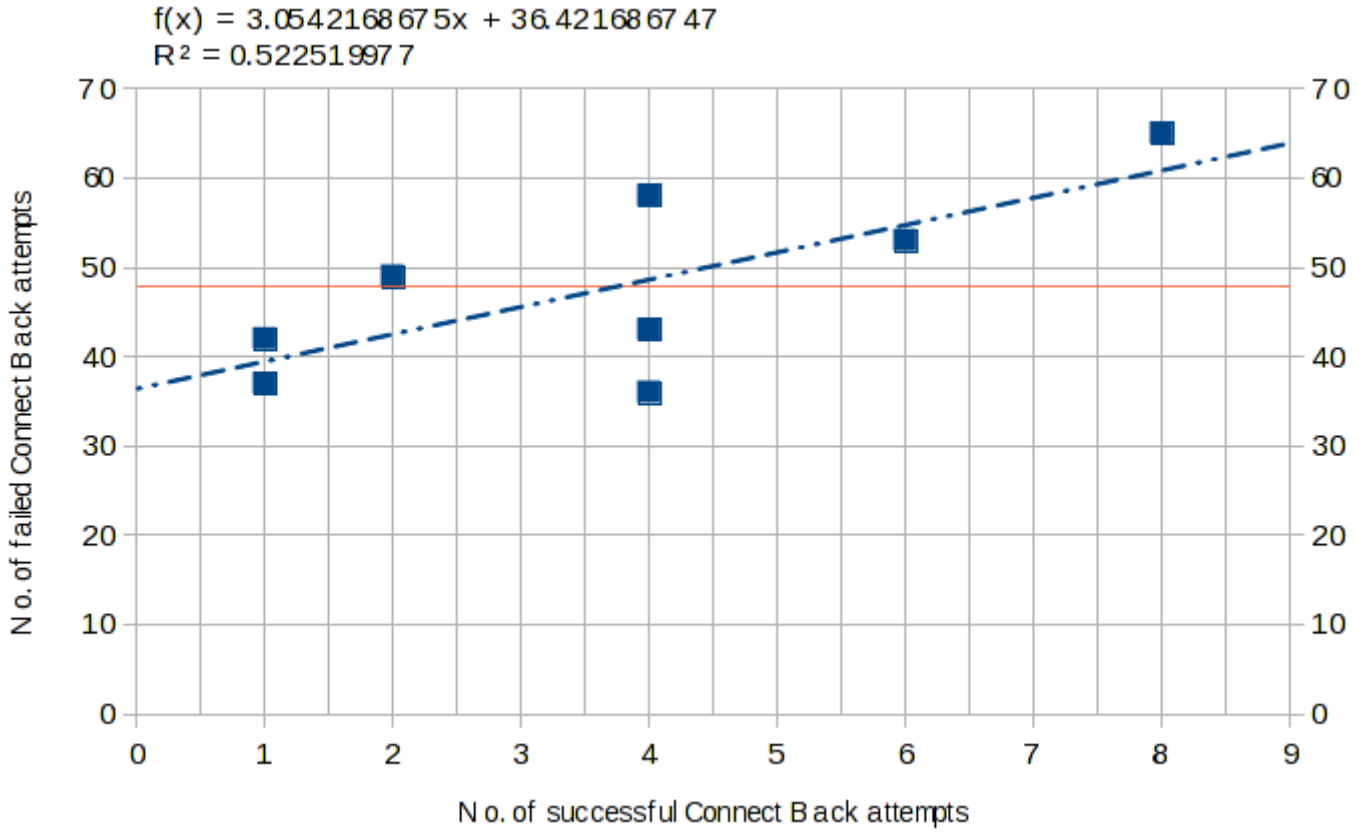

Figure 13: Plot for p2o1 - p2o8, successful vs. failed connect-back attempts.

I Possible Group Passwords

```
khaled-dico-ana-wla-akhou-charmouta-tfeh-kess-ekhtak-bi-ayri-a5ou-a7beh
ckwS2nrN&&0(x=;1E}2l=}8*9bfGSz6kVx7lLKm!LID5]nu8hW<QN)^nbX'K
ortega.123#TradeLinuxKi!l|iN6#Th3Ph03$%nix@NdR3b!irD
123parola321esniffu321$#@!nuirootutaudeateuita#@!@#$
deathfromromaniansecurityteamneversleepba
vreau.sa.urc.255.de.emechi.pe.undernet
efwef58sdf2cvsd1*!#&$#_)claudia69iLiE
youhaveabubasuckmypula!x*#!$@*O(221!
[www.cinenustieparolasugepula.biz%5d
Fum4tulP0@t3Uc1d3R4uD3T0t!@#$%^%^&*?
NKtfgCjQRr9TtjfRPmJdIINGOODWETRUST
dragos3443gff@665$G455454dragos2sd
$3cr3t#%%DiafstigmaNumelemeumic%%#
UIYORYIPRTEWFDJDHGKJRRTEWEGSDFHFS
@!#$%&*Th3@#$!F0RcE%&*@#IS!@#$%!&
$3NH4#%%DiafstigmaNumelemeumic%%#
w7aThexApruP3asWQ8kURa9rphe8rEpR
!#$%&*Th3@#$!F0RcE%&*@#IS!@#$%!&
spargeosu#^%*&138cucapulinpicior
SK!587eN9a@Y61e3iOG63!Nsv81E7hL4
nobodywasherexXXx012132*8ushd8ss
@n!md@mP#$@&#3141$&#@!#mTadm!n$@
f41rwayfds^&789fdsa%^*&fds@!#@$%
------Brz-O-Baga-n-Mata----------
ana.este.o.dulceata.de.fata.2011
@#$WEFHERI(*FQR23587tfwAGBFUIDF
Ki!l|iN6#Th3Ph03$%nix@NdR3b!irD
Sugq1w2e4^^1qzarolaMeaDeLaSSHD
!@*(@HBsd8H!@#&@EDBAS*@B#!(BD
$3cr3t#mafiavafute197532@%!?*
Rh3I5Lik3P4rtY@@@v3rmagnnumm
#hackm3baby#logrono1#cancel#
@#$%hackin2inf3ctsiprepe@#$%
biMNC.!@#$^AdelFedora24.+_}P
tr4yt0d1sc0aarm4ype4as5w0rP
@n!md@mP#$@?$&#@!#mTadm!n$@
L@pt0pF1nLuXuS33baie22dus?!
$3cr3t!Q@W#E$R%T^Y&U*I(O)P_
0wn3d-6BD1714F.dedicated.tu
ZUH4LT3R_FUCK_YOU_ZUH4LT3R
p0w3rOF//Rullers@L$%L$%-00
h5a2n4d7a9o1l$%i*()an(&*g)
7hur@y@t3am$#@!(*(
```

J Example Wordlist

User	Password	User	Password
root	P@ssw0rd	----	----
root	----	root	nokia123
root	12345	root	p@$$w0rd
root	1234qwer!@#$	root	12qwaszx
root	michael	root	Pa$$word
root	asdasd	root	sebica1234
root	P@$$w0rd	root	qwer!@#$
root	888888	root	..-55
root	7hur@y@t3am$#@!(*(	root	redroot
root	asdf1234	root	123qweasd
root	123654	root	theking
root	!@#$QWER	root	!qazxsw@
root	power	root	samsung
root	sysadmin	root	test123
root	1q2w3e4r	root	r00t
root	1qaz@WSX	root	abc123!@#
root	qwerasdf	root	silver
root	Passw0rd	root	access
root	password1	root	testbox
root	kagome	root	linux123
root	123$%^789	root	maverick
root	123456789	root	x
root	prueba	root	wvhlyf
root	Password1!	root	id
root	sunshine	root	123qaz
root	asshole	root	blahblah
root	123456!@#$%^	root	testing
root	justdoit	root	11111111
root	p4ssw0rd	root	123!@#
root	1	root	1z2x3c4v
root	xxxxxx	root	asdf123
root	viper	root	super
root	123	root	dzpyerg9
root	Pa$$w0rd	root	compaq
root	zaq123edc	root	secret
root	milan	root	jordan23
root	foobar	root	q1w2e3r4t5y6
root	1q2w3e	root	servidor
root	support	root	4dm1n
root	q1w2e3r4	root	cisco
root	Password123	root	junior
root	testing123	root	zaq12wsx
root	2wsx3edc	root	server123
root	bagabu	root	whatever
root	fuckyou	root	cisco123
root	123qwe123qwe	root	joshua
root	temp123	root	7ujm8ik,
root	Password1	root	sw0rdf1sh
root	branburica	root	qwe123qwe123
root	alex	root	toto
root	1234567	root	Password
root	internet	root	852963
root	stephen	root	football
root	qwerty!	root	qwertyui
root	abc123!	root	vps123
root	buster	root	acer
root	monkey	root	fuck
root	passw0rd	root	qwert123
root	P@ssw0rd!	root	qwert
root	1qa2ws3ed	root	0571749e2ac330a7455809c6b0e7af90
root	111111	root	startrek
root	dolphin	root	zxc!@#
root	pingpong	root	qwerty12
root	qwerty123	root	asdfasdf
root	qwerty	root	a
root	felix	root	danny
root	control	root	pokemon
root	motorola	root	11

K Used Honeypot Systems

```
rDNS: <REDACTED FOR PRIVACY CONCERNS>
IPv4: <REDACTED FOR PRIVACY CONCERNS>
Location: DE, AS24940, Hetzner Online AG
Data-Reference: p2o1

rDNS: <REDACTED FOR PRIVACY CONCERNS>
IPv4: <REDACTED FOR PRIVACY CONCERNS>
Location: DE, AS35366, ISPpro Internet KG
Data-Reference: p2o2

rDNS: <REDACTED FOR PRIVACY CONCERNS>
IPv4: <REDACTED FOR PRIVACY CONCERNS>
Location: US, Phoenix, AS20454, Dolorem Ipsum, s.r.o.
Data-Reference: p2o3

rDNS: <REDACTED FOR PRIVACY CONCERNS>
IPv4: <REDACTED FOR PRIVACY CONCERNS>
Location: US, Dallas, AS36351, Dolorem Ipsum, s.r.o.
Data-Reference: p2o4

rDNS: vps.node71.nqhost.com
IPv4: 109.68.191.166
Location: RU, AS52201, Dolorem Ipsum, s.r.o.
Data-Reference: p2o5

rDNS: test.wybt.net
IPv4: 195.191.196.2
Location: DE, AS31078, WYBT-NET
Data-Reference: p2o6

rDNS: euve8465.vserver.de
IPv4: 62.75.139.144
Location: DE, AS8972, PlusServer AG
Data-Reference: p2o7
Remarks: Down on 01/08/13 due to powerfailure. Exact downtime not measured, assumed to be
         around 12h. The node is not excluded, downtime is considered noise.

rDNS: hull.practicum.os3.nl
IPv4: 145.100.104.167
Location: NL, AS1103, UvA-Master-SNE-NET
Data-Reference: p2o8
Remarks: Down on 01/09/13 due to powerfailure. Exact downtime not measured, assumed to be
         around 8h. The node is not excluded, downtime is considered noise.
```

L Sourcecode: ssh-cb.py

```
#!/usr/bin/env python
# This code is loosely based on the paramiko dem ssh-server. A copy
# of that demo server can be found at:
# http://mcs.une.edu.au/doc/python-paramiko-1.7.7.1/demos/demo_server.py
# and in Appendix B of this document. All differences between that file and
# this file have been created by Tobias Fiebig.
#
# Copyright (C) 2012-2013 Tobias Fiebig <tobias.fiebig@os3.nl>
# Copyright (C) 2003-2007  Robey Pointer <robeypointer@gmail.com>
#
# This is free software; you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation; either version 2.1 of the License, or (at your option)
# any later version.
#
# This software is distrubuted in the hope that it will be useful, but WITHOUT ANY
# WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
# A PARTICULAR PURPOSE.  See the GNU Lesser General Public License for more
# details.
#
# You should have received a copy of the GNU Lesser General Public License
```

```
22 # along with this software; if not, write to the Free Software Foundation, Inc.,
23 # 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA.
24
25 from binascii import hexlify
26 import os
27 import socket
28 import sys
29 import threading
30 import traceback
31 import datetime
32 import paramiko
33 import threading
34 import signal
35
36
37 # host-key used
38 host_key = paramiko.RSAKey(filename='test_rsa.key')
39 paramiko.util.log_to_file('demo_server.log')
40
41 class Server (paramiko.ServerInterface):
42
43     clientAddr = "";
44
45     def __init__(self, client):
46         self.event = threading.Event()
47         self.clientAddr = client
48         print client[0]
49
50     def check_channel_request(self, kind, chanid):
51         if kind == 'session':
52             return paramiko.OPEN_SUCCEEDED
53         return paramiko.OPEN_FAILED_ADMINISTRATIVELY_PROHIBITED
54
55     def check_auth_password(self, username, password):
56         try:
57             ssh = paramiko.SSHClient()
58             ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
59             ssh.connect(self.clientAddr[0], 22, username, password)
60             date = str(datetime.datetime.now())
61             f_log = open("./userdata-success" , "a+")
62             f_log.write("Host: "+self.clientAddr[0]+"\n")
63             f_log.write("Username: "+username+"\n")
64             f_log.write("Password: "+password+"\n")
65             f_log.write("Date: "+date+"\n")
66             f_log.write("---------\n")
67             f_log.close()
68             return paramiko.AUTH_FAILED
69         except:
70             date = str(datetime.datetime.now())
71             f_log = open("./userdata-fail" , "a+")
72             f_log.write("Host: "+self.clientAddr[0]+"\n")
73             f_log.write("Username: "+username+"\n")
74             f_log.write("Password: "+password+"\n")
75             f_log.write("Date: "+date+"\n")
76             f_log.write("---------\n")
77             f_log.close()
78             return paramiko.AUTH_FAILED
79
80     def get_allowed_auths(self, username):
81         return 'password'
82
83     def check_channel_shell_request(self, channel):
84         self.event.set()
85         return True
86
87     def check_channel_pty_request(self, channel, term, width,
88                                   height, pixelwidth, pixelheight, modes):
89         return True
90
91 class RequestHandler(threading.Thread):
92     def __init__(self, (sock, addr)):
93         self.sock = sock
94         self.addr = addr
95         threading.Thread.__init__(self)
```

```
96
97      def run(self):
98          try:
99              t = paramiko.Transport(self.sock)
100             t.local_version = "SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1"
101             try:
102                 t.load_server_moduli()
103             except:
104                 print '(Failed to load moduli -- gex will be unsupported.)'
105                 raise
106             t.add_server_key(host_key)
107             server = Server(self.addr)
108             try:
109                 t.start_server(server=server)
110             except:
111                 print '*** SSH negotiation failed.'
112
113             chan = t.accept(20)
114
115             if chan is None:
116                 i = 1;
117             else:
118                 chan.close()
119
120         except Exception, e:
121             print '*** Caught exception: ' + str(e.__class__) + ': ' + str(e)
122             traceback.print_exc()
123             try:
124                 t.close()
125             except:
126                 print "Exception caught"
127
128 def bind_local():
129     try:
130         sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
131         sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
132         sock.bind(('', 2200))
133         sock.listen(10)
134     except Exception, e:
135         print '*** Bind failed: ' + str(e)
136         traceback.print_exc()
137         sys.exit(1)
138
139     return sock
140
141 def listen_sock(sock):
142     try:
143         sa = sock.accept()
144     except Exception, e:
145         print '*** Listen/accept failed: ' + str(e)
146         traceback.print_exc()
147     return sa
148
149 def cleanup(*args):
150     sys.exit(1)
151 #
152 def main(argv):
153     sock = bind_local()
154     threads = []
155     signal.signal(signal.SIGINT, cleanup)
156     signal.signal(signal.SIGTERM, cleanup)
157     while("true"):
158
159         rh = RequestHandler(listen_sock(sock))
160         rh.daemon = True
161         rh.start()
162         threads.append(rh)
163
164 if __name__ == "__main__":
165     main(sys.argv[1:])
```

M Sourcecode: doc/python-paramiko-1.7.7.1/demos/demo_server.py

```
1  # Copyright (C) 2003-2007  Robey Pointer <robeypointer@gmail.com>
2  #
3  # This file is part of paramiko.
4  #
5  # Paramiko is free software; you can redistribute it and/or modify it under the
6  # terms of the GNU Lesser General Public License as published by the Free
7  # Software Foundation; either version 2.1 of the License, or (at your option)
8  # any later version.
9  #
10 # Paramiko is distrubuted in the hope that it will be useful, but WITHOUT ANY
11 # WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
12 # A PARTICULAR PURPOSE.  See the GNU Lesser General Public License for more
13 # details.
14 #
15 # You should have received a copy of the GNU Lesser General Public License
16 # along with Paramiko; if not, write to the Free Software Foundation, Inc.,
17 # 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA.
18
19 import base64
20 from binascii import hexlify
21 import os
22 import socket
23 import sys
24 import threading
25 import traceback
26
27 import paramiko
28
29
30 # setup logging
31 paramiko.util.log_to_file('demo_server.log')
32
33 host_key = paramiko.RSAKey(filename='test_rsa.key')
34 #host_key = paramiko.DSSKey(filename='test_dss.key')
35
36 print 'Read key: ' + hexlify(host_key.get_fingerprint())
37
38
39 class Server (paramiko.ServerInterface):
40     # 'data' is the output of base64.encodestring(str(key))
41     # (using the "user_rsa_key" files)
42     data = 'AAAAB3NzaC1yc2EAAAABIwAAAIEAyO4it3fHlmGZWJaGrfeHOVY7RWO3P9M7hp' + \
43            'fAu7jJ2d7eothvfeuoRFtJwhUmZDluRdFyhFY/hFAh76PJKGAusIqIQKlkJxMC' + \
44            'KDqIexkgHAfID/6mqvmnSJf0b5W8v5h2pI/stOSwTQ+pxVhwJ9ctYDhRSlF0iT' + \
45            'UWT10hcuO4Ks8='
46     good_pub_key = paramiko.RSAKey(data=base64.decodestring(data))
47
48     def __init__(self):
49         self.event = threading.Event()
50
51     def check_channel_request(self, kind, chanid):
52         if kind == 'session':
53             return paramiko.OPEN_SUCCEEDED
54         return paramiko.OPEN_FAILED_ADMINISTRATIVELY_PROHIBITED
55
56     def check_auth_password(self, username, password):
57         if (username == 'robey') and (password == 'foo'):
58             return paramiko.AUTH_SUCCESSFUL
59         return paramiko.AUTH_FAILED
60
61     def check_auth_publickey(self, username, key):
62         print 'Auth attempt with key: ' + hexlify(key.get_fingerprint())
63         if (username == 'robey') and (key == self.good_pub_key):
64             return paramiko.AUTH_SUCCESSFUL
65         return paramiko.AUTH_FAILED
66
67     def get_allowed_auths(self, username):
68         return 'password,publickey'
69
70     def check_channel_shell_request(self, channel):
71         self.event.set()
```

```
72          return True
73
74      def check_channel_pty_request(self, channel, term, width, height, pixelwidth,
75                                    pixelheight, modes):
76          return True
77
78
79  # now connect
80  try:
81      sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
82      sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
83      sock.bind(('', 2200))
84  except Exception, e:
85      print '*** Bind failed: ' + str(e)
86      traceback.print_exc()
87      sys.exit(1)
88
89  try:
90      sock.listen(100)
91      print 'Listening for connection ...'
92      client, addr = sock.accept()
93  except Exception, e:
94      print '*** Listen/accept failed: ' + str(e)
95      traceback.print_exc()
96      sys.exit(1)
97
98  print 'Got a connection!'
99
100 try:
101     t = paramiko.Transport(client)
102     try:
103         t.load_server_moduli()
104     except:
105         print '(Failed to load moduli -- gex will be unsupported.)'
106         raise
107     t.add_server_key(host_key)
108     server = Server()
109     try:
110         t.start_server(server=server)
111     except paramiko.SSHException, x:
112         print '*** SSH negotiation failed.'
113         sys.exit(1)
114
115     # wait for auth
116     chan = t.accept(20)
117     if chan is None:
118         print '*** No channel.'
119         sys.exit(1)
120     print 'Authenticated!'
121
122     server.event.wait(10)
123     if not server.event.isSet():
124         print '*** Client never asked for a shell.'
125         sys.exit(1)
126
127     chan.send('\r\n\r\nWelcome to my dorky little BBS!\r\n\r\n')
128     chan.send('We are on fire all the time!  Hooray!  Candy corn for everyone!\r\n')
129     chan.send('Happy birthday to Robot Dave!\r\n\r\n')
130     chan.send('Username: ')
131     f = chan.makefile('rU')
132     username = f.readline().strip('\r\n')
133     chan.send('\r\nI don\'t like you, ' + username + '.\r\n')
134     chan.close()
135
136 except Exception, e:
137     print '*** Caught exception: ' + str(e.__class__) + ': ' + str(e)
138     traceback.print_exc()
139     try:
140         t.close()
141     except:
142         pass
143     sys.exit(1)
```

N Application for RIPE-NCC Provided Networks

```
% Temporary Internet Number Assignment Request Form
% RIPE NCC members (LIRs) can use this form to request
% a Temporary Internet Assignment. Please see "Supporting Notes for the Temporary
% Internet Assignment Request Form" for instructions on how to complete this form.
% http://ripe.net/ripe/docs/temp-assign-support
%
% Please note that an End User should have a signed "Temporary Independent
% Assignment Request and Maintenance Agreement" with a sponsoring LIR.
% http://ripe.net/lir-services/resource-management/temp-assign-agreement

#[GENERAL INFORMATION]#
% Please add your RegID.

request-type: temp-assign
form-version: 1.0
x-ncc-regid:

#[ASSIGNMENT USER]#
% Who will use the requested assignment?
legal-organisation-name: Tobias Fiebig
organisation-location: Natrupper Str. 98, D-49090 Osnabrueck, GERMANY
website-if-available: https://www.os3.nl/

% Is this request being sent by a sponsoring LIR on behalf of
% an End User? (yes/no)

end-user-of-sponsoring-lir: yes

% If yes, please confirm that the "Temporary Independent Assignment Request and
% Maintenance Agreement" contains all of the elements listed in paragraph 2.0 of
% "Contractual Requirements for Provider Independent Resource Holders in the
% RIPE NCC Service Region".(yes/no)
% Please also attach a copy of the signed agreement and the company registration
% papers of the End User.

confirmation: yes

#[INITIAL INFORMATION]#

% Which type of assignment is the End User requesting? (IPv4/IPv6/ASN)


type-of-assignment: IPv4

% Why do you need this temporary assignment?

why: Research Project

% The End User should be aware that this resource will be for a specific time
% period and will be automatically de-registered at the end of the approved
% assignment period.
% Please add more information on the purpose (Event/Research) and duration of this
% request.

purpose: The University van Amsterdam accepted the attached research proposal.
During the course of this research it became apparent, that the results of the
experiment do not reach those of a pre-evaluation. This pre-evaluation was done
with one /24 DNATed to one host, while the currently active evaluations utilizes
single hosts with a single /32. This resulted in a new hypothesis, claiming that
the performance of the devised method can be increased, if a whole /24 is used
for honeypot purposes instead of only one /32. In order to retrieve a wide spread
data-basis, i.e. gather data from different ssh bruteforce systems, usually
harvesting on a single /16 at a time, multiple /24 from multiple /16 are needed.
The use of six different /24 is a design decission, which keeps the limited amount
of left IPv4 resources in mind, while still providing a reasonable sample size in
comparisson to the single host study which utilizes eight different /32.

website-if-available: http://rp.delaat.net/2012-2013/index.html (#22)

% The date should be in the following format: yyyymmdd

```

```
start-date:20120107
end-date:20120128

% The next three sections (IPv4, IPv6 and ASN) will give us an overview of the
% detailed usage of the resources. Please fill in only the relevant
% sections as per the resource being requested and remove the sections that are not
% applicable.

#[IPv4 section]#
%
% Why is PI address space required rather than PA address space?

why-pi-v4: Current LIR can not provide enough PA /24 from different /16.

% Is the End User requesting extra address space for routing and/or
% administrative reasons? If yes, explain why.

why-routing-v4: yes



% Please confirm if the End User is aware of the consequences and disadvantages
% of PI address space? (yes/no)
% For details, you can refer to section 8.0[U+FFFD]PAvs. PI Address Space[U+FFFD] of the IPv4
% Address Allocation and Assignment Policies.

confirmation-v4: yes

Each block needs to be globally routable, therefore each should be a /24 minimum.

% ADDRESSING PLAN
% How will the End User use this IPv4 address space?
%
%       Subnet       Immediate    Intermediate  Entire   Purpose
%       size (/nn)   Requirement  Requirement   Period
subnet: /24                                     x        DNAT to evaluation host
subnet: /24                                     x        DNAT to evaluation host
subnet: /24                                     x        DNAT to evaluation host
subnet: /24                                     x        DNAT to evaluation host
subnet: /24                                     x        DNAT to evaluation host
subnet: /24                                     x        DNAT to evaluation host
totals: /21
number-of-subnets: 6

#[IPv6 section]#
%
% Why is PI address space required rather than PA address space?

why-pi-v6:

% Is the End User requesting extra address space for routing and/or
% administrative reasons? If yes, explain why.

why-routing-v6:

% Please confirm if the End User is aware of the consequences and disadvantages
% of PI address space? (yes/no)
% For details, you can refer to section 8.0[U+FFFD]PAvs. PI Address Space[U+FFFD] of the IPv4
% Address Allocation and Assignment Policies.

confirmation-v6:



%ADDRESSING PLAN
% How will the End User use this IPv6 address space?
%
%       Subnet       Immediate    Intermediate  Entire   Purpose
%       size (/nn)   Requirement  Requirement   Period
subnet:
subnet:
totals:
%
% Please list the Autonomous System Numbers and email contact addresses
```

```
% of the peering partners for the requested IPv6 PI assignment.

peering-v6:
peering-v6:
#[ASN section]#

%[ADDRESS SPACE TO BE ANNOUNCED]%
% If this ASN will originate other prefixes than are requested
% in this request, please list these below.

prefix-asn:

% If you require a 16-bit AS Number instead of a 32-bit AS Number,
% please indicate this below and tell us why. For more information,
% see http://www.ripe.net/news/asn-32-guide.html

as-number-type: 32-bit [change as required]
why-16-bit:

% Please list the Autonomous System Numbers and email contact addresses
% of the peering partners.

peering-asn:
peering-asn:

#[SUPPORTING DOCUMENTATION]#

% Please add more information if you think it will help us understand
% this request. You can attach a network diagram or other relevant
% supporting documentation.
See Research Proposal Attached.


%<add more information>
#[ DATABASE TEMPLATE IPv4]#
%
% If you are requesting IPv4, complete this IPv4 database template.
% If you are not requesting IPv4, please remove this IPv4 database template.

inetnum:        <leave empty>
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:       Temporary assignment
               ================================================
                Duration of assignment:
               ================================================
                Start date: 20120107
                End date:   20120128
               ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
changed:        hostmaster@ripe.net
source:         RIPE
```

O Used IPv4 Networks

O.1 Network: 145.100.109.0/24

```
whois 145.100.109.0/24
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '145.100.96.0 - 145.100.111.255'

inetnum:        145.100.96.0 - 145.100.111.255
netname:        UvA-Master-SNE-NET
descr:          Universiteit van Amsterdam
descr:          Master SNE
descr:          www.os3.nl
country:        NL
admin-c:        MSNE-RIPE
tech-c:         MSNE-RIPE
status:         ASSIGNED PI
mnt-by:         SN-LIR-MNT
mnt-irt:        irt-SURFcert
source:         RIPE # Filtered

role:           UvA Master SNE
address:        UvA Master SNE
                SNE Room B1.23
                Science Park 908
                NL-1098XH Amsterdam
                The Netherlands
remarks:        Please use abuse@os3.nl for complaints and/or abuse,
remarks:        for further/other information see: http://www.os3.nl/
abuse-mailbox:  abuse@os3.nl
admin-c:        JPV1024-RIPE
tech-c:         JPV1024-RIPE
mnt-by:         OS3-MNT
nic-hdl:        MSNE-RIPE
source:         RIPE # Filtered

% Information related to '145.100.0.0/15AS1103'

route:          145.100.0.0/15
descr:          SARA-LAN SURFNET-UNO
origin:         AS1103
mnt-by:         AS1103-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS3)
```

O.2 Network: 151.216.20.0/24

```
whois 151.216.20.0/24
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '151.216.20.0 - 151.216.20.255'

inetnum:        151.216.20.0 - 151.216.20.255
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:        Temporary assignment
                ================================================
                Duration of assignment: 3 weeks
                ================================================
                Start date: 20120108
                End date:   20120129
                ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '151.216.20.0/24AS31078'

route:          151.216.20.0/24
descr:          SNE-RP1-EVAL-TMP Route via Netsign
origin:         AS31078
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS3)
```

O.3 Network: 151.217.0.0/24

```
whois 151.217.0.0/24
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '151.217.0.0 - 151.217.0.255'

inetnum:        151.217.0.0 - 151.217.0.255
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:        Temporary assignment
                ================================================
                Duration of assignment: 3 weeks
                ================================================
                Start date: 20120108
                End date:   20120129
                ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '151.217.0.0/24AS31078'

route:          151.217.0.0/24
descr:          SNE-RP1-EVAL-TMP Route via Netsign
origin:         AS31078
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS1)
```

O.4 Network: 151.220.0.0/24

```
whois 151.220.0.0/24
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '151.220.0.0 - 151.220.0.255'

inetnum:        151.220.0.0 - 151.220.0.255
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:        Temporary assignment
                ================================================
                Duration of assignment: 3 weeks
                ================================================
                Start date: 20120108
                End date:   20120129
                ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '151.220.0.0/24AS31078'

route:          151.220.0.0/24
descr:          SNE-RP1-EVAL-TMP Route via Netsign
origin:         AS31078
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS2)
```

O.5 Network: 151.221.0.0/24

```
whois 151.221.0.0/24
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '151.221.0.0 - 151.221.0.255'

inetnum:        151.221.0.0 - 151.221.0.255
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:        Temporary assignment
                ================================================
                Duration of assignment: 3 weeks
                ================================================
                Start date: 20120108
                End date:   20120129
                ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '151.221.0.0/24AS31078'

route:          151.221.0.0/24
descr:          SNE-RP1-EVAL-TMP Route via Netsign
origin:         AS31078
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS1)
```

O.6 Network: 151.222.0.0/24

```
whois 151.222.0.0/24
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '151.222.0.0 - 151.222.0.255'

inetnum:        151.222.0.0 - 151.222.0.255
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:        Temporary assignment
                ================================================
                Duration of assignment: 3 weeks
                ================================================
                Start date: 20120108
                End date:   20120129
                ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '151.222.0.0/24AS31078'

route:          151.222.0.0/24
descr:          SNE-RP1-EVAL-TMP Route via Netsign
origin:         AS31078
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS1)
```

O.7 Network: 151.223.0.0/24

```
whois 151.223.0.0/24
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '151.223.0.0 - 151.223.0.255'

inetnum:        151.223.0.0 - 151.223.0.255
netname:        SNE-RP1-EVAL-TMP
descr:          Tobias Fiebig
country:        NL
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
remarks:        Temporary assignment
                ================================================
                Duration of assignment: 3 weeks
                ================================================
                Start date: 20120108
                End date:   20120129
                ================================================
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '151.223.0.0/24AS31078'

route:          151.223.0.0/24
descr:          SNE-RP1-EVAL-TMP Route via Netsign
origin:         AS31078
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS2)
```

O.8 Network: 195.191.197.0/24

```
whois 195.191.197.0/24
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '195.191.196.0 - 195.191.197.255'

inetnum:        195.191.196.0 - 195.191.197.255
netname:        WYBT-NET
descr:          Tobias Fiebig
remarks:        WYBT-NET assigned PI Space
country:        DE
org:            ORG-wA159-RIPE
admin-c:        WYBT-RIPE
tech-c:         WYBT-RIPE
status:         ASSIGNED PI
mnt-by:         RIPE-NCC-END-MNT
mnt-lower:      RIPE-NCC-END-MNT
mnt-by:         WYBT-MNT
mnt-by:         NETSIGN-MNT
mnt-routes:     WYBT-MNT
mnt-routes:     NETSIGN-MNT
mnt-domains:    WYBT-MNT
mnt-domains:    NETSIGN-MNT
source:         RIPE # Filtered

organisation:   ORG-wA159-RIPE
org-name:       Tobias Fiebig
org-type:       other
address:        Natrupper Str. 98
                49090 Osnabrueck
                GERMANY
abuse-mailbox:  abuse@wybt.net
mnt-ref:        WYBT-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

person:         Tobias Fiebig
address:        Natrupper Str. 98
                D-49090 Osnabrueck
                GERMANY
phone:          +495413436597
mnt-by:         WYBT-MNT
nic-hdl:        WYBT-RIPE
source:         RIPE # Filtered

% Information related to '195.191.196.0/23AS31078'

route:          195.191.196.0/23
descr:          WYBT-NET Route via Netsign
origin:         AS31078
mnt-by:         NETSIGN-MNT
mnt-by:         WYBT-MNT
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.50.5 (WHOIS2)
```