Investigating Characteristics of Attacks on Public Cloud Systems

Davide Bove
IT Security Infrastructures
Friedrich-Alexander-University Erlangen-Nuremberg
Erlangen, Germany
davide.bove@fau.de

Tilo Müller
IT Security Infrastructures
Friedrich-Alexander-University Erlangen-Nuremberg
Erlangen, Germany
tilo.mueller@cs.fau.de

***Abstract*—In this work, honeypots were set up on several public cloud infrastructures of Amazon, Microsoft and Google located in different regions around the world, including North America, Asia and Europe. The honeypots, simulating different popular services like SSH and VNC, were used to collect data over a period of two month, resulting in over 170 million log entries. Further analysis of the log entries regarding *attack patterns* and *geographic characteristics* are presented in this paper. For example, the attacks originated from 216 countries involving 268,614 unique IPs, dominated by China with a share of 25.83%.**

***Index Terms*—cloud computing, security, public cloud, attack, intrusion detection, honeypot**

I. Introduction

For years a recent trend for companies has been to move IT infrastructures from private data centers to specialized public cloud providers. Over the years, a number of large-scale data breaches happened on cloud-based systems [16, 17, 21]. These attacks not only affect private companies, but also their customers and users, who wittingly or unwittingly upload their most personal photos and videos, contact lists and private messages to the cloud.

To learn more about the reasons and methods of such attacks, honeypots can be deployed to capture the interaction of attackers with cloud systems. Honeypots are decoy systems, as they *attract* attackers and trick them to interact with them. They are set up to "be probed, attacked or compromised" [23] and usually have no economic value added for a business. They are not connected to any production systems and are merely part of the underlying company network. Any user would need to explicitly search for such systems. If the threat is new and still unknown to Intrusion Detection Systems (IDS) and anti-malware solutions, forensic investigators can analyze and learn how a vulnerability is exploited. This is achieved by collecting and analyzing the log files of the honeypot system. Insights from the analysis can be used to implement new threat signatures for IDS and firewalls, for example.

For this work, honeypots were used for research, analyzing current threats and methods of attackers. Specifically, *low-interaction* honeypots were deployed, a category first defined by Makube and Adams [15]. These honeypots simulate a limited portion of a real system, usually confined to a single service or file system, as opposed to *high-interaction* honeypots which often consist of an actual operating system and real services. High-interaction systems generally yield more relevant data for analysis, but for our experiment, low-interaction honeypots were sufficient to analyze how popular services are exploited.

A. Contributions

In this paper, our goal is to get a holistic view about current cyber threats and dangers affecting systems hosted on platforms of public cloud providers around the world. The contributions of our work are:

- *Attack Patterns*: We present the results of our empirical research of log entries from honeypots that were hosted at Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP). For a period of 63 days, we collected data from real-world attacks and analyzed recurring patterns in access credentials, session durations and post-compromise actions. For example, the most typical SSH attacks log-in as root user with empty password, last 0 to 30 seconds and perform several cleanup operations, trying to hide their activities from log files.
- *Geographic Characteristics*: We set up honeypot instances of Cowrie, Vnclowpot, Honeytrap, Mailoney, Dionaea, Glastopf and RDPY on five physically different machines in North America, Asia and Europe. While the location of the honeypots was centralized, the attacks originated from IPs world-wide, dominated by China (25%), the United States (25.21%), Russia (14.50%) and a single IP from the Seychelles (~5%). For example, in Russia the majority of attacks is generated at 7 a.m. and 7 p.m. local time, indicating a large amount of attacks that is started before and after typical business hours.

B. Related Work

Honeypots have been used extensively to research network traffic and malware. The majority of papers in the field involve honeypot software operating on regular physical hardware.

The security of cloud instances of various providers has first been researched in 2012 [3]. Incoming traffic was analyzed using a limited number of honeypots for different cloud providers. Compared to our study, the duration of their experiment and the available resources were limited, and the results

are not as representative. Similar studies on cloud services focus on SSH attacks only [1, 13]. Related research focuses on attack vectors of mobile devices, using honeypots that emulate Android and iOS systems [29] and Internet-of-Things (IoT) devices [5]. Multiple studies set up honeypots on local or virtual systems to catch and analyze malware samples [2, 12, 27]. There are also implementations of honeypots for detecting botnets [23], both on physical hosts [4] and cloud infrastructures [5].

II. Experimental Setup

For the experiment, a number of five servers were used. Table I shows the configuration of the virtual instances, with two servers belonging to Amazon Web Services (AWS), two servers from Microsoft Azure and one machine provided by the Google Cloud Platform (GCP).

TABLE I: Overview of used servers for the experiment

Server name	Provider	Region identifier	Actual region
aws-us	Amazon	us-east-2	Ohio
aws-mumbai	Amazon	ap-south-1	Mumbai
azure-us	Microsoft	East US 2	Virginia
azure-eu	Microsoft	North Europe	*not specified*
gcp-us	Google	us-east1-b	South Carolina

A. Architecture

The physical machines are distributed over three different regions of the world: North America, Asia and Europe. For every provider, one machine was selected that is located in the east of the US for comparison. The other regions were selected based on operational costs. The goal was to have a geographical comparison of the results, based on the location of the honeypot system.

In order to reduce the financial costs, the machines were set up with enough computing power to run the honeypots, but limited storage space. Therefore, a sixth server was used to store the collected log data. A similar architecture was first proposed in [28], where multiple honeypots send their log data to a central database for storage and analysis. Figure 1 depicts the relationships between the servers.

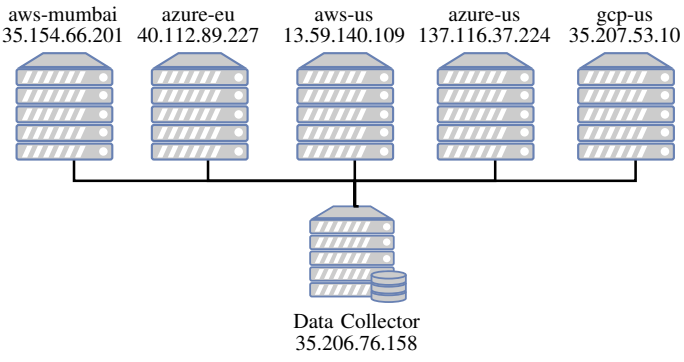

Fig. 1: Architecture of the honey network

In our setup, the data collector server hosts Elasticsearch [9], a search engine with an integrated document-based database. The software is part of the ELK stack, a popular combination of three individual tools for analytics and data visualization: Elasticsearch, Logstash and Kibana. This combination is used in similar works [1, 18] to perform analysis on data sets of any size. Logstash [11] is a software that parses various log files and converts them to a data format that is readable by Elasticsearch. It can handle both binary and plain text files, which is ideal to unify the outputs of different honeypot tools. The last software is Kibana [10], a visualization tool that directly communicates with Elasticsearch. It offers an interactive user interface for working on the data and allows to create visualizations and aggregations on the Elasticsearch database.

To collect and store the honeypot outputs, the data collector uses two methods to retrieve information. The first method collects the raw output of the different honeypot tools. For this, SSH is configured on the honeypots such that the data collector can safely download the log files through the SSH File Transfer Protocol (SFTP). As an additional measure, the connection is protected by public-key cryptography and password authentication is explicitly disabled. The second method involves Logstash, which is deployed on every honeypot system and sends the relevant log entries to the Elasticsearch instance on the data collector. This has multiple advantages: the data is available in near real-time for live analysis and it is directly saved to the database. This allows to work on the data in a timely manner without having to import or convert the raw data in the analysis phase of the experiment, saving time and resources.

B. Software

All servers for the experiment run the same honeypot tools, which simulate a wide range of services. To facilitate the setup of all servers, T-Pot [6] was used. T-Pot offers a collection of preconfigured honeypots, an IDS and various other tools to enrich the honeypot data. Through various scripts and the use of the container technology Docker, the honeypots are deployed in separate environments on the system, but run on the same network interface.

For the data collection we exclusively used low-interaction honeypots. Compared to high-interaction honeypots, they are easier to deploy and configure, as they mostly operate on the application layer and can be easily secured and put in isolated environments. The scope of the experiment is to monitor the interactions with specific popular services, thus low-interaction honeypots were used, even though the output of high-interaction honeypots is often considered superior. Also, Denial-of-Service (DoS) attacks were excluded, as they offer no insights into single attacker patterns. The following software packages were deployed for the data collection:

- Cowrie [19]
- Dionaea [7]
- Glastopf [25]
- Honeytrap [30]
- Mailoney [8]
- RDPY [22]

- Vnclowpot [14]

Additionally, the Suricata IDS [20] was used in order to identify known threats and the fingerprinting software p0f [31] added information about the incoming requests.

C. Security

The security of the connection between honeypots and data collector is a key problem. Any attacks on the data collector or the connection to the honeypots can harm the integrity and correctness of the log data. Additionally, a compromised honeypot system could be hijacked to access the data collector. While the honeypots used in our experiment do not have any elevated privileges on the system, potential vulnerabilities in the software could lead to a machine takeover, so the probability of a compromise is not negligible. In our experiment, these dangers are countered by two different security policies.

First, as the single honeypot systems should be accessible from the Internet, the firewalls on the systems are configured to allow all inbound connections, which means any port on the system is accessible by any IP. At the same time the outbound connections, the connections going from the honeypot to other hosts, are strictly limited. Only connections on port 80 (HTTP) and to the data collector are allowed. This is a compromise to both minimize damage if a host is compromised and allow malicious scripts to download payloads and additional software, exposing more useful data.

The second policy affects the connection between data collector and honeypot. The data collector has more strict firewall rules, only allowing incoming connections to Elasticsearch on port 9200. The access is further limited to the specific honeypot IPs, therefore no external access is possible. This policy was adapted to include additional personal computers in order to access the data for analysis once the data collection phase was terminated.

III. Results

The experiment was conducted between March 2018 and August 2018 for 63 days and 176,158,872 individual log entries were collected. These involve about 268,614 unique origin IPs. The following results are divided into two categories: Attack patterns and geographic analysis.

A. Analyzing Attack Patterns

Every connection to the honeypots has a specific motivation and target. As Table II shows, the majority of requests targeted SSH, Telnet and VNC services. The remaining 24.6 % of connection is split between Honeytrap, which covers any unused port, and the other honeypots.

1) Credentials: We further inspected SSH sessions, as collected by the Cowrie honeypot, in order to find common patterns used by attackers. For this, Cowrie is configured to accept up to five username and password combinations before it gives access to a user. During the authentication phase, these credentials are logged by the honeypot. As most password-protected systems, SSH authentication is vulnerable to brute-force or dictionary attacks, therefore password authentication is often disabled and replaced with public-key authentication, which requires elevated resources and time to break.

TABLE II: Share of connections to honeypots

Honeypot	Percentage
Cowrie	39.49 %
Vnclowpot	35.91 %
Honeytrap	12.37 %
Mailoney	6.45 %
Dionaea	3.37 %
Glastopf	< 2.00 %
RDPY	< 1.00 %

TABLE III: Top usernames and passwords attempted during SSH attacks

Username	Count	Password	Count
root	2,786,944	(empty)	344,588
admin	657,464	system	260,939
enable	259,427	sh	201,674
shell	259,314	admin	106,217
(empty)	149,456	1234	104,116

The captured credentials can be seen in Table III. Some usernames, such as *root*, *Administrator* or *admin* are the most common ones for privileged accounts on Linux and Windows systems. Therefore a high occurrence was expected for these.

Regarding the passwords used during the login procedure, there are more distinct results. This is because a password can be totally random and unique, while usernames are often chosen to be more memorable and a lot of systems share the same usernames. It is also notable how most attempts at guessing passwords try to omit the password. This could be intentional or caused by faulty automated clients as a reaction to unsuccessful password attempts.

2) Session Duration: As the honeypot saves timestamps of all interactions, it was possible to calculate the duration of every SSH session. A session starts with the first connection request to the server and ends when the client disconnects from the server. If during a session the client successfully logs into the service, a terminal session is created, where the user is able to enter commands.

Figure 2 shows the duration of both types of sessions in comparison. The duration of full sessions ranges primarily between 0 and 30 seconds. This includes different attempts at password guessing, or failed attempts that are aborted quickly. Between 30 and 60 seconds, there is a significant increase in terminal sessions. Therefore, sessions that last this long have a greater chance at using the hacked credentials to login and execute commands. Also, a greater terminal session duration might be an indicator that users manually enter commands, while quick SSH sessions may indicate automated hacking attempts.

The vast majority of attacks on SSH take less than a minute. The greater the duration of full sessions and terminal sessions, the greater the probability for a human behind the request. Since it is difficult to define a threshold for a correlation between session length and attacker type, more factors need to be considered.

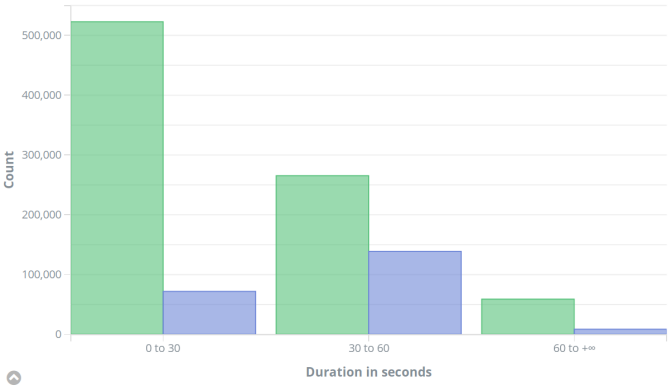

Fig. 2: Average duration of full SSH sessions (green) compared to terminal sessions (blue).

3) Post-compromise Actions: Once the authentication process is passed, attackers usually connect with their successful credentials to access the system through an interactive terminal session. The Cowrie honeypot simulates a Unix-like system, therefore typical bash commands (e.g. `cd`) are available. By analyzing these inputs, one can gain more insights on how attackers operate.

During the data acquisition phase, a total of 83,278 entries were collected, consisting of 79,946 (96 %) valid commands. Of these, 778 distinct commands were identified. The mean duration of a command, more precisely the timespan between two commands, is approximately 792 milliseconds.

We manually analyzed the commands and developed a categorization method, adapted from two related approaches [24, 26]. Every command was mapped to one of the following categories:

1) **Check** – Information gathering and exploring. Check the system and identify versions, users, and file system.
2) **Persist** – Secure the access to the system. This step involves increasing the foothold into the system, creating (privileged) users and/or changing passwords to be able to access the system even after disconnecting.
3) **Exploit** – Install and run software and scripts. This should be the main action of an attacker.
4) **Cleanup** – Cover traces and remove evidence of an intrusion. Often attackers try to eliminate traces of the compromise by manipulating or deleting log files. Deactivation of terminal logging features also falls under this category.
5) **Human** – Commands that hint at a human attacker. Some commands only make sense for humans, such as `man` or `clear`.

A summary of the resulting categorization is provided in Table IV. Any entry that could not be mapped to a valid command is marked as *unmatched*. Commands that had no significant side effects, such as `sleep`, were grouped as *no operation* commands. Since the categorization was developed based on the analyzed commands, over 97.5 % of commands could be successfully matched to one of the categories.

TABLE IV: Categorization of individual commands

Category	Commands	Percentage
Check	51	6.56 %
Persist	1	0.13 %
Exploit	653	83.93 %
Cleanup	26	3.34 %
Human	25	3.21 %
(unmatched)	19	2.44 %
(no-op)	3	0.39 %
Total	778	100.00 %

Once the categories are assigned to commands, they can be chained to build specific attack patterns. These patterns have actions – the specific command – and states, which is the assigned category. When state changes are analyzed, for example going from a *Check* to an *Exploit* phase, they form a graph that can be plotted as a state diagram.

This concept is visualized in Figure 3. The plot shows the type of operations an attacker has taken and the order in which these are executed, including only the main categories. For example, most attackers either start an *Exploit* or try to disable logging mechanisms directly after an intrusion.

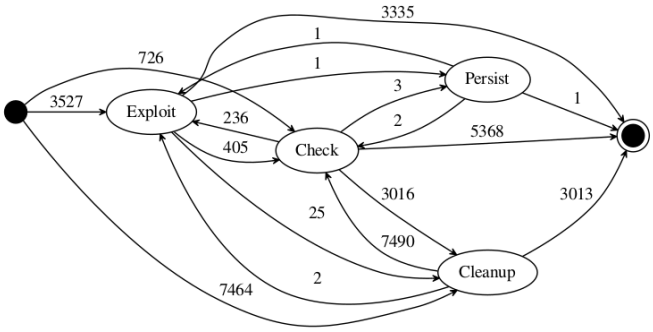

Fig. 3: State diagram of attacker behavior after login

The most encountered pattern is composed by several *Cleanup* operations, followed by *Check* commands. This pattern is repeated several times per day.

B. Geographic Observations

Through geolocation databases, it is possible to map IP addresses to geographic locations. The T-Pot framework includes the MaxMind GeoLite2[1] database, which was also used for the experiment. Overall, traffic originated from 216 different countries. The following sections present results of our traffic analysis, differences between providers, regional differences and an hourly comparison of attacks.

1) Traffic: Using IP geolocation, we matched request IPs to countries. The resulting overview shows where the requests to the honeypots originate from and how much of the traffic is generated by specific countries or users.

In total, devices from 216 different countries have connected to our honeypots. Table V shows the ten countries with

[1]https://dev.maxmind.com/geoip/geoip2/geolite2/

the most amount of requests. In summary, China, USA and Russia are responsible for nearly 2/3 of the connections to our honeypots, while the other countries have a significantly smaller share. As the results show, the other 206 countries that are missing from the table have a total share of 12.88 %, with any country contributing below 1.4 % of connections.

TABLE V: Top 10 originating countries

Country name	Number of connections	Percentage of total
China	26,342,814	25.83 %
United States	25,713,416	25.21 %
Russia	14,793,086	14.50 %
Seychelles	5,881,405	5.77 %
Netherlands	4,603,325	4.51 %
Hong Kong	3,770,987	3.70 %
Canada	3,007,309	2.95 %
Vietnam	1,809,847	1.77 %
France	1,515,002	1.49 %
Germany	1,426,856	1.40 %
Total (top 10)	88,864,047	87.12 %
Total	101,994,243	100.00 %

These results are even more revealing when looking at the individual IP addresses. The most active IP was located in China and accounted for 12.53 % of all and 48.5 % of Chinese connections. This can also be observed with other countries, such as Seychelles, Hong Kong, the Netherlands and Canada, where a single IP makes over 50 % of requests in the respective country. In comparison, USA has two IPs in the top 10 list, with around 13 % and 16 % of all US connections. In short, most countries that generated significant traffic to our honeypots have only a limited number of IPs from where attacks are launched. This could be attributed to both criminals and specific organizations with elevated resources, such as governments or research facilities.

2) Provider Differences: Every cloud instance in our experiment has the same configuration, runs the same honeypots and they only differ in provider and physical location. For our provider analysis, we compared three servers located in the US east region. The findings show that GCP was targeted most, with 5,292,372 connections compared to AWS (4,124,037) and Azure (2,973,389). As Azure is often associated with Microsoft services and systems and our honeypots only offered limited emulation of Windows services, we assume that our deployed honeypots were not attractive enough for attackers of the Azure cloud platform. We did not find further indicators that could reasonably explain the above differences.

3) Regional Differences: We selected three different locations to compare, having three servers based in the US. With a server located in Europe and one in India, we have a measure of how other continents are affected by the attacks, compared to the US region.

We found that the US servers collected 67 % of traffic, averaging 4,129,932 requests per server, while India collected 3,150,499 requests (17 %) and Europe 2,921,225 requests (16 %). As the cloud providers and several popular services on the Internet are based in the US, attackers might focus on infrastructures located in the US regions to compromise attractive targets.

4) Daytime Evaluation: An approach that is not much explored for intrusion detection is the extraction of timestamps from the data in order to attribute attack waves to specific countries or regions. The main idea is to aggregate timestamps of incidents and time zones retrieved through fingerprinting tools.

In our analysis, we aggregated the number of attacks and the individual timestamps and created a distribution over the day. The results show the distribution of attacks over the day, for any country in the local timezone determined by the geographic IP location.

The results for Russia are shown in Figure 4. This graph has peaks around 7 am and 7 pm, suggesting that most attacks happen outside of business hours. Such graphs can be created for countries, regions or even cities, depending on the precision of the geolocation data source, even though a minimum number of requests need to be collected over a longer period of time.

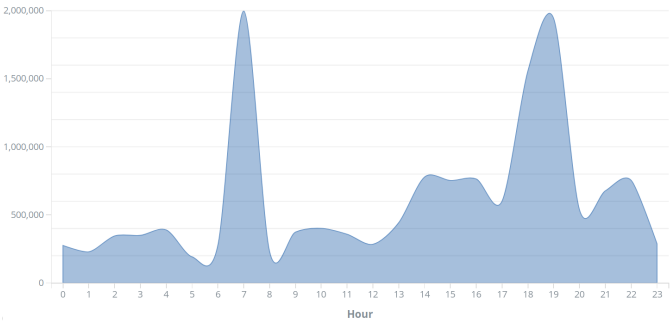

Fig. 4: Hourly distribution of connections originating in Russia

IV. Conclusion

In this work we presented a methodology to capture honeypot data and analyze it for typical attack patterns and regional differences. The analysis reveals that the majority of attacks originate in China, USA and Russia and target mostly VNC and SSH services. Often the attacks are automated and repeated over time, and IP ranges of cloud providers are constantly scanned for exposed or vulnerable services. We also demonstrated how honeypot data can be used for cyber attribution when combined with additional data sources. For the future, our setup and our database of log entries offers the potential for an even more in-depth research in this field.

References

[1] H. Almohannadi et al. “Cyber Threat Intelligence from Honeypot Data Using Elasticsearch”. In: *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. May 2018, pp. 900–906.

[2] Paul Baecher et al. “The Nepenthes Platform: An Efficient Approach to Collect Malware”. In: *Recent Advances in Intrusion Detection, 9th International Symposium, RAID 2006, Hamburg, Germany, September 20-22, 2006, Proceedings*. 2006, pp. 165–184.

[3] Stephen Brown et al. “Honeypots in the cloud”. In: *University of Wisconsin-Madison* (2012).

[4] Rajab Challoo and Raghavendra Kotapalli. “Detection of botnets using honeypots and p2p botnets”. In: *International Journal of Computer Science and Security (IJCSS)* 5.5 (2011), p. 496.

[5] Ryan Chinn. “Botnet Detection: Honeypots and the Internet of Things”. PhD thesis. University of Arizona, 2015.

[6] Deutsche Telekom AG. *T-Pot*. Version 17.10. URL: https://github.com/dtag-dev-sec/tpotce (visited on October 1, 2018).

[7] DinoTools. *Dionaea*. Version 0.8.0. URL: https://github.com/DinoTools/dionaea (visited on October 1, 2018).

[8] Brandon Edmunds. *Mailoney*. Version 0.1. URL: https://github.com/awhitehatter/mailoney (visited on October 1, 2018).

[9] Elastic. *Elasticsearch*. Version 5.6.9. URL: https://www.elastic.co/products/elasticsearch (visited on October 1, 2018).

[10] Elastic. *Elasticsearch*. Version 5.6.9. URL: https://www.elastic.co/products/kibana (visited on October 1, 2018).

[11] Elastic. *Logstash*. Version 5.6.9. URL: https://www.elastic.co/products/logstash (visited on October 1, 2018).

[12] Esmaeil Kheirkhah et al. “An Experimental Study of SSH Attacks by using Honeypot Decoys”. In: *Indian Journal of Science and Technology* 6.12 (2013). ISSN: 0974 -5645.

[13] Ioannis Koniaris, Georgios Papadimitriou, and Petros Nicopolitidis. “Analysis and visualization of SSH attacks using honeypots”. In: *EUROCON, 2013 IEEE*. IEEE. 2013, pp. 65–72.

[14] Stuart McMurray. *vnclowpot*. URL: https://github.com/magisterquis/vnclowpot (visited on October 1, 2018).

[15] Iyatiti Mokube and Michele Adams. “Honeypots: concepts, approaches, and challenges”. In: *Proceedings of the 45th Annual Southeast Regional Conference, 2007, Winston-Salem, North Carolina, USA, March 23-24, 2007*. 2007, pp. 321–326.

[16] Eric Newcomer. *Uber Paid Hackers to Delete Stolen Data on 57 Million People*. November 2017. URL: https://www.bloomberg.com/news/articles/2017-11-21/uber-concealed-cyberattack-that-exposed-57-million-people-s-data (visited on May 30, 2018).

[17] Lily Hay Newman. *Blame Human Error for WWE and Verizon’s Massive Data Exposures*. July 2017. URL: https://www.wired.com/story/amazon-s3-data-exposure/ (visited on May 30, 2018).

[18] Amos O Olagunju and Farouk Samu. “In search of effective honeypot and honeynet systems for real-time intrusion detection and prevention”. In: *Proceedings of the 5th Annual Conference on Research in Information Technology*. ACM. 2016, pp. 41–46.

[19] Michel Oosterhof. *Cowrie*. Version 1.5.1. URL: https://github.com/micheloosterhof/cowrie (visited on October 1, 2018).

[20] Open Information Security Foundation. *Suricata*. Version 4.0.5. URL: https://suricata-ids.org/ (visited on October 1, 2018).

[21] Dan O’Sullivan. *Black Box, Red Disk: How Top Secret NSA and Army Data Leaked Online*. November 2017. URL: https://www.upguard.com/breaches/cloud-leak-inscom (visited on May 30, 2018).

[22] Sylvain Peyrefitte. *RDPY*. Version 1.3.2. URL: https://github.com/citronneur/rdpy (visited on October 1, 2018).

[23] Niels Provos and Thorsten Holz. *Virtual Honeypots - From Botnet Tracking to Intrusion Detection*. Addison-Wesley, 2008. ISBN: 978-0-321-33632-3.

[24] Daniel Ramsbrock, Robin Berthier, and Michel Cukier. “Profiling Attacker Behavior Following SSH Compromises”. In: *The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2007, 25-28 June 2007, Edinburgh, UK, Proceedings*. 2007, pp. 119–124.

[25] Lukas Rist et al. “Know your tools: Glastopf - A dynamic, low-interaction web application honeypot”. In: *The Honeynet Project* (2010).

[26] Gabriel Salles-Loustau et al. “Characterizing Attackers and Attacks: An Empirical Study”. In: *17th IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2011, Pasadena, CA, USA, December 12-14, 2011*. 2011, pp. 174–183.

[27] Tomas Sochor and Matej Zuzcak. “Study of Internet Threats and Attack Methods Using Honeypots and Honeynets”. In: *Computer Networks*. Ed. by Andrzej Kwiecień, Piotr Gaj, and Piotr Stera. Cham: Springer International Publishing, 2014, pp. 118–127. ISBN: 978-3-319-07941-7.

[28] Pavol Sokol, Patrik Pekarčık, and Tomáš Bajtoš. “Data collection and data analysis in honeypots and honeynets”. In: *Proceedings of the Security and Protection of Information. University of Defence* (2015).

[29] Matthias Wählisch et al. “Design, Implementation, and Operation of a Mobile Honeypot”. In: *CoRR* abs/1301.7257 (2013).

[30] Tillmann Werner. “Honeytrap – Ein Meta-Honeypot zur Identifikation und Analyse neuer Angriffe”. In: *Proceedings of the 14th DFN-CERT Workshop Sicherheit in Vernetzten Systemen*. 2007.

[31] Michal Zalewski. *p0f*. Version 3.09b. URL: http://lcamtuf.coredump.cx/p0f3/ (visited on October 1, 2018).