What Scanners do at L7?
Exploring Horizontal Honeypots for Security Monitoring

Thomas Favale,[1] Danilo Giordano,[1] Idilio Drago,[2] Marco Mellia[1]
[1]Politecnico di Torino, first.last@polito.it
[2]University of Turin, idilio.drago@unito.it

***Abstract*—Honeypots are a common means to collect data useful for threat intelligence. Most efforts in this area rely on vertical systems and target a specific scenario or service to analyse data collected in such deployment. We here extend the analysis of the visibility of honeypots, by revisiting the problem from a horizontal perspective. We deploy a flexible honeypot system hosting multiple services, relying on the T-Pot project. We collect data for 5 months, recording millions of application requests from tens of thousands of sources. We compare if and how the attackers interact with multiple services. We observe attackers that always focus on one or few services, and others that target tens of services simultaneously. We dig further into the dataset, providing an initial horizontal analysis of brute-force attacks against multiple services. We show, for example, clear groups of attackers that rely on different password lists on different services. All in all, this work is our initial effort to build a horizontal system that can provide insights on attacks.**

1. Introduction

To gain visibility into network attacks, *honeypots* are a common means to collect data. Their goal is to engage with the attackers with either emulators that replicate the basic functions of real systems (low-interaction honeypots), or fully-working live systems deployed in controlled environments (high-interaction honeypots). Usually, honeypots are deployed as vertical systems targeting a specific scenario [1] such as databases, terminal servers, or web applications [2], [3], [4], each supporting the respective protocols.

Researchers have focused on different aspects when dealing with honeypots. Most work has focused on the introduction of new honeypots, focusing on attacks against particular services [5], [6]. Others evaluate the effectiveness of different honeypots [7] in exposing useful data. Another body of works focuses on techniques to detect, and avoid the detection, of honeypots [8], [9]. Considering the analysis of honeypot logs, previous works compare deployments in different geographic locations [10] and study how attackers respond to different honeypot sophistication [11]. However, such works focus on single deployments or services, providing an in-depth analysis of logs in these vertical deployments.

We here aim at extending these analyses by revisiting the visibility offered by honeypots from a *horizontal perspective*. We compare if and how the same attackers interact with different honeypots offering multiple applications. Do attackers typically attack a single system or do they extend the attack surface on multiple systems? Do they use the same strategies for multiple honeypots?

To answer these questions, we deploy a flexible honeypot system hosting multiple services. We rely on the T-Pot project [12] that organises and bundles multiple low-interaction honeypots. We collect data for 5 months at the transport and application layers, recording millions of application requests from tens of thousands of sources. Exploiting this perspective, we observe scanners that always attack the same service as well as horizontal attackers targeting multiple services simultaneously. We turn off our honeypots for 2 weeks, turning them back online on a different IP address space. We observe attackers that dismiss the previous targets, discover new systems and return to full-speed campaigns.

Finally, we dig further into the dataset, providing an initial horizontal analysis of the brute-force attacks against multiple services. We study the credentials used on login attempts against our honeypots, collecting passwords used on each system. We observe attackers using password found in known data breaches [13], but also groups of attackers relying on other password lists.

We revisit and update known facts about honeypot deployments, highlighting the greediness of some attackers against multiple services. We believe this work offers some preliminary insights that highlight the benefits of a horizontal perspective when characterising attacks captured by honeypots. The complexity and multi-facet nature of the data honeypots expose calls for cooperation on the deployment of (open) honeypot infrastructure and on the sharing of honeypot data. For that, we here report initial analysis of our dataset and deployment, which are available to other researchers upon NDA agreement to protect eventual sensitive information present in the data.

After describing our deployment in Sec. 2, we provide a general characterisation of L7 traffic reaching our deployment in Sec. 3. Then, we describe the brute-force attempts recorded by multiple honeypots in Sec. 4. Sec. 5 summarises related work, while we list conclusions and future work in Sec. 6.

2. Methodology and Dataset

We set up an infrastructure that relies on the honeypots organised and distributed by the T-Pot project [12]. In this work we report on a subset of the T-Pot honeypots, excluding Telnet and SSH, which we leave for future work. All honeypot services are low-interaction honeypots [1]. We configure T-Pot to expose the services listed in Table 1. Each honeypot logs and registers all application (L7)

TABLE 1. HONEYPOT SERVICES AND AMOUNT OF TRAFFIC SEEN IN OUR DEPLOYMENT.

Service	Sender Addr.	L7 Requests
smb [2]	30 854	25 348 188
http [2], [4]	18 370	1 931 553
mssql [2]	7 119	396 391
rdp [14]	4 813	46 036 456
ftp [2]	1 845	20 834
mysql [2]	1 527	32 091
mongodb [2]	1 364	21 112
epmapper [2]	1 203	71 014
pptp [2]	912	59 488
smtp [15]	827	3 994 871
mqtt [2]	567	4 287
echo [2]	314	20 378
vnc [16]	172	12 741 040
postgresql [16]	27	28 163
pop3 [16]	12	26 194

interactions, including brute-force login attempts, requests to specific service functionality etc. We rely on the T-Pot 20.06 bundle, which deploys services on their standard port(s), as well as honeypots that perform simplistic deep packet inspection on other ports for some protocols, e.g., identifying and responding http traffic on non-standard http ports.[1]

We use two /24 networks of a university network that we reserve to run our experiments. The honeypots are thus hosted in a regular campus network, with no firewall to protect them. To monitor all the incoming traffic, we record packet-level traces using tcpdump, which are regularly processed along with all logs of the honeypots.

We activated our deployment on October 27th, 2021. We here analyse data collected until March 1st, 2022. Specifically, from October 27th, 2021 to January 25th, 2022, we configured our honeypots on 8 IP addresses in the first /24 network. All IP addresses expose precisely the same services (see Table 1). On January 25th, 2022, we shut the honeypots down for two weeks, after which we restarted all services on February 9th, 2022, using 8 previously silent IP addresses in the second /24 network. We perform this experiment to observe patterns related to the discovery and subsequent attacks against new systems.

We collected about 100 *million* application layer (L7) requests coming from more than 57 000 unique IP addresses. In the following, we generically refer to these IP addresses as "attackers", i.e., senders that have interacted with one of our honeypots at application layer at least once during the observed period. In this work we thus ignore cases where a real attacker may use multiple IP addresses, or cases where multiple real attackers reach our systems from the same IP address, e.g., on different time periods. Notice that not all of such senders are malicious. We will show later that many senders are actually crawlers, e.g., from security companies. We will highlight these cases, and give particular attention to the clearly malicious activity, such as brute-force password attacks.

Table 1 details the number of senders and of L7 requests on each honeypot. Entries are sorted by the number of senders. Unsurprisingly, the most targeted services are Samba (*smb*), (*http*), Microsoft SQL Server (*mssql*), and Remote Desktop Protocol (*rdp*). For these services, we observe thousands of senders. Interestingly, the number of attempts is clearly disproportional to the number of senders, with RDP attackers generating much more attempts than other cases. Notice the 172 attackers that target the VNC protocol. They sent more than 12 million brute-force login attempts. These figures already shows the heterogeneous picture each honeypot produces.

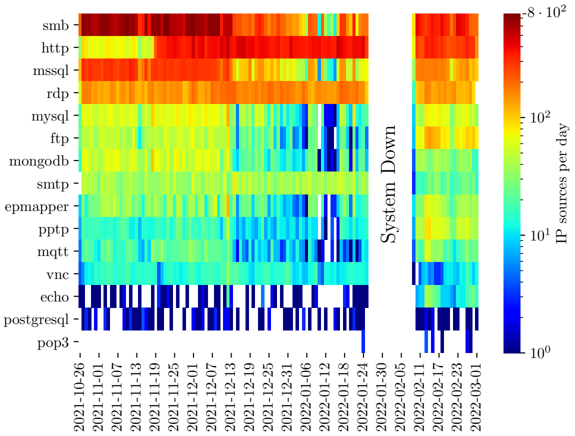


Figure 1. Number of IP sources per service for each day of observation.

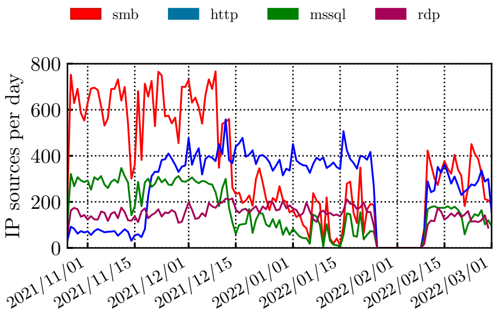


Figure 2. Evolution on number of IP sources for four honeypots.

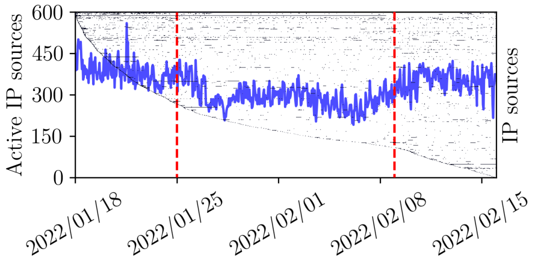


Figure 3. Activity of IP sources probing at application level around the period we shutdown the honeypots.

3. Honeypot Traffic Patterns

We report a high-level characterisation of the honeypot traffic to understand overall attacking patterns.

[1] Our setup does not include any honeypot that may be abused for amplification attacks, e.g., we disable all UDP-based services.

TABLE 2. DISTRIBUTION OF TRAFFIC PER SERVICE COMING FROM SOME KNOWN SOURCES.

Class	Service (% L7 Requests)									
	smb	http	rdp	ftp	vnc	smtp	mongodb	mqtt	mysql	others
Mirai	16.13	1.86	**59.44**	0.1	16.98	4.78	0.02	-	0.03	0.66
Google	-	**100**	-	-	-	-	-	-	-	-
Shadowserver	14.52	18.55	**37.10**	12.10	-	-	13.70	4.03	-	-
Shodan	-	-	12.66	22.15	15.82	**26.58**	-	12.03	10.76	-
Onyphe	**100**	-	-	-	-	-	-	-	-	-
Stretchoid	-	**100**	-	-	-	-	-	-	-	-
Unknown	**97.94**	0.45	0.89	0.17	-	-	0.28	0.17	-	0.1

3.1. Evolution over Time

We first check the traffic evolution over time for different honeypots. Fig. 1 shows the number of source IP addresses seen each day by the services. For readability, services are sorted by popularity. As also observed by others [1], [11] the traffic pattern is highly irregular, with sudden changes and spikes. Fig. 2 details the evolution over time for services with the largest traffic share. Notice for instance the sudden increase of *http* sources in mid-Nov and the sudden decrease of *smb* attackers in mid-Dec. These sudden changes are common, and usually are related to changes in attacking patterns [1].

Focus now on the period when we turn back on the honeypots after the shutdown. We immediately notice a pattern almost similar to the one before the shutdown. Fig. 3 details the traffic on the complete set of honeypots. Given all the source IP addresses that generate at least one L7 exchange in the whole time period, we monitor TCP/SYN segments they sent starting from one week before the shutdown, and ending one week after the restart of our infrastructure on the different /24 network.[2]

In Fig 3, the red lines highlight the offline period, while the blue line describes the number of IP sources per hour (left y-axis). Each row (right y-axis) refers to a given IP address. Dots are single TCP/SYN packets observed at a given time. IP addresses are sorted by increasing time as they are seen in the infrastructure during this interval. Several considerations hold:

- Attackers return over time, sometimes being active for rather long periods (see dark horizontal segments);
- Old and new attackers continue to search for the honeypots when the systems are offline. The arrival rate is smaller during the shutdown (see the dark external envelope);
- As soon as the systems become active again, attackers immediately discover them – the arrival rate of new IP sources grows again;
- No major changes are seen when comparing before and after the shutdown, i.e., attackers get back and keep trying to enter the systems (see also Fig. 1);
- The availability of new possible victims attracts new attackers (see the appearance of new attackers that appear starting from Feb. 9th only);

The solid blue line quantifies number of active IP sources per hour. The interest of attackers decreases when the honeypots are unavailable, but it does not vanish. Recall that here we consider only IP addresses performing L7 interactions at least once, so hosts scanning the network are not counted. While an average of 350 hourly IP sources are seen active before and after the shutdown, we still see around 300 of these attackers active when the infrastructure is offline.

Takeaway Attacking rate is variable over time. Attackers keep returning, even after the honeypots are unreachable for weeks. Attackers' arrival rate grows fast when new honeypots become available.

3.2. Popular Sources and Countries

We now check if source IP addresses are associated to well-known actors. For this, we tag source IP address using well-known lists of scanners and crawlers, taken from [17], which include IP addresses of security companies. We also use the well-known Mirai fingerprint to tag sources as Mirai-like node [18].

Tab. 2 details the percentage of traffic from these sources to each honeypot. Rows are sorted by popularity. Mirai attackers are the most popular ones (18586 IP addresses in total). They primarily target remote desktop applications that use *rdp* and *vnc*, often performing brute-force password attacks against these services [3]. A significant percentage of traffic to *smb* services comes from Mirai attackers too.

Next, we observe a large quantity of non-malicious sources. For example, we see many requests coming from Google IP addresses on the *http* honeypot, which we associate with Google's crawlers. We also observe traffic from security organisations. Some of them focus on specific services, e.g., Onyphe on *smb* and Stretchoid on *http*. Others scan services horizontally, such as Shadowserver and Shodan. We note that these crawlers do send L7 traffic to test applications, such as trying to login as anonymous in our ftp honeypot.

The remaining sources (42 IP addresses) target mostly the *smb* service, with small percentage of traffic to other services. These are likely bots looking for vulnerabilities.

We also break down the traffic according to the geographic location of source IP addresses. For that, we tag each IP address with its geographic location using MaxMind's GeoIP database [4]. Tab. 3 shows the traffic share (number of L7 requests) per country considering all honeypots. The geographic distribution of requests is similar to what is reported in recent previous works [11].

Takeaway Honeypots observe a large volume of non-malicious traffic, coming from crawlers of security com-

[2] The traffic *during* the shutdown refers to the initial /24 network.

[3] Mirai is known to scan also for Telnet, ADB and other protocols, which we do not consider in this work.

[4] https://www.maxmind.com/en/geoip-demo

TABLE 3. PERCENTAGE L7 REQUESTS PER COUNTRY CONSIDERING ALL HONEYPOTS.

Traffic Volume					
DE	Germany	19.86%	UK	UK	0.89%
RU	Russia	10.12%	UA	Ukraine	0.70%
US	US	5.62%	IT	Italy	0.67%
LT	Lithuania	4.79%	IR	Iran	0.63%
VN	Vietnam	4.57%	CO	Colombia	0.54%
BR	Brazil	1.41%	PA	Panama	0.15%
PL	Poland	1.05%	HK	Hong Kong	0.12%
CN	China	1.02%	BE	Belgium	0.03%

panies that do actively test the services (e.g., trying to login). Mirai-like bots still stand among malicious actors.

3.3. Vertical vs. Horizontal Activity

Here we quantify if IP sources tend to focus on single service (e.g., in vertical attacks) or multiple services (e.g., in horizontal attacks). Given two services i and j, we extract the set of sources observed for each service, i.e., $S(i)$ and $S(j)$. Then, we compute the *overlap coefficient* defined as the ratio between the intersection of the sets and the size of smallest one:

$$Overlap(i,j) = \frac{|S(i) \cap S(j)|}{\min(|S(i)|, |S(j)|)}$$

The *overlap* takes values between 0 and 1, where 0 means the intersection of the two sets is empty, while 1 means that the smallest set is included in the largest one.[5]

Fig. 4 shows a heatmap of the overlap. The warmer the colour, the higher the overlap. Focus on *smb* service (first column). Most of the sources contacting other services are not interested in *smb* (dark blue cells). Exceptions are seen for those contacting *epmapper, pptp, mssql*: 50% of them do target *smb* too. While this is largely expected for *epmapper* (a service related to Samba), it also suggests attackers search for several Microsoft services at once (e.g., *mssql* and *smb*). Similar pattern is seen for *vnc* and *rdp*, both offering remote login, with overlap at 70%.

The case for *http* is interesting. Most of those contacting the *http* honeypot also contact other services, but not *smb*, *mssql* and *rdp*, all three related to Microsoft services.

We next quantify how many services each source contacts and how often they change the contacted services. For each source, we extract the sequence of contacted services in temporal order considering the entire 5 months of data. We then compute, for each source: (i) the number of times the source changes service; (ii) the total number of unique services it contacts; and (iii) its total number of requests. Fig. 5 shows the scatter plot where each dot represents a source. Using log-log scale, the x-axis reports the total number of contacts, the y-axis is the total number of service changes, and colours show the number of unique services per source.

At the bottom, points form a dense horizontal layer. Those are sources that contacted only one service (dark blue), thus never changing protocol (0 changes, artificially reported in the base of the y-log scale). These are *vertical* attackers. Some of these sources contacted our honeypots few times; Others returned millions of times. Manual inspection confirms that these are true attackers focusing on *rdp* and *smb* protocols.

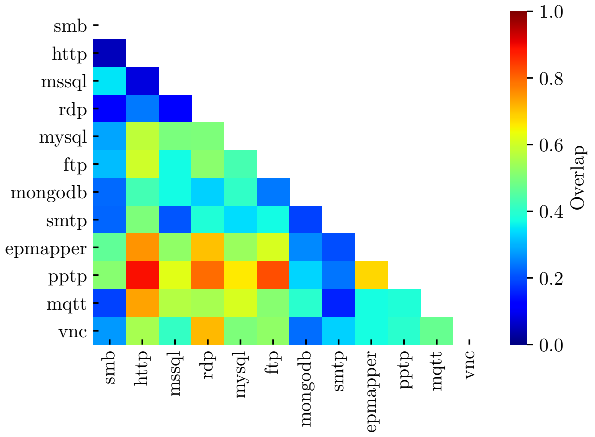


Figure 4. Source overlap among the top-12 services.

At the second layer of points from the bottom, we see sources that contacted 2 services (light blue). These are attackers that moved from one service to a second and then stayed on this second service (1 change only). A sizeable number of attackers also targeting 2 services (same light blue colour) keep alternating between the 2 services multiple times (more than 2 changes). Here we see the cases of attackers focusing on *categories* of services, such as the remote desktop cases mentioned above.

Finally, we observe multiple sources that keep rotating regularly over multiple services. These are scattered over the diagonal, showing in some case a number of changes proportional to the number of attempts. We here identify horizontal scanners, i.e., sources contacting many services (green to red dots) and alternating among these services. Some sources are also particularly active, contacting our honeypots thousands of times. We associate this behaviour with some security crawlers, which perform application-layer handshakes to check for service availability – thus, not necessarily performing malicious activities. For instance, those sources who contacted more than 10 services, sending more than 1 000 requests in total (red dots in the top right) are scanners run by security companies, the majority of which managed by *AVAST Software*.

Takeaway Honeypots observe both vertical and horizontal activity. The former is predominately attackers focusing on categories of services. The latter is dominated by scanners and security crawlers.

4. Brute-Force Attacks

We now focus on a single type of attack often observed in the honeypots: brute-force password attempts. We first study the used passwords, then we compare attackers' origins and strategies across the various honeypots.

4.1. Known vs. Unknown Passwords

Some honeypots in the T-Pot bundle deployed in our network show only the login functionality of services to

[5] Given the disparity of sources for each service (see Tab. 1) the overlap offers a clearer information than the Jaccard Index.

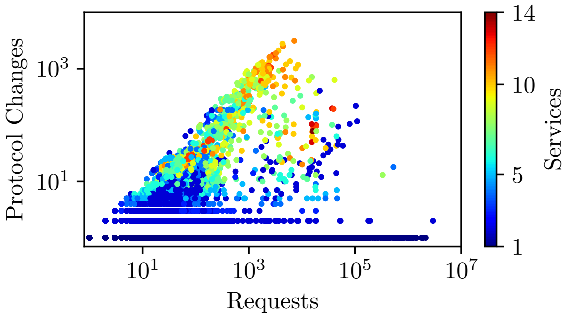


Figure 5. Attackers' activity.

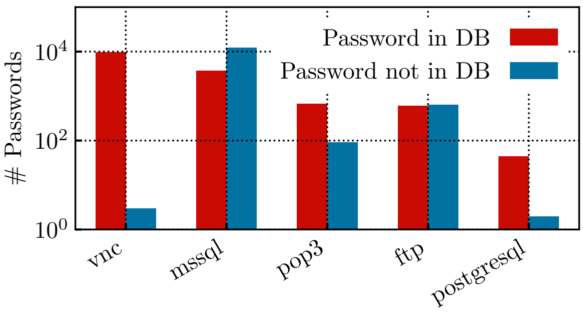
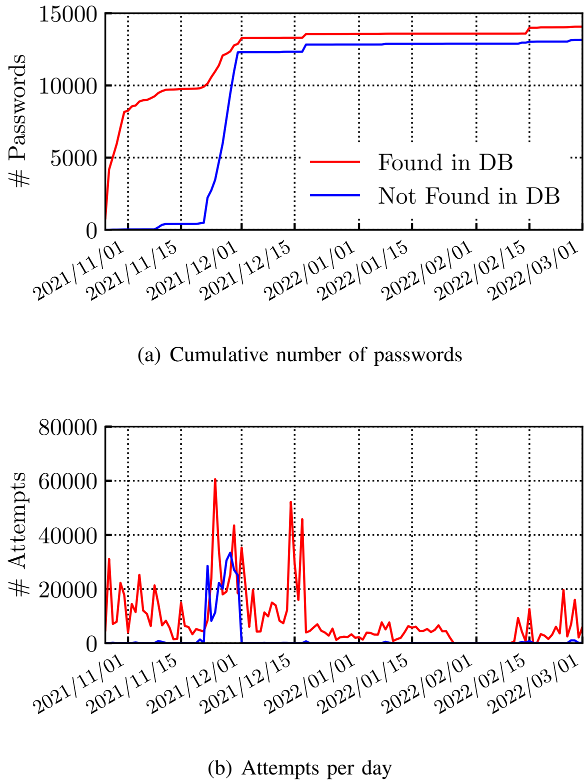


Figure 7. Unknown and known passwords for multiple honeypots.



(a) Cumulative number of passwords

(b) Attempts per day

Figure 6. Attacks with passwords from known/unknown sources.

attackers. The honeypots then save the brute-force attempts, thus allowing us to evaluate the attackers' strategies in terms of used passwords.

We first verify whether attackers rely on well-known lists of leaked passwords to perform the brute-force attacks. We compare the hashes of passwords seen in our deployment to those in the *Pwned Passwords* project [13].

Fig. 6(a) shows the cumulative number of unique passwords seen in our honeypots. The red line illustrates the number of passwords found in the *Pwned Passwords* database, while the blue line summarises the ones not found in the database.

We observe an increasing number of known passwords (i.e., present in the database) until Nov 10th, 2021. After that date, only few new entries are observed, although attackers still send traffic to the honeypots – see Fig. 6(b). In the last week of Nov 2021, more attempts are observed, this time using both new known passwords and passwords that are *not* present in the *Pwned Passwords* database.

Fig. 6(b) extends the analysis by showing the total number of attempts per day using known/unknown passwords. Initially, we record up to 30 thousand attempts per day, all of them using passwords found in the *Pwned Passwords* database. Comparing this figure with Fig. 6(a), it is clear that attackers keep repeating the same passwords over an over, thus pointing to different actors relying on the same password dictionaries. We see that even during the periods in which few new passwords are observed attackers' activity continue – compare the plateau regions in Fig. 6(a) with the same period in Fig. 6(b). The figure also reports the start of the attempts with unknown passwords at around Dec 1st, 2021. Notice how the attempts with such unknown passwords almost vanish after two weeks. We manually inspect this latter set and found that these passwords are probably automatically generated strings.

Finally, in Fig. 7 we report the number of unique known/unknown passwords for the five honeypots that save passwords. Both groups of passwords are seen in all honeypots, but in strikingly different proportions. For example, *vnc* has recorded thousands of passwords, almost all of them present in the *Pwned Passwords* database. The *mssql* and *ftp* honeypots, on the other hand, have received more unknown than known passwords overall.

Takeaway Multiple attackers rely on well-known passwords that are seen over and over in various honeypots. Some attackers rely on other lists, not present in well-known password databases. The latter is more common in some honeypots.

4.2. Origins of Brute-Force Attacks

Fig. 8 breaks down the brute-force password attempts per country. Again, we use the MaxMind's GeoIP database in this analysis. The figure shows different bars for the passwords present in the *Pwned Passwords* database (red) and those not present in the database (blue). Note the y-axis log scale, reporting the percentage of attempts.

An interesting figure emerges, which is completely different from the overall per-country traffic distribution for all honeypots, reported in Sec. 3.2.

The majority of password attempts in our dataset comes from Lithuania. The country represents 30% of the attempts with a password found in the *Pwned Passwords* database, as well as 20% of the attempts using passwords

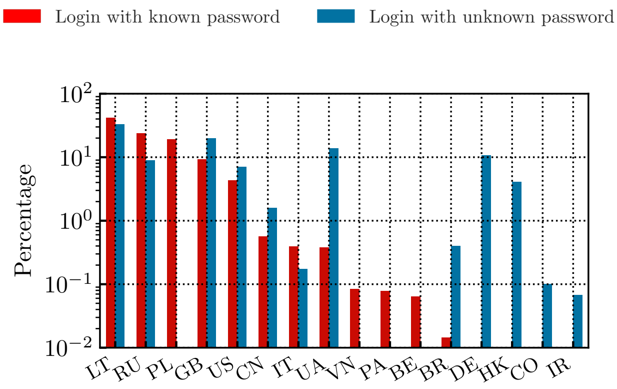


Figure 8. Percentage of password attempts per origin country.

not in the database. Recalling that Lithuania generates less than 5% of overall traffic observed in the honeypots (cfr. Tab. 3), we see that sources in this country are particularly focused on brute-force attacks.

Even more interesting is to observe that *all* brute-force attempts coming from some countries use passwords in the *Pwned Passwords* database (e.g., Poland and Vietnam). Equally, attempts coming from several other countries mainly (or exclusively) use passwords that are not available in the database, i.e., Ukraine, Brazil, Germany, Hong Kong, Colombia and Iran. This result suggests disjoints groups of attackers, with the separation among the groups already visible at country level.

Takeaway The attackers using well-known password databases are clearly distinct from those relying on other lists. This separation is visible even when considering their countries of origin.

5. Related Work

Honeypots have been deployed for cybersecurity purposes for a long time. Operating honeypots is not straightforward and poses many risks. Multiple honeypot projects exist, and previous work discusses their deployments: (i) targeting particular protocols or services [5], [6], (ii) evaluating the effectiveness of different types of honeypots [7], and (iii) presenting techniques to uncover the presence of honeypots[8], [9]. Well-established projects such as the Honeynet Project [19] and TPot [12] are a great asset for building a honeypot infrastructure, since they include services already set to collect data at reduced risks. Here we present our effort operating an infrastructure built with one of such distributions, characterising the dataset obtained with our initial deployment.

Some authors [6], [20], [21], [22] present general characterisation of honeypot traffic, focusing on the origin of attacks, targeted services, frequency of attacks etc. Most previous work is however focused on vertical deployments, looking at the activity recorded in honeypots deployed for a particular type of attack, eventually deployed in several regions and networks [23], [24].

We instead report initial data captured with multiple honeypots simultaneously. We shed light on the differences and similarities of the traffic observed in this heterogeneous setup, reporting not only patterns in terms of traffic sources, but also common activities for multiple L7 services, such as brute-force attacks across various honeypots [25], [26].

6. Conclusions and Future Work

We presented an initial characterisation of the data collected in our horizontal honeypots. We showed how attackers are fast in discovering and trying to abuse the infrastructure. We identified not only groups of attackers performing large-scale attacks against single services, but also those focusing on categories of services or horizontal attempts against all services. We evaluated the passwords used in brute-force login attempts and identified i) attackers relying on well-known password lists; ii) attackers with completely different sets of passwords. The latter ones usually come from different geographic places and focus on particular services.

As future work we plan to extend the infrastructure to other honeypots and locations. We will also pursue the creation of *open honeypot datasets*, which would represent an important asset for security analysts and researchers that need to understand cyber-threats and fight attacks. The creation of such open datasets comes with multiple challenges, however. For example, the privacy and security of previous victims must be preserved, as attackers abuse their information to perpetrate new attacks. Finally, we plan to use data from our horizontal honeypots to build updated profiles of active attackers, using automated methodologies to find groups of attacks showing coordinated strategies in the multiple honeypots.

References

[1] M. Nawrocki, M. Wählisch, T. Schmidt, C. Keil, and J. Schönfelder, “A Survey on Honeypot Software and Data Analysis,” *arXiv:1608.06249*, 2016.

[2] Dionaea, “Generic low interaction honeypot,” 2021. [Online]. Available: https://github.com/DinoTools/dionaea

[3] Cowrie, “Ssh/telnet honeypot,” 2021. [Online]. Available: https://github.com/cowrie/cowrie

[4] SNARE/TANNER, “Web application honeypot sensor,” 2021. [Online]. Available: http://mushmush.org/

[5] E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq, “Paying for likes? understanding facebook like fraud using honeypots,” in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC ’14, New York, NY, USA, 2014, pp. 129–136.

[6] S. Liebergeld, M. Lange, and R. Borgaonkar, “Cellpot: A concept for next generation cellular network honeypots,” *Internet Society*, pp. 1–6, 2014.

[7] W. Han, Z. Zhao, A. Doupé, and G.-J. Ahn, “Honeymix: Toward sdn-based intelligent honeynet,” in *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, ser. SDN-NFV Security’16, New York, NY, USA, 2016, p. 1–6.

[8] A. Vetterl and R. Clayton, “Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale,” in *Proceedings of the 12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD, USA: USENIX Association, 2018. [Online]. Available: https://www.usenix.org/conference/woot18/presentation/vetterl

[9] S. Morishita, T. Hoizumi, W. Ueno, R. Tanabe, C. Gañán, M. J. G. van Eeten, K. Yoshioka, and T. Matsumoto, “Detect me if you... oh wait. an internet-wide view of self-revealing honeypots,” in *Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 134–143.

[10] J. Thom, Y. Shah, and S. Sengupta, "Correlation of cyber threat intelligence data across global honeypots," in *Proceedings of the IEEE 11th Annual Computing and Communication Workshop and Conference*, ser. CCWC, pp. 0766–0772.

[11] R. Hiesgen, M. Nawrocki, A. King, A. Dainotti, T. C. Schmidt, and M. Wählisch, "Spoki: Unveiling a new wave of scanners through a reactive network telescope," *arXiv preprint arXiv:2110.05160*, 2021.

[12] TPot, "The all in one honeypot platform," 2021. [Online]. Available: https://github.com/telekom-security/tpotce

[13] Cloudflare, "Pwned passwords," Last visit March 2022. [Online]. Available: https://haveibeenpwned.com/Passwords

[14] Rdpy, "Python implementation of the microsoft rdp protocol," 2020. [Online]. Available: https://github.com/citronneur/rdpy

[15] Mailhoney, "Smtp low interaction honeypot," 2021. [Online]. Available: https://github.com/phin3has/mailoney

[16] Heralding, "Credentials catching honeypot," 2021. [Online]. Available: https://github.com/johnnykv/heralding

[17] L. Gioacchini, L. Vassio, M. Mellia, I. Drago, Z. B. Houidi, and D. Rossi, "Darkvec: Automatic analysis of darknet traffic with word embeddings," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 76–89. [Online]. Available: https://doi.org/10.1145/3485983.3494863

[18] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[19] Honeynet, "The honeynet project," 2021. [Online]. Available: https://www.honeynet.org/

[20] E. Alata, V. Nicomette, M. Kaâniche, M. Dacier, and M. Herrb, "Lessons learned from the deployment of a high-interaction honeypot," in *2006 Sixth European Dependable Computing Conference*. IEEE, 2006, pp. 39–46.

[21] C. F. Torres, M. Steichen, and R. State, "The art of the scam: Demystifying honeypots in ethereum smart contracts," in *28th USENIX Security Symposium (USENIX Security 19)*, Santa Clara, CA, 2019, pp. 1591–1607. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/ferreira

[22] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1666–1677, 2015.

[23] G. K. Sadasivam and C. Hota, "Scalable honeypot architecture for identifying malicious network activities," in *2015 International Conference on Emerging Information Technology and Engineering Solutions*, 2015, pp. 27–31.

[24] H. Griffioen, K. Oosthoek, P. van der Knaap, and C. Doerr, "Scan, test, execute: Adversarial tactics in amplification ddos attacks." Association for Computing Machinery, 2021.

[25] D. Fraunholz, D. Krohmer, S. D. Anton, and H. Dieter Schotten, "Investigation of cyber crime conducted by abusing weak or default passwords with a medium interaction honeypot," in *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*, 2017, pp. 1–7.

[26] D. Fraunholz, M. Zimmermann, S. D. Anton, J. Schneider, and H. Dieter Schotten, "Distributed and highly-scalable wan network attack sensing and sophisticated analysing framework based on honeypot technology," in *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*, 2017, pp. 416–421.