MikroTik Devices Landscape, Realistic Honeypots, and Automated Attack Classification

João M. Ceron
University of Twente
The Netherlands
j.m.ceron@utwente.nl

Christian Scholten
University of Twente
The Netherlands
c.p.b.scholten@student.utwente.nl

Aiko Pras
University of Twente
The Netherlands
a.pras@utwente.nl

Jair Santanna
University of Twente
The Netherlands
j.j.santanna@utwente.nl

***Abstract*—In 2018, several malware campaigns targeted and succeed to infect millions of low-cost routers (malwares e.g., VPNFilter, Navidade, and SonarDNS). These routers were used, then, for all sort of cybercrimes: from DDoS attacks to ransomware. MikroTik routers are a peculiar example of low-cost routers. These routers are used to provide both last mile access to home users and are used in core network infrastructure. Half of the core routers used in one of the biggest Internet exchanges in the world are MikroTik devices. The problem is that vulnerable firmwares (RouterOS) used in home-users houses are also used in core networks. In this paper, we are the first to quantify the problem that infecting MikroTik devices would pose to the Internet. Based on more than 4 TB of data, we reveal more than 4 million MikroTik devices in the world. Then, we propose an easy-to-deploy MikroTik honeypot and collect more than 17 millions packets, in 45 days, from sensors deployed in Australia, Brazil, China, India, Netherlands, and the United States. Finally, we use the collected data from our honeypots to automatically classify and assess attacks tailored to MikroTik devices. All our source-codes and analysis are publicly available. We believe that our honeypots and our findings in this paper foster security improvements in MikroTik devices worldwide.**

***Index Terms*—low-cost routers, MikroTik, RouterOS, honeypot, security, vulnerabilities, hacker attacks**

I. Introduction

Network infrastructure devices have been actively exploited by cyber actors [1, 2]. A variety of attacks can be deployed abusing such devices. In 2018, more than half a million low-cost routers were infected by the VPNFilter malware [3]. For disrupting this malware campaign, The Federal Bureau of Investigation (FBI) [4] reacted issuing an urgent request for users to reboot their routers. In the same year, there were several other campaigns aiming at low-cost routers (e.g., GhostDNS malware, Navidade, and SonarDNS) [5]. The most worrying news is that in some countries, for example Brazil, low-cost routers are used to provide last mile access, while also being used in core network infrastructure.

Half of the core routers used in one of the biggest Internet exchanges in the world (*i.e.*, connecting 1467 autonomous systems) [6] are manufactured by MikroTik. This manufacturer uses the same operational system (*i.e.*, RouterOS) for all their low-cost routers (used by home-users and in the core network infrastructure). To improve the security of these type of routers and set proper defences, it is crucial to understand the risk and characteristics of attacks targeting these routers first.

978-1-7281-4973-8/20/$31.00

An effective way to investigate attacks targeting devices connected to the Internet is by using a *honeypot*. As demonstrated by Lobato et al. [7], honeypots are valuable resources in detecting new or unknown attacks targeting a system. The network management and operations community have been using honeypots and similar approaches to get an insight of malicious activities inside the network [8, 9, 10, 11]. In this paper, we propose a honeypot of MikroTik devices aiming to assess attacks and risks in the worldwide landscape. Although we focus our analysis in MikroTik routers, any other low-cost router could be used. The main contributions of this paper are the following.

- **We reveal the landscape of MikroTik devices worldwide.** For achieving this, we investigate (1) how many MikroTik devices are reachable via the Internet, (2) what are the most common open port numbers in these devices, and (3) where these devices are located. This contribution highlights the importance of the investigation of MikroTik devices and it also facilitates a better definition of a realistic MikroTik device honeypot;
- **We propose a realistic, easy to deploy honeypot that mimics low-cost MikroTik routers.** The proposed honeypot uses virtualization to run the system in the cloud and it enables remote management and mechanisms to implement security, by using a set of modules. Our honeypot image is publicly available on the project website [12]. Based on the previous contribution (MikroTik devices landscape), we are able to define and deploy such realistic honeypot, *i.e.*, (1) where to place the sensors, (2) which port numbers should be open for interacting with attackers, and (3) which ethical issues are related to our design;
- **We propose an automated classification of the traffic collected at the honeypot and discuss ways for mitigating the collected attacks.** This classification facilitates the quantification of the attacks and it is based on two databases with manually created signatures. The signatures developed in this work are publicly available on the project website and are compatible with the Berkeley Packet Filter. Finally, we discuss how to mitigate the attacks on MikroTik devices.

For the landscape of MikroTik devices we used more than 4 TB of data (from Shodan.io). For the classification of attacks we used more than 17 millions packets and 1.5 million log records. The remaining part of this paper is structured as follows. First, in Section II, we describe the uniqueness of this work compared to the state-of-the-art. Then, in Section III, we describe the methodology and the results on investigating the landscape of MikroTik devices worldwide. In Section IV, we describe the design, implementation, and limitations of our MikroTik device honeypot. In Section V, we first discuss how we automatically classify the traffic collected in the honeypot. Then, we present and discuss our findings based on 45 days of collected data. Finally, in Section VI, we discuss our conclusions and future directions.

II. Related Work

Honeypots are usually designed to improve the security of systems based on learning ill-intentioned behaviours of attackers. Do Carmo et al. [8] and Nassar et al. [9], for example, described the use of honeypots for improving the security of VoIP systems. There are some other works that, although not proposing a specific honeypot itself, aimed at detecting malicious behaviour based on network measurements. For example, François et al. [10] analysed the communication behaviour patterns to infer potential botnet activities; and Sperotto et al. [11] showed how flow-based techniques can be used to detect scans, worms, botnets and denial of service attacks. Our work is similar to these works, however, we focus our attention on low-cost devices that have (also) been used in the core infrastructure of networks. These type of devices play a crucial role in the network of development countries (described in the next section).

There are some works on analysing vulnerabilities in generic low-cost routers and proposing countermeasures. For example, Niemietz and Schwenk [13] evaluated home routers and showed how these routers are vulnerable to cross-site scripting attacks and User-Interface (UI) redressing; Mujtaba and Nanda [14] analysed vulnerabilities in the BGP protocol on low-cost routers; Ghourabi et al. [15] discussed the abuse on another routing protocols used by low-cost routers, i.e., on the RIP protocol. While these works focus on generic low-cost routers, only Mazdadi et al. [16] investigated MikroTik devices. This last research was limited to analyze a single attack and did not propose a reproducible way on monitoring and characterizing attacks.

Baines [17] has the most similar work to ours. He provides insights on the landscape of MikroTik devices worldwide and investigating its vulnerabilities. The main difference between his work and ours is that his entire observations are based in an active scanning on port number 8291. We, instead, first analyze the ports that actual MikroTik devices use. Then we use these ports to create a realistic MikroTik honeypot. Therefore, we have a more complete understanding of the landscape and the number of vulnerabilities. Besides we classify more attacks than in Baines [17] work.

Finally, and very important for our work, investigation of honeypots usually discusses the ethical and legal perspectives of using it for research. Sokol and Andrejko [18] discussed the issue of liability in using honeypots. This problem arises when honeypots are exploited by attackers and used to launch attacks. The paper discusses the systems that need to be taken into account when designing a honeypot to minimize the risks. Similarly, Hecker et al. [19] argued for the use of dynamic honeypots instead of low or high-interaction honeypots. In our work we take all the lessons learned in these papers into account in the design of our honeypot to minimize issues.

III. MikroTik Devices Landscape

For investigating the landscape of MikroTik devices worldwide, we rely on the information collected by the Shodan.io project. This project port-scans the entire IPv4 address space. Although other similar projects could be used, e.g., Censys.io, Shodan collects more generic ports (giving a higher chance to port-scan specific ports of MikroTik devices). For reducing the amount of traffic generated, Shodan scans only a set of IP addresses and a set of service-ports per day. It takes around two weeks for scanning the entire IPv4 space. Shodan also tries to retrieve the response from services running on IP addresses (i.e., banner), which they usually use, for example, to classify the types of services running in the device (i.e., product).

Initially, we performed a couple of queries at Shodan's online platform, i.e., "mikrotik" (1,657,859 results), "product:mikrotik" (3,700,193 results), and "product:mikrotiksmb" (1,323 results). Each result is related to an IP address and an open service-port. It means that each product/service-port running at an IP address is one single entry in Shodan's dataset. For getting the precise number of devices we would need to merge all the results related to MikroTik. After noticing that we were unable to download all results (more than 1M), for further analysis, we contacted the maintainers of Shodan, who granted us access to one month of their dataset.

Dataset and Methodology. Each day of Shodan's scanning is a file with size around 130 GB. We download the data from 17/07/2019 until 17/08/2019. In total more than 4 TB of data had been retrieved. We filtered the entire dataset by records that contained the string 'mikrotik'. This filter implies that the banner (response from a device) is certainly related to a MikroTik device (true positive).

Limitations. First, if a banner of MikroTik devices is empty, neither Shodan nor us are able to classify it, possibly implying in false negative. Secondly, Shodan does not scan all possible service-port, implying in devices that will not be found explicitly related to a specific port. Thirdly, although Shodan updates their dataset every two weeks (accordingly to the owner of the project), we investigate the total number of devices in one month without flushing the data. The implication for our decision is that we do not remove IP addresses that are potentially not pointing to MikroTik devices anymore. Finally, we expect that a small number of devices are not discoverable by Shodan because they are, for example, behind NAT, or do

not answer to any service-port open, or they were not online at the moment of the measurement.

Our observations regarding the number of records and the number of IP address in each day are presented in Figure 1. After one month we observed 4,742,944 distinct IP addresses and 6,484,420 distinct records related to MikroTik devices. We observed that every day, both the number of records and the number of IP addresses scanned/found (blue bars) were similar, ∼500k per day (median 535,260 and 510,173, respectively).

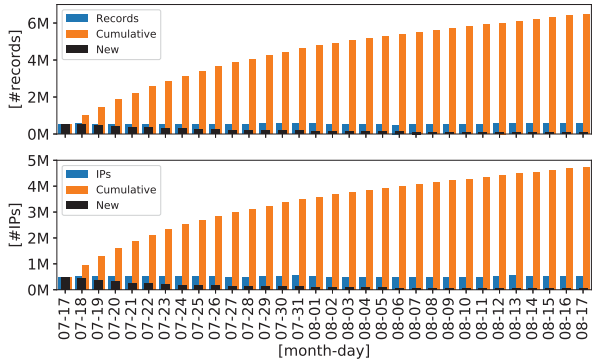

Fig. 1. Cumulative number of records (up) and IP addresses (down) related to MikroTik devices, overtime.

In Figure 1, we also depict the cumulative number of distinct records (graph up) and IP addresses (graph down); and the number of *new* records found in a day (related to the cumulative) (black bars). Differently than we expected, the number of **new** records and IP address (related to the cumulative) is never zero. We consider the fact that not 'flushing' our observations is the cause for this. After an interval of two weeks both the number of *new* records and the number of new IP address seems to get stable (as advised by Shodan's owner). We consider these numbers as the potential turnover of MikroTik devices.

In Table I, we summarize the top 10 discovered ports related to MikroTik devices. A first observation is that HTTP (line 3, 6, 8 and 9) runs on different port numbers, possibly for preventing malicious access. Usually, the management of a MikroTik router is performed using the HTTP(S) protocol. A second observation is that most of the found ports are listed as services running by default in MikroTik devices (i.e. FTP–21, Telnet–23, SSH–22, HTTP–80, HTTPS–443, Bandwidth-Test-Server–2000, Winbox–8291, API–8728, and API-SSL–8729) [20]. However, one of the top ports, 1723, does not run by default. Intentionally, this port was made open by their operators or hackers that got access to the device.

We also noticed in Table I that there are **no** entries retrieved by Shodan related to port 8291 (Winbox), 8728 (API) and 8729 (API-SSL). The reason is that Shodan does not scan for those ports. The implication is that the number of devices that we found are potentially smaller than the actual number of MikroTik devices. However, according to Baines [17], there are at least 565,648 devices (IP addresses) running port

TABLE I
TOP 10 SERVICE-PORTS OPEN ON MIKROTIK DEVICES.

	Port #	Description	# Records	%
1	2000	Bandwidth-test	3,769,843	58.1%
2	1723	PPTP	1,265,191	19.5%
3	80	HTTP	410,289	6.3%
4	21	FTP	311,952	4.8%
5	23	Telnet	164,330	2.5%
6	8080	HTTP	139,277	2.1%
7	161	SNMP	91,453	1.4%
8	8888	HTTP	41,233	0.6%
9	81	HTTP	36,292	0.6%
10	22	SSH	28,705	0.4%

8291 (Winbox), which is mainly used by MikroTik devices. Comparing their findings, available at [21], with our one month analysis, we still observed more than 80% of overlap (80.17%). This means that although Shodan does not cover port 8291, the majority MikroTik devices run multiple ports and Shodan is able to identify them.

Finally, in Table II, we summarize the top 10 countries with most IP addresses related to MikroTik devices. This information is important for deciding in which countries our honeypot sensors (described in the next section) should be placed. Those ten countries represent 64.88% of all MikroTik devices that we found. Note, also that 8 out of 10 countries are considered 'emerging economies' (*). A possible reason for this is that MikroTik devices are known as low-cost routers. Therefore it is more interesting to 'emerging economies' to make a smaller investment in their network infrastructure. To sustain this argument, note that the top 1 country is Brazil and according to [6] half of the core routers in the largest Brazilian Internet Exchange are MikroTik devices.

The takeaway message. In this section, we observed that there are a large number of MikroTik devices in the world, largely located in emerging economies, some of them used in the core network infrastructure of countries.

This finding highlights the importance of the study in this paper. Besides the default service-ports open on MikroTik devices, for mimicking such device, a MikroTik-honeypot

TABLE II
TOP 10 COUNTRIES WITH IP ADDRESSES RELATED TO MIKROTIK DEVICES

	Country	# IP add.	%
1	BRA*	759,770	16.0%
2	CHN*	715,325	15.1%
3	USA	272,470	5.7%
4	RUS*	260,553	5.5%
5	IDN*	239,598	5.1%
6	ITA	207,229	4.4%
7	IRN*	197,756	4.2%
8	IND*	153,757	3.2%
9	THA*	137,036	2.9%
10	ZAF*	134,124	2.8%

must also consider the top 10 most open ports described in Table I.

IV. System design

In this section, we describe the design of the proposed honeypot, present deployed aspects and discuss the legal considerations.

A. Honeypot design

To enable the honeypot for reproducibility and easy deployment, we used a system based on paravirtualization. As depicted in Figure 2, it defines a host system based on Linux that is responsible for running the RouterOS. We opted to use the version 6.39.3 of RouterOS since this version has not been patched for most of the critical vulnerabilities from recent years. The communication between the host system and RouterOS is performed by software API developed using three building blocks described in sequence.

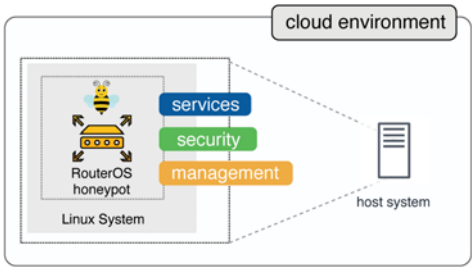

Fig. 2. MikroTik honeypot design.

Services: the RouterOS system has been configured to provide a set of service externally. Those services were deployed in the honeypot and exposed through the host system. As a consequence, the host access policy has been updated to enable connection to those services directly. Despite RouterOS enables a set of services by default, we have customized the system configuration to turn the device more realistic according to the finding discussed in Section III. This customization has added 4 (*) services on the default configuration setup, see Table III for a list of enabled services in our honeypot.

Security: the security module is responsible for controlling the access to the RouterOS and manage the connections to the running services. An emulated virtual private network was created by running the virtual machine in a host-only network to separate the traffic to the server and the honeypot. This ensures that the honeypot will not be able to gain access to the rest of the server. The honeypot is a high-interaction honeypot. This has the advantage that the chances of receiving and detecting attacks are larger than with low-interaction honeypots. The disadvantage is that more damage could be done to the device and therefore proper security measures should be taken to ensure that the router can be reset easily and that the bandwidth is limited to significantly limit the damages attackers can do. For this reason, rate limiting has been used to limit all traffic to the testbed to 1 Mbps. Furthermore, the router is restored to the original state every day at midnight to prevent abuse.

Management: our system was designed to collect all the traffic inbound and outbound to the honeypot. We have collected the full packets using the TCPDUMP capture tool for each running sensor and stored that information in a centralized server. Apart from the network traffic, we also collect the system events (logs) available in the running RouterOS. As described by Mazdadi et al. [16], it is possible to use the interface provided by RouterOS to actively retrieve relevant system information, such as: DHCP leases, configuration changes, uploaded files, BGP data and more. The tool developed in this paper was not available, so a custom script was written to imitate this functionality [12]. Our tool has been configured to collect the system events every 5 minutes via the RouterOS API-SSL service on port 8729.

B. Honeypots deployment

A total of six honeypots were created in different regions of servers of the Google Cloud Platform. The locations of the honeypots were chosen based on the most common regions for MikroTik devices based on Table II. Another factor for the placement is that the honeypots should cover most of the continents. With this in mind, the regions we settled on were: Australia, Brazil, China (Hong Kong), India, the Netherlands and the United States of America.

A central 'collector' computer was used to collect the traffic from the six honeypots. This computer implements routines to collect all the new logs generated by TCPDUMP capture tool as well as the system events from the honeypot. The information gathering process is done every hour, so even if the honeypot is compromised and we lose the access, we still can get the last our of logs. It is important to note that in this case, our honeypot will be restored using our clean system snapshot.

TABLE III
RouterOS services enabled in our honeypots.

#	Port #	Service
1	2000	Bandwidth-test
2	1723	PPTP*
3	80	HTTP
4	21	FTP
5	23	Telnet
6	8080	HTTP-Proxy*
7	22	SSH
8	139	SMB*
9	445	SMB*
10	8291	WinBox
11	8728	API
12	8729	API-SSL

C. Legal considerations

Legal obligations are an important factor to consider in the design of the honeypot. According to EU law, "a duty to act positively to protect others from damage may exist if the actor creates or controls a dangerous situation" [22]. This law gives honeypot owners a responsibility to take proper actions to secure the honeypot, since a honeypot can be seen as a potentially dangerous situation by attracting real world attacks.

Research from Sokol and Andrejko [18] shows that a secure honeypot meeting the requirements laid down by EU law consists of the five parts mentioned below, which have been used as a guideline for the design of the honeypot for this research:

- Firewall. Only allow connections to restricted ports;
- Dynamic connection redirection mechanism. Only trusted connections can have access outside the honeypot;
- Emulated private virtual network. The honeypot should be run in a restricted private network to restrict attackers;
- Testbed. The controlled environment to analyse vulnerabilities in applications;
- Control center. The administrator of the honeypot should monitor connections and quickly respond to incidents.

Limitations. As discussed in this section, our honeypots were placed in a cloud infrastructure (i.e., Google Cloud). Some attackers may avoid well-known address IP range such ones from cloud provider. Some attacker could also avoid our honeypots because they are not actual routers (not providing last-mile access).

V. Attack Classification, Findings, and Discussion

In this section, we first present our methodology to automate the classification of attacks on MikroTik devices. Then, we present our observations over almost two months of data collection.

A. Attack Classification Methodology

Our methodology for classifying attacks relies on comparing the collected data with manually created signatures. We created two databases of attack containing: (1) signatures that cover the majority of Common Vulnerabilities and Exposures (CVE) targeting MikroTik devices (listed by CVE [23]); and (2) signatures of known attacks that target low-cost routers in general.

For our attack classification, we consider two types of input files collected by our MikroTik honeypots: (1) packet-based network traces (*.pcap* files) and (2) logs. While *.pcap* files are used for CVE-MikroTik-related attacks, logs are used for classifying generic attacks. Signatures generated from *.pcap* data were made compatible with Berkeley Packet Filter and are publicly available on the project website. All the developed signatures were validated using the prove-of-concept vulnerability traffic by Tenable Research [24], which is the most known dataset for validating attacks on MikroTik devices.

For a complete usage of the *.pcap* files, we decrypt eventual encrypted MikroTik proprietary protocol. For example, packets with Winbox protocol were decrypted using the tool proposed by Tenable Research [25] and eventual encrypted Web traffic was decrypted using the tool proposed by Tenable Research [26]. Figure 3 depicts the entire process of our methodology.

Dataset. We collected 45 days of data (.pcap and logs), from 23/07/2019 to 5/09/2019, which targetted our 6 honeypots (in Australia, Brazil, China, India, Netherlands, and United States of America). In total, we collected more than 17 millions of packets (5 millions of flows) and 1.5 million of log records. The authors would be glad to share the entire dataset upon request, for researching purpose.

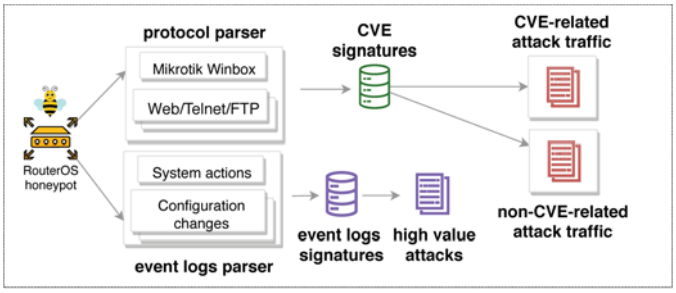

Fig. 3. Automated attack classification process.

Limitations. It is important to highlight that we do **not** focus our classification on the overall backscatter traffic collected by our honeypot. The majority of related work already covers this type of analysis. Instead, we analyse a small and most important portion of the entire dataset, which is mainly related to attacks tailored to MikroTik devices. Another limitation is that we were able to generate signatures only for half of the known CVEs. The implication is that we cover only a partial picture of the attacks tailored to MikroTik devices. Also, more sophisticated attacks may be missed.

B. MikroTik CVE-Related Attacks

We have identified 5371 attacks related to CVEs of MikroTik devices. These attacks represent only 0.03% of the total number of packets collected. Although these attacks represent a very small percentage of the total traffic, these attacks were tailored to MikroTik devices. It means that these attacks would have produced more damage to the real devices and its users than generic attacks.

The most popular attack vector that we observed was 'directory transversal' attacks (related to CVE-2018-14847 and CVE-2019-3943). These types of attacks enable attackers to access restricted files and directories within the router. In total, our honeypots observed 4928 attacks exploiting these vulnerabilities (92% of the total). By investigating the payload of these attacks, we observed that 38% aimed to change the system job scheduled for executing commands. The remaining 62% successfully acquired the credentials of administrator accounts. In the next subsection we focus on the successful login attempts after attackers exploited these vulnerabilities (CVE-2018-14847 and CVE-2019-3943).

Although some vulnerabilities were mapped to signatures, we did not observed any attack trying to perform remote code execution. For example, MikroTik RouterOS has two vulnerabilities that enable attackers to run arbitrary code or commands in the system (i.e., CVE-2018-7445 and CVE-2018-1156). These vulnerabilities affect the protocol NETBIOS and an specific service module triggered by an particular request to port 80/TCP.

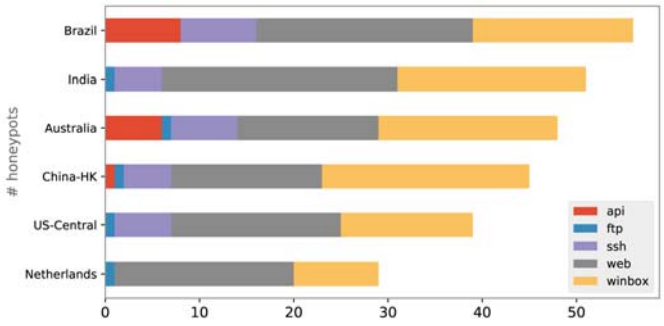

Fig. 4. The number of successful login attempts in our honeypots.

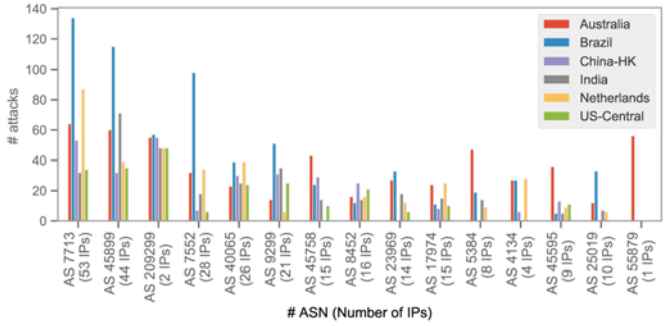

Fig. 5. The number of attacks trying to abuse well-known vulnerabilities associated with MikroTik devices group by origin (AS).

C. Successful Login

Although we used a strong and not easy to guess password (16 random letters and digits, with capital and non-capital letters), we observed several successful logins, depicted in Figure 4. In this Figure we show, per honeypot, the distribution of service with successful login (i.e., API, FTP, SSH, Web/HTTP, and Winbox). In contrast to these successful logins, we did not observed exhaustive attempts to log in the honeypots (brute force attacks).

The logins using Winbox and HTTP interface were preferred in our six honeypots. The most plausible explanation for this observation is that these two protocols are the easiest for managing MikroTik devices (via visual interface). Targeted attacks usually use automated tools to connect to the system and establish a channel with the attacker. Most of these tools, for MikroTik devices are based on Winbox and Web API (HTTP). Attempts using FTP and SSH protocols provide a more restrictive interface to manage RouterOS services when compared with MikroTik protocol (e.g., Winbox).

From Figure 4 we also observe that the distribution of attacks are not uniform compared to all honeypots. For example, there were no login attempts using SSH in the Netherlands, while in Brazil and Australia this method was largely used. This heterogeneous distribution of authentication methods suggests multiple vectors of attacks, resulting in different automated tools of attacker *modus operandi*. In the next subsection we investigate these vectors of attacks.

Attributing Successful Logins: Figure 5 we depict the top 15 Autonomous Systems that we observed successfully attempting to log in into our honeypots. The honeypot hosted in Brazil is the one that received most targeted attacks. This is interesting since Brazil is the country that has more discoverable MikroTik devices, as pointed in Section III.

However, by investigating the origin of the attacks it is not possible to see any bias regarding the IP origin. For example, IP addresses located in Brazil perform attacks in all other honeypots with similar distribution. From the 613 unique IP addresses we identified that 96% of them have contacted only one honeypot.

In AS level, Figure 5, the attacks are quite distributed as well. We have not detected an outlier AS that could reveal a *bullet prof* or any service model that may offer attacks tailored to MikroTik. For example, The top1 AS (AS 7713: Telekomunikasi Indonesia), is a large ISP in Indonesia and the top2 AS (AS 45899: Viet Nam) is another large ISP, located in Vietnam.

D. Successful Tunnel Creation

Till this point, we observed that successful login-related attacks (Subsection V-C) occurred after the exploitation of known CVEs (Subsection V-B). Following this chain of attacks, we also observed traffic tunnelling-related attacks. The usage of tunnels is a well-known technique in the context of attacks against routers. This type of attack aims to redirect, intercept or deny network traffic from/to routers. In our honeypots we observed the most common tunnelling protocols, Point-to-Point Tunneling Protocol (PPTP) and Simple Service Discovery Protocol (SSDP) have a similar abnormal *modus operandi*. The attackers first sign into the system via the WinBox protocol using the correct credentials and then set up a tunnelled connection to the outside end-point for exporting traffic. We have identified 1112 successfully established tunnels, 999 of these using the protocol PPTP and 113 using SSTP. In the next subsection we discuss the attribution of these types of attacks in detail.

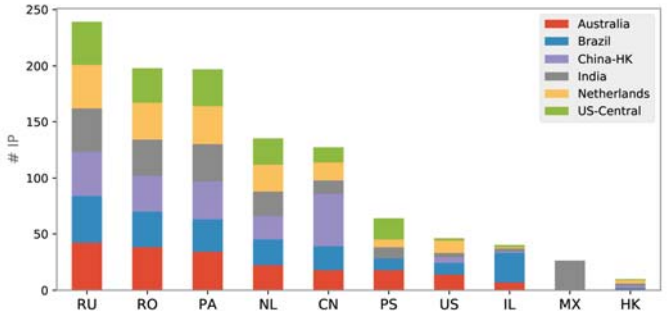

Fig. 6. PPTP/SSDP tunnel endpoint established.

Attributing Successful Tunnel Creation: In Figure 6, we depict the distribution of IP addresses related to the country code, which a tunnel was created towards. We observe that IP addresses located in Russia, Panama, Romania, Netherlands, and China represent more than 80% of all IP address exploiting traffic tunnels. Towards investigating the correlation between

the endpoints, in Figure 7, we present the number of times the top 13 IP address were used as tunnel endpoint.

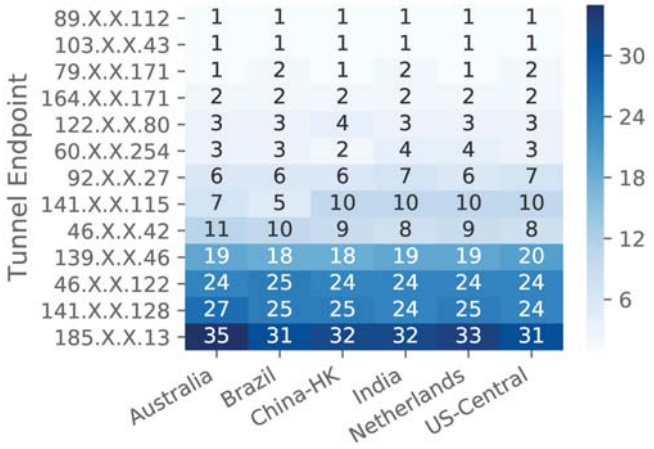

Fig. 7. Endpoint and honeypot correlation.

Surprisingly, 5 IP addresses are responsible for 58.30% of the total number of tunnels. The IP address 185.X.X.13 has been used as endpoint in all the honeypots and was observed being created in 30 days of analysis (out of 45 days). Important to remember that our honeypot was rebuilt/re-initiated every day forcing attacker to compromise the system again to establish a tunnel. We anonymized the two middle octets of the top IP addresses for avoiding ethical/legal issues.

Different from target attacks that exploit vulnerabilities, the tunnel endpoints are concentrated in a set of IPs. IPs allocated in some countries (Figure 6) and endpoints (Figure 7) are more likely to connect in all the honeypots.

E. Brute-Force Attacks and Mirai Botnet

Besides attacks tailored to MikroTik devices and attacks that were followed by the former, we observed other types of attacks. Brute force attacks is one of them. We identified a total of 48313 unique IP addresses performing such attacks. For classifying this type of attacks Zeek tool (previously called BRO) was used. MikroTik devices do not implement protection mechanism against brute-force attacks. Considering the number of services that enabling remote management, MikroTik devices are an attractive target. In the traffic that we have collected at our honeypots, the service Telnet (23/TCP) was the most targeted by brute-force attackers (89% of the attacks).

A large subset (85%) of brute-force attacks on Telnet used a known signature of the Mirai botnet [27]. Mirai is an IoT malware that tries to compromise devices using brute-force attacks and turn them in bots. We observed 39449 unique IP addresses that were part of a Mirai botnet. As reported by Ceron et al. [28], in 2018 there were around 200.000 MikroTik devices compromised and part of Mirai botnet.

F. Log-Based Analysis for Validating our Observations

Aiming at validating the attacks found in the previous sections (based on packet-based measurement) and eventually finding other attacks, in this section we analyse the content of the system event logs of our honeypots. Table IV summarizes some of our findings.

TABLE IV
SYSTEM ACTIONS PERFORMED BY ATTACKERS.

#	Log-record Partial Content	Count
1	”New script scheduled by admin”	87
2	”PPTP Server settings changed”	5
3	”SSTP Server settings changed”	5
4	”NAT rule added”	5
5	”SSTP Server settings changed by admin”	2
6	”Pool PPTP added by admin”	2
7	”Traffic logger configuration changed by admin”	1

The first observation is that the log files are unstructured. This fact turns the pattern identification process challenging. Still, from Table IV we observe that lines 2, 3, 5 and 6 emphasise changes in the tunnelling (discussed in Subsection V-D). Similar to the tunnelling, in line 4, attackers changed firewall rules. The most common event in the system log (Line 1), however, are attackers adding a script (crontab) for further controlling the MikroTik device. The most remarkable is that one attacker removed some records of the logs to hide his/her activities. Note that all the events listed in Table IV happened after a successful login (described in Subsection V-C), which was only possible because attackers exploited known MikroTik CVEs (described in Subsection V-B)

G. Overall Discussion

Our classification methodology has proven valuable to identify and classify attacks tailored to MikroTik devices. In the same way that we used signatures to classify attacks based on offline data, signatures could be use for active blocking those attacks. We observed that attacks targeting this type of devices follows a chain of problems. All problems begin with MikroTik RouterOS being left outdated. Since 2011, MikroTik has released a total of 129 stable versions of RouterOS. In the last version (version 6.44.5) a total of 6 security fixes were deployed.

All the current 16 CVEs could be easily solved by just updating the RouterOS version. Firstly, This act would, for example, mitigate revealing administrator credentials, consequently mitigating successful logins, consequently mitigating the creating of traffic tunnels, and so on.

Although this is a simple solution, there are thousands of MikroTik devices with old RouterOS versions. We estimate that more than two million MikroTik devices are still vulnerable to the majority of CVEs. We hope that the findings in this paper foster the improvement of the MikroTik devices security status.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we explored the attacks targeting MikroTik devices. To investigate those attacks we proposed an automated-classification method based on network signatures and system

logs events. To improve the quality of the results we (1) investigated the landscape of MikroTik devices worldwide using 30 days of data from an Internet scanner service (shodan.io); (2) designed a realistic honeypot that mimics the characteristics from MikroTik RouterOS discoverable over the Internet.

To validate our approach, we have deployed 6 honeypots and placed them in countries where those devices are popular according to our findings. By using the honeypots for 45 days we have collected more than 17 Millions of packets originated from 181.581 unique IP addresses. Using this dataset, we have evaluated our proposed classification methodology and discovered targeted attacks. In total, we have identified 5371 attempts trying to exploit well-known vulnerabilities and more than 50.000 IPs performing brute-force attacks on our honeypots.

The majority of vulnerabilities exploitations are aiming to retrieve the credentials and then using these credentials to manage the system remotely. We have observed different actions on the system performed by the attackers that include changes in the packet filter configuration and system event logs. For the authors, one of the most interesting findings is the creation of IP tunnelling on the compromised devices. They use these tunnels tp redirect traffic and secretly monitor/inspect/manipulated the data. We have identified more than 1000 successfully established tunnels on the honeypots.

The most common protocol detected in our honeypot was PPTP (173/TCP) a service that is not activated by default on MikroTik devices. Surprisingly, port 173/TCP, as revealed by our investigation using data from shodan.io is the second most common port on the MikroTik worldwide (984349 devices). This fact suggests either most of the administrators are manually activating PPTP service or those devices were compromised. This suspect requires further investigation.

In this research, only attacks on low-cost routers from MikroTik were analyzed. In future research, honeypots simulating other brands of low-cost routers could be used to discover if there are differences in the characteristics of attacks between multiple vendors. Further investigation should consider to place honeypots primarily on an access network where a different class of attacks might be mapped.

Acknowledgements

This research was supported, in parts, by EC H2020 GA 830927 (CONCORDIA project) and by SIDNfonds 174058 (DDoSDB project). A special thanks to John Matherly, from Shodan.io, who promptly provided us with a unique dataset.

References

[1] US-CERT. Russian state-sponsored cyber actors targeting network infrastructure devices. (ta18-106a)). https://www.us-cert.gov/ncas/alerts/TA18-106A, August 2018. Accessed on 20 September 2019.

[2] US-CERT. The increasing threat to network infrastructure devices and recommended mitigations. https://www.us-cert.gov/ncas/alerts/TA16-250A, August 2016. Accessed on 20 September 2019.

[3] Norton Symantec. VPNFilter malware now targeting even more router brands. How to check if you're affected.

[4] The Federal Bureau of Investigation (FBI). Foreign cyber actors target home and office routers and networked devices worldwide. https://www.ic3.gov/media/2018/180525.aspx, 2018.

[5] Catalin Cimpanu. Brazil is at the forefront of a new type of router attack. https://www.zdnet.com/article/brazil-is-at-the-forefront-of-a-new-type-of-router-attack/, 2019.

[6] CERT.br. New national initiatives. https://www.cert.br/docs/palestras/certbr-natcsirts2019.pdf, 2019.

[7] Antonio Gonzalez Pastana Lobato, Martin Andreoni Lopez, Igor Jochem Sanz, Alvaro A Cardenas, Otto Carlos MB Duarte, and Guy Pujolle. An adaptive real-time architecture for zero-day threat detection. In *2018 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2018.

[8] Rodrigo Do Carmo, Mohamed Nassar, and Olivier Festor. Artemisa: An open-source honeypot back-end to support security in VoIP domains. In *12th IFIP/IEEE International Symposium on Integrated Network Management*, 2011.

[9] Mohamed Nassar, Radu State, and Olivier Festor. VoIP honeypot architecture. In *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, 2007.

[10] Jérôme François, Shaonan Wang, Thomas Engel, et al. BotTrack: tracking botnets using NetFlow and PageRank. In *International Conference on Research in Networking*, 2011.

[11] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, Cristian Morariu, Aiko Pras, and Burkhard Stiller. An overview of ip flow-based intrusion detection. *IEEE communications surveys & tutorials*, 12(3):343–356, 2010.

[12] João M. Ceron. Mikrotik exposed project website. https://mikrotik-exposed.org/, 2020.

[13] Marcus Niemietz and Jörg Schwenk. Owning your home network: Router security revisited. *arXiv*, June 2015.

[14] Muhammad Mujtaba and Priyadarsi Nanda. Analysis of bgp security priyadarsi nanda. In *Proceedings of the 9th Australian Information Security Management Conference*, pages 204–214, December 2011.

[15] Abdallah Ghourabi, Tarek Abbes, and Adel Bouhoula. Honeypot router for routing protocols protection. In *2009 Fourth International Conference on Risks and Security of Internet and Systems*, pages 127–130, October 2009.

[16] Muhammad Itqan Mazdadi, Imam Riadi, and Ahmad Luthfi. Live Forensics on RouterOS using API Services to Investigate Network Attacks. *International Journal of Computer Science and Information Security*, 15:406–410, February 2017.

[17] Jacob Baines. Help me, vulnerabilities you're my only hope. https://github.com/tenable/routeros/blob/master/slides/defcon_27_cleaner_wrasse.pdf, 2019.

[18] Pavol Sokol and Maros Andrejko. Deploying honeypots

and honeynets: Issues of liability. In *International Conference on Computer Networks*, pages 92–101. Springer, May 2015.

[19] Christopher Hecker, Kara L. Nance, and Brian Hay. Dynamic honeypot construction. In *Proceedings of the 10th Colloquium for Information Systems Security Education*, pages 95–102, June 2006.

[20] Mikrotik. Mikrotik documentation–manual:ip/services. https://wiki.mikrotik.com/wiki/Manual:IP/Services, 2019.

[21] Jacob Baines. Results scanner port 8291. https://github.com/tenable/routeros/blob/master/8291_scanner/results/8291_results_06292019.zip, 2019.

[22] Bernhard A. Koch. The "principles of european tort law". In *ERFA Forum*, volume 8, pages 107–124. Springer, March 2007.

[23] Mikrotik Routeros Security Vulnerabilities. https://www.cvedetails.com/vulnerability-list/vendor_id-12508/product_id-23641/Mikrotik-Routeros.html, 2019. Accessed on 29 April 2019.

[24] Tenable Research. [R1] Mikrotik RouterOS Multiple Authenticated Vulnerabilities. https://www.tenable.com/security/research/tra-2018-21, August 2018. Accessed on 30 April 2019.

[25] Tenable Research. Mikrotik routeros winbox parser. https://github.com/tenable/routeros/tree/master/pcap_parsers/winbox_pcap_parser, April 2019. Accessed on 30 August 2019.

[26] Tenable Research. Mikrotik routeros winbox parser. https://github.com/tenable/routeros/tree/master/pcap_parsers/jsproxy_pcap_parser, April 2019. Accessed on 30 August 2019.

[27] Trend Micro. Over 200,000 mikrotik routers compromised in cryptojacking campaign, August 2018.

[28] João M. Ceron, Klaus Steding-Jessen, Cristine Hoepers, Lisandro Zambenedetti Granville, and Cíntia Borges Margi. Improving iot botnet investigation using an adaptive network layer. *Sensors*, February 2019.