Economics of Cybersecurity Block 2
Telnet scans

Calvin Hendriks - c.hendriks@student.utwente.nl
Christiaan van den Bogaard - c.h.m.vandenbogaard@student.utwente.nl
Sander Bakkum - s.bakkum@student.utwente.nl
Riccardo Colombo - r.colombo@student.utwente.nl

September 23, 2019

1 Security Issue

1.1 Description of data set

The analyzed data set consists of a collection of results of port scans performed by Shadowserver [1]. These port scans have been performed on both port 23 and port 2323, where devices commonly listen with a Telnet server. Telnet is a simple line-based plaintext protocol, commonly used to expose a command line interface to devices over the internet. Telnet is an unencrypted protocol, as opposed to for example SSH which has similar functionalities. Often, Telnet servers ask for login credentials when a connection is successfully established.

Devices with an open Telnet port are exposed to remote command execution when authorization is disabled for the device, or its login credentials are easily guessed (for example when manufacturer passwords remain unchanged). When a device is compromised, it allows an attacker with malicious intent a set of actions:

- Steal any privacy-sensitive information on the device.
- Use the device as a pivot to gain further access to the network it resides in.
- Perform DDoS-attacks from the compromised device.

1.2 Mirai botnet

Port scans on Telnet ports are for example used by the Mirai-botnet to spread to IoT-devices connected to the internet. The Mirai botnet represented one of the largest botnets in existence, accounting for over 600K infected devices in its peak moment [2]. Launched in September 2016, the infection worked in a very simple yet extremely effective way. The initial phase consists in probing

pseudo random IPv4 addresses with TCP SYN packets on port 23 and 2323 (common Telnet ports). Once a device replies to the probe, Mirai enters the second phase of the infection, effectively trying to bruteforce the login in order to establish a Telnet connection by using a credential combination selected from a pre-configured list. When the login is successful Mirai sends the IP address of the compromised device to the *report server*. At this point the second part of the infection would remotely login, get technical details on the victim and install hardware-compatible malware to take control over the device. Once control is taken over the device, it can be used as part of a vast variety of attacks.

Mirai has been used in the past to launch record breaking DDoS attacks. Examples include a 620 Gbps on security news website KrebsOnSecurity and a 1+ Tbps attack on the french webhost OHV [3]. Other notable mentions include the attack on DNS provider Dyn, which affected major websites such as Amazon, Netflix and Paypal, and the attack on Liberian ISP Lonestar Cell [4].

1.3 The security issue of vulnerable IoT devices

While DDoS attacks targeted at business can be bankrupting, attacks aimed at the public infrastructure can be disastrous for society. Unfortunately, there is a misalignment of incentives to protect these IoT devices. In this paper we take a look at a major security issue of vulnerable IoT devices; **DDoS attacks launched from IoT botnets**. To get a better understanding of the issue, we first take a look at the actors involved. Next, we will give a set of ideal metrics and metrics in practice. Finally, we will define our own set of security metrics (based on the dataset we have) to measure the issue and evaluate them.

1.3.1 Actors

- IoT owners

  The most obvious actor in this list are the owners of the devices that are compromised. Because a Mirai infection is barely noticeable for the owner, there is little incentive to fix the issue. Furthermore, recommended actions include installing patches and changing default router configurations [5], something a regular internet user is unable to do. Finally, while users are often aware of the vulnerabilities that computers and laptops may have, they are often unaware that these also reside in the newer IoT devices like drones and children's toys [6].

- Internet Service Providers (ISP)

  Even though ISPs could prevent botnets, such as Mirai, from spreading [7], they also lack the incentive to do so. While the bandwidth these botnets create is quite high, it does not compare to the data normal users generate by streaming media [8]. As long as the network does not suffer from these attacks, ISPs will not likely be resolving this issue.

- Manufacturers

  Manufacturers of the devices that make up IoT botnets could help tackle this problem. However, there is again a problem with the incentive to do so. In order to remain competitive, the bare minimum is invested in device security, leading to a massive amount of vulnerable devices on the internet.

- Governments

  To tackle the IoT botnet problem, governments could set up new laws and regulations that require manufacturers to adhere to minimum requirements. This would give the manufactures incentive they need to finally ship their products in a secure state. A recent example can be found in the UK, where minimum requirements for surveillance camera manufacturers were given by the Surveillance Camera Commision [9].

- DDoS Victims

  While there is little victims can do to prevent DDoS attacks from IoT botnets, there are a hand full of solutions to mitigate such an attack. The most used solution can be found in DDoS Protection Services (DPS). When a subscription is purchased at a DPS, all traffic to the victim is rerouted through the network of the DPS. Since this network has a much higher capacity than the network of the victim, it can handle huge amount of traffic. Next, the traffic is scrubbed from malicious traffic and the benign traffic is sent to the victim.

- Attackers

  Finally, we have the attackers (or adversaries) who create (or copy) the malware and spread it to create a botnet. IoT devices have become popular among hackers for a number of reasons. First, there is the aforementioned lack of security. Second, IoT devices are often powered on 247/, making them ideal for DDoS Attacks since they are always available. Third, IoT devices are often poorly maintained and never looked at again after installing. Finally, IoT devices almost never have a fully interactive user interface, making infections easier to go unnoticed.

The metrics described in this paper will benefit multiple actors. After measuring the metrics, it will give a better insight into the amount of vulnerable machines. This will benefit regulatory bodies and help them make better decisions. If you apply the metrics on a more local scale, it can greatly benefit system administrators and the company they work for. Although the devices in a company network will not suffer from being part of a botnet, if they are used in a major DDoS attack it can ruin the reputation. Finally, after measures have been taken based on measurements of the metrics, it will greatly benefit victims of DDoS attacks.

1.4 Data details

This research is based on a dataset of port scans performed between January 1st and August 30th 2018. For each scan the dataset features:

- **Timestamp**: Date and time of the portscan.
- **IP**: Target IP-address.
- **Protocol**: IP-protocol used in the portscan (always 'TCP').
- **Port**: Target port.
- **Hostname**: Hostname associated with the IP-address (if any).
- **Tag**: 'telnet' if port is 23, 'telnet-alt' if port is 2323.
- **ASN**: Autonomous System Number of the target network.
- **Geo**: Country in which the target IP-address resides (always 'NL', Netherlands).
- **Region**: Province in which the target IP-address resides.
- **City**: City in which the target IP-address resides.
- **NAICS**: North American Industry Classification System number.
- **SIC**: Standard Industrial Classification number.
- **Banner**: Text response received upon successfully established connection.

2 Ideal metrics for security decision makers

In order for security decision makers to say something meaningful and act to the severity of the security issue at hand, they would want to know certain statistics about the specific security issue. Depending on the level of decision making, they might want to know different metrics:

2.1 Metrics for regulatory bodies

On a global level, it would be interesting to know the amount of devices that enable telnet access within a country, region or city. If normalized against the amount of users in the same region, we can compare which regions are more vulnerable. Similarly, we can look at the origin of attacks to see which regions are more likely to launch attacks. We can see whether attacks occur mostly within the same (sub-)region or if some countries are more often used to launch attacks from. A government of a country that is often used as an attack origin could consider changing their laws on cyber criminality.

2.2 Metrics for system administrators

A system administrator would be interested in how exposed and vulnerable their systems are to attacks. Firstly, they want to know how many devices in their network are accessible through telnet. To add to that, the security of these telnet connections are considered. Are these telnet devices accessible over the internet or local network? Do they require authentication. Furthermore, they might want to consider how much information about the device is leaked upon connection.

Even when devices require authentication, one should consider if these devices' login credentials have been changed from their default configuration. The Mirai botnet for example relies heavily on logging in to devices with their default credentials. A metric should be considered that describes the strengths of set credentials and whether default passwords have been changed.

3 Metrics that exist in practice

To measure the issue of IoT Botnets capable of launching a DDoS attack, metrics have been proposed in practice. Antonakakis et al. [2] have created a few metrics to measure the size and capacity of Mirai botnet variants. First, they measure the amount of packets seen at a network telescope that has a fingerprint (almost) unique to Mirai. Before the rise of Mirai, they saw just a handful of packets with this fingerprint (by randomness). However, after a while, they observe 116 billion probes from Mirai hosts. Next, they set up honeypots that acted as vulnerable IoT devices and observed the connection attempts made from unique IP addresses and extracted the binaries the attackers tried to install. From the binaries that were collected, the domain names of the Command and Control C&C servers that bots connected to were reverse engineered and a third metric that gives insight in the size of a botnet could be measured. By checking the lookup values of the domains used for (C&C) servers, they could estimate the relative size of each Mirai botnet variation.

While the previous metric were all used to measure the size of the botnet, Antonakakis et al. also provide a metric to measure how active the botnet is used. By acting as an infected device, they measured the amount of attacks commands send by the C&C servers. Finally, they compared the IP addresses seen at the network telescope with the IP addresses observed during DDoS attacks to get an insight of the various botnet capacities.

4 Designed metrics from the data set

4.1 Access control of devices

As mentioned before, Mirai used the unprotected telnet service on the device to become infected. This was done using a list of default passwords to gain access or no login attempt at all, as there was no authentication needed. The ideal

case would be that insight is given in the amount of devices which uses default credentials, however measuring this is impossible with this data. If we were to measure whether devices have default credentials in place, we need to interact with the device. In this case, we need to run a dictionary attack on the device which is intrusion and therefore illegal. What we can measure is how many devices use authentication and how many devices block the incoming request by observing the banner. Therefore, the following metrics are computed:

- What percentage requires authentication?
- What percentage is rejected?

For both cases, all duplicate rows based on IP and Banner are dropped, as it would not give a realistic view if one IP is scanned more often.

4.1.1 What percentage requires authentication?

This metric gives insight on how many entries in the dataset require authentication. This can easily be calculated by searching the banner for occurrences of words like "login" or "password". There are 24996 unique banners which contain words like this. Figure 1 shows that most devices require some sort of authentication thus have some form of protection.

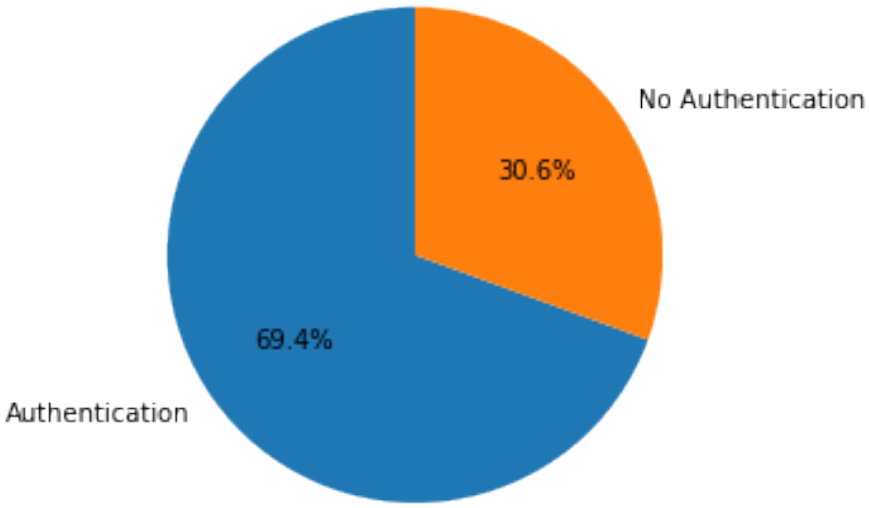

Figure 1: Pie chart of devices which require authentication

4.1.2 What percentage is rejected?

In the data there are occurrences of banners where the connection is rejected, for example because the source IP is not white-listed or it is blocked by a firewall. In this case, the scanned devices have even better protection as there is an extra line of defense in place. In order to find these cases the banners are scanned for words such as "rejected" or empty banner. We can see in Figure 2 that only a small fraction of devices having this extra line of defense in place. $122,927$ devices accept the connection which is troublesome.

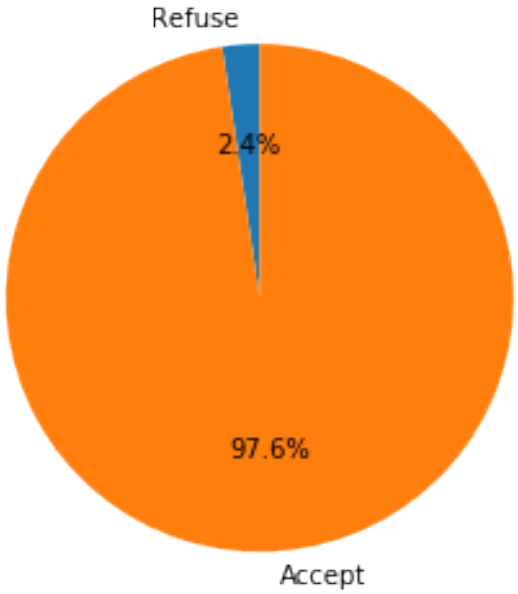

Figure 2: Pie chart of devices which refuse connection

4.2 Machines disclosing information on running software or services

This metric is meant to measure how many machines disclose any kind of information on either the software or the service running. Such information is in fact extremely useful for an attacker to understand whether and how the system can be comprimised. We define the portion P of the machines disclosing information as follows:

$$P = \frac{disclosing\_ips}{total\_ips}$$

Where $total\_ips$ represents the total number of unique IP addresses present in the whole dataset while $disclosing\_ips$ represents instead the number of unique

IP addresses which disclose any kind of information on either the software or the service running. To compute *disclosing_ips* we use a regular expression to match meaningful keywords (e.g., Linux, Raspbian, Broadband, NTP..).

4.2.1 Results

For the analyzed dataset we obtained that $total\_ips = 125,915$ and $disclosing\_ips = 76,720$. Therefore the portion of machines disclosing information on the running software or service equals to:

$$P = \frac{disclosing\_ips}{total\_ips} = \frac{76,720}{125,915} = 0.609 \simeq 61\%$$

4.3 Patch adoption related to a severe vulnerability

This metric is meant to analyze whether and how promptly devices are patched in response to the disclosure of a severe vulnerability.

In our case, we considered the devices running MikroTik RouterOS, such devices were in fact affected by a score 10 buffer overflow vulnerability reported in CVE-2018-7445 [10]. The relative patch was released on 2018-03-12[1] and the vulnerability was publicly disclosed on 2018-03-19, seven days later. Therefore, every device running an operating system prior to v6.41.3 after 2018-03-19 is considered to be vulnerable. Section 4.3.1 shows the fraction of machines which upgraded to the patched version of the OS within the six months following the disclosure of the vulnerability, section 4.3.2 delves instead deeper into analyzing the promptness of the patch adoption by considering both the days following the update release and the days following the vulnerability disclosure.

4.3.1 Fraction of machines vulnerable to CVE-2018-7445

In this sub-metric we consider the fraction P of unique IP addresses running the MikroTik OS that have upgraded to a safe version by the end of August, almost six months after the disclosure of the vulnerability. The following equation shows how we computed such fraction:

$$P = \frac{nonVuln\_ips}{total\_ips} \tag{1}$$

Where *total_ips* represents the total number of unique IP addresses running MikroTik OS, while $nonVuln\_ips$ represents instead the number of unique IP addresses running a non-vulnerable version of the OS.

Results

For the analyzed dataset we obtained that 19,166 unique IPs are running MikroTik OS. To avoid cases where an IP address is only scanned in a period that precedes the date of the disclosure, we have limited our analysis to the

[1] `https://mikrotik.com/download/changelogs`

subset of the dataset starting on 2018-03-12. Our final result is that $total\_ips = 11,510$ and $nonVuln\_ips = 519$. Therefore we can compute the portion of the machines that have upgraded their OS to a patched version within the first six months following the disclosure of the vulnerability as:

$$P = \frac{nonVuln\_ips}{total\_ips} = \frac{519}{11,510} = 0.0451 \simeq 4.5\%$$

This sub-metric highlights how only a slight portion ($< 5\%$) of the machines actually updated to the patched version of the OS, leaving the vast majority completely vulnerable to a score 10 CVE.

4.3.2 Patch adoption rate

In this sub-metric we consider the adoption rate of the patched software over two time intervals:

1. $I1$ will represent the interval spanning from the patch release date (2018-03-12) to the day before of the vulnerability disclosure (2018-03-18).
2. $I2$ will represent the interval spanning from the vulnerability disclosure date (2018-03-19) to the end of the month (2018-03-31).

For each time interval the fraction of machines which have adopted the new software will be computes as in (1).

Results

It must be noted that in order to obtain meaningful results we only focused on the IP addresses that are scanned at least once in every period. Therefore we have obtained that $total\_ips = 497$ and for each period we have that:

$$P_{I1} = \frac{nonVuln\_ips_{I1}}{total\_ips} = \frac{13}{497} = 0.0262 \simeq 3\%$$

$$P_{I2} = \frac{nonVuln\_ips_{I2}}{total\_ips} = \frac{23}{497} = 0.0462 \simeq 5\%$$

This metric shows how only $\sim 3\%$ of the machines run the OS update during the first week and how only an additional $\sim 2\%$ patched the OS in the 12 days after learning about the vulnerability.

4.4 Update status for OpenSSH

This metric is meant to analyze how recently updated the machines running OpenSSH are. In order to do this, we computed the fraction P of machines running a particular version V_x_ips over the total number $total\_ips$ of unique IPs running OpenSSH:

$$P = \frac{V_x\_ips}{total\_ips}$$

Results

Considering the dataset, we have obtained that $total\_ips = 12,149$. Moreover, figure 3 gives an overview on the fraction of machines running a specific version.

Release Date	< Aug '15	Aug '15	Aug '15	Feb '16	Aug '16	Dec '16	Mar '17	Oct '17	Apr '18
Version	<7	7.0	7.1	7.2	7.3	7.4	7.5	7.6	7.7
V	519	0	107	10970	11	439	19	65	19
P	4,3%	0%	0,9%	90,3%	0,1%	3,6%	0,1%	0,5%	0,1%

Figure 3: Fraction of machines running a specific version of OpenSSH.

The metric highlights how the vast majority of the machines is not running the most recent version of OpenSSH. Furthermore, considering CVE-2016-10012 it is possible to note how all versions previous to v7.4, $\sim$ 95,6% in our case, are vulnerable to a score 7.2 vulnerability that can allow an attacker to gain privileges and completely affect the system's availability, confidentiality and integrity [11].

5 Conclusion

In this report we looked at the security issue of DDoS Attacks launched from IoT botnets. Since these botnets are often formed by abusing vulnerabilities and weak credentials, these will be reflected in the metrics as well. First, we took a look at the access control metric. While a large part of the scanned IP addresses ($\sim$ 70%) required some form of authentication (although we could not measure the strength of such authentication), a considerable part of the devices ($\sim$ 30%) did not require any authentication at all. Moreover, the second submetric shows that only a very small fraction has some extra form of access control, for example a firewall.

Next, metrics were set up to measure what fraction of devices disclose meaningful information on the version of their OS or services running and what fraction of this was out-of-date software. Surprisingly, 61% of the IP addresses disclose information of version numbers and, when considering devices running MikroTik RouterOS, only 4.5 % of them updated to the patched version 6 months after the disclosure of CVE-2018-7445. Moreover, when analyzing how quickly the RouterOS updates were run it was possible to see that only a slight fraction updated the OS in the 7 days following the release and only an additional 2% run the update in the week following the disclosure of the vulnerability.

Finally, the last metric took into consideration the machines running OpenSSH and analyzed how recent the running version is. We concluded that the vast majority (90.3%) is running v7.2 which was released in February 2016 and therefore vulnerable to CVE-2016-10012.

References

[1] "The Shadowserver Foundation: Telnet Scanning Project." [Online]. Available: https://telnetscan.shadowserver.org/

[2] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[3] D. Goodin, "Brace yourselves—source code powering potent iot ddoses just went public," 2016. [Online]. Available: https://arstechnica.com/information-technology/2016/10/brace-yourselves-source-code-powering-potent-iot-ddoses-just-went-public/

[4] CloudFlare, "Inside the infamous mirai iot botnet: A retrospective analysis," 2017. [Online]. Available: https://blog.cloudflare.com/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/#toc-4

[5] NJCCIC, "Ddos: Internet-of-things likely to fuel more disruptive attacks," 2016. [Online]. Available: https://www.cyber.nj.gov/threat-analysis/ddos-internet-of-things-likely-to-fuel-more-disruptive-attacks

[6] Intel, "New mcafee survey reveals only 42 percent of consumers take proper security measures to protect their new gadgets," https://newsroom.intel.com/news-releases/new-mcafee-survey-reveals-42-percent-consumers-take-proper-security-measures-protect-new-gadgets/#gs.4mfvb2, 2016.

[7] O. Çetin, C. Gañán, L. Altena, T. Kasama, D. Inoue, K. Tamiya, Y. Tie, K. Yoshioka, and M. van Eeten, "Cleaning up the internet of evil things: Real-world evidence on isp and consumer efforts to remove mirai." in *NDSS*, 2019.

[8] CloudFlare, "What is the mirai botnet?" [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/

[9] SC Magazine, "Cctv configuration requirements aim to prevent another mirai botnet?" 2019. [Online]. Available: https://www.scmagazineuk.com/cctv-configuration-requirements-aim-prevent-mirai-botnet/article/1588325

[10] "Mikrotik buffer overflow vulnerability," 2018. [Online]. Available: https://www.cvedetails.com/cve/CVE-2018-7445/

[11] "Openssh overflow gain privileges," 2016. [Online]. Available: https://www.cvedetails.com/cve/CVE-2016-10012/