Fraunhofer
FKIE

FRAUNHOFER-INSTITUT FÜR KOMMUNIKATION, INFORMATIONSVERARBEITUNG UND ERGONOMIE FKIE

WHITE PAPER

HOME ROUTER SECURITY REPORT 2020

Fraunhofer
FKIE

FRAUNHOFER-INSTITUT FÜR KOMMUNIKATION, INFORMATIONSVERARBEITUNG UND ERGONOMIE, FKIE

Home Router Security Report 2020

Peter Weidenbach
Johannes vom Dorp

June 2020

Executive Summary

This report analyses 127 current routers for private use developed by seven different large vendors selling their products in Europe. An automated approach was used to check the router's most recent firmware versions for five security related aspects.
We were able to extract completely 117 of the 127 firmware images. Four firmware images could be extracted partly and six firmware images could not be extracted at all. 116 of 127 (91%) devices are powered by Linux. One was powered by ThreadX and another one by eCos.
The security aspects addressed in this report are:

- When were the devices updated last time?
- Which operating system versions are used and how many known critical vulnerabilities affect these operating system versions?
- Which exploit mitigation techniques do the vendors use? How often do they activate these techniques?
- Do the firmware images contain private cryptographic key material?
- Are there any hard-coded login credentials?

Our results are alarming. There is no router without flaws. 46 routers did not get any security update within the last year. Many routers are affected by hundreds of known vulnerabilities. Even if the routers got recent updates, many of these known vulnerabilities were not fixed. What makes matters even worse is that exploit mitigation techniques are used rarely. Some routers have easy crackable or even well known passwords that cannot be changed by the user. Most firmware images provide private cryptographic key material. This means, whatever they try to secure with a public-private crypto mechanism is not secure at all.
Nonetheless, vendors seem to prioritize security differently. Especially AVM does a better job than the other vendors regarding most of the security aspects. However, AVM routers are not flawless as well. ASUS and Netgear do a better job on some aspects than D-Link, Linksys, TP-Link and Zyxel.
To sum it up, much more effort is needed to make home routers as secure as current desktop or server systems.

Additionally, our evaluation showed that large scale automated security analysis of embedded devices is possible today. We used the the Firmware Analysis and Comparison Tool (FACT)[1] and it worked very well for almost all firmware images analyzed during this study. FACT is an open source software available on GitHub[2].

[1] `https://fkie-cad.github.io/FACT_core/`
[2] `https://github.com/fkie-cad/FACT_core`

Contents

1 Introduction

Botnets like Bashlite and Mirai have shown that vulnerable embedded devices can be a major risk to users and the internet itself. In 2016 these botnets launched a distributed denial-of-service attack (DDoS)[1] with a capacity of more than 1 terabit per second.[2] Routers are more vulnerable than other embedded device because they can be reached from the internet directly. Additionally, they normally operate 24/7. Furthermore, these devices just got more important, since the Covid-19 virus forces many people to stay at home and work from home.
In this study we looked at the security of 127 current routers for private use of seven large vendors selling their products in Europe. We used the Firmware Analysis and Comparison Tool (FACT)[3] to automatically extract and analyze the most recent firmware version of these routers. The following five security related aspects were analyzed:

- When were the devices updated last time?
- Which operating system versions are used and how many known critical vulnerabilities affect these operating system versions?
- Which exploit mitigation techniques do the vendors use? How often do they activate these techniques?
- Do the firmware images contain private cryptographic key material?
- Are there any hard-coded login credentials?

Our analysis shows alarming results. There is no router without flaws. 46 routers did not get any security update within the last year. Many routers are affected by hundreds of known vulnerabilities. Exploit mitigation techniques are used rarely, which makes matters even worse. Some routers have easy crackable or even well known hard-coded passwords.
However, there are differences between the vendors. Some of the vendors seem to care more about security than other vendors. Especially one is ahead of the others in most categories. Nevertheless, we cannot tell for sure that some vendors really do a better job all the time, because there might be false positive and false negative results regarding static analysis done by FACT. Therefore, we discuss the reliability of the results regarding security related aspects, as well.
This report is structured in the following way: In chapter 2 we give a detailed overview of our evaluation corpus and why we built the corpus the way it is. We provide some information about the CPU architectures and Operating Systems powering the devices, too. Chapter 3 presents details about the analyzed aspects and the results of our evaluation. The individual sections of this chapter include the reliability discussions as well. Finally, chapter 4 describes our conclusions.

[1] https://en.wikipedia.org/wiki/Denial-of-service_attack
[2] https://arstechnica.com/information-technology/2016/10/brace-yourselves-source-code-powering-potent-iot-ddose
[3] https://fkie-cad.github.io/FACT_core/

2 Evaluation Corpus

Our evaluation is based on the FKIE-HRS-2020 corpus[1]. The FKIE-HRS-2020 corpus provides samples of home routers from large vendors selling their products in Europe. The corpus was created on March 27th 2020. It provides the latest stable firmware versions of all routers promoted on the vendor's websites at that date. If there was more then one hardware revision of a router, we just took the firmware of the most recent hardware revision. There are some routers missing because their firmware was not provided on the vendor's website or ftp server. Therefore, Huawei is not part of the evaluation at all, since they do not provide any firmware on their website. Furthermore, we do not evaluate routers of telecommunication providers because their routers are developed and manufactured by OEMs. The providers buy from different OEMs so that each hardware revision of one router can be developed by a different OEM. Additionally, the providers generally do not develop or maintain the firmware of these devices at all.
The corpus consists of 127 firmware samples from seven vendors. The total size of the firmware samples is 3.25 GiB. A detailed list of all samples can be found in the footnote reference.

We used FACT[2] to extract the firmware images. FACT was able to extract 117 of the 127 firmware images. The AVM Fritz!Box 4020 was partly extracted. FACT failed on the file system, which seems to be proprietary squashfs format that even the FREETZ project[3] cannot handle. The Linksys EA9500 V2 was partly extracted as well. In this case FACT did not detect the file system. The Linksys E2500 V4, E5350, E5400, EA2750, EA6100 and Zyxel 6615 have a propritary format with an unknown compression or encryption method. The Zyxel NBG-418N V2 and WAP3205 V3 seem to be extracted correctly. However, we could not find any executable code in it.
We provide some more information about the corpus in the following sections.

2.1 Operating Systems

Routers are quite complex. Therefore, vendors do not develop their firmware from scratch but use third party code. In this section we take a look at the router's Operating Systems (OS). Figure 2.1 shows, which operating systems power the routers of our corpus. The most popular OS is Linux. At least 116 of 127 (91%) routers run this OS. Figure 2.1 quotes 87% because some routers provide more than one OS. ASUS Blue Cave, AVM Fritz!Box 7583, Netgear RAX40 and TP-Link Archer AX50 are powered by Linux. However, they provide a WiFi chip Firmware powered by ThreadX.
D-Link DSL-321B Z is the only router powered by ThreadX only. Zyxel NBG6615 utilizes the eCos OS. FACT detected MicroC/OS on the Netgear R8000. But, this seems to be a false positive.

2.2 CPU Architectures

The CPU architecture has nothing to do with security. Nevertheless, we got the data for free and it might help some people who do not know which assembler language to learn. As you can see in figure 2.2 ARM and MIPS are the only relevant architectures. 58 devices are powered by ARM and 58 devices are powered by MIPS as well. We do not know the architecture of 11 devices for different reasons. The D-Link DSL-321B is powered by the ThreadX Operating System and the Zyxel NBG6615 is powered by eCos. FACT's CPU detection technique does not support these operating systems. The Zyxel WAP3205 V3 and NBG-418N V2 have an unknown operating

[1] https://github.com/fkie-cad/embedded-evaluation-corpus/blob/master/2020/FKIE-HRS-2020.md
[2] https://github.com/fkie-cad/FACT_core
[3] https://freetz.github.io/

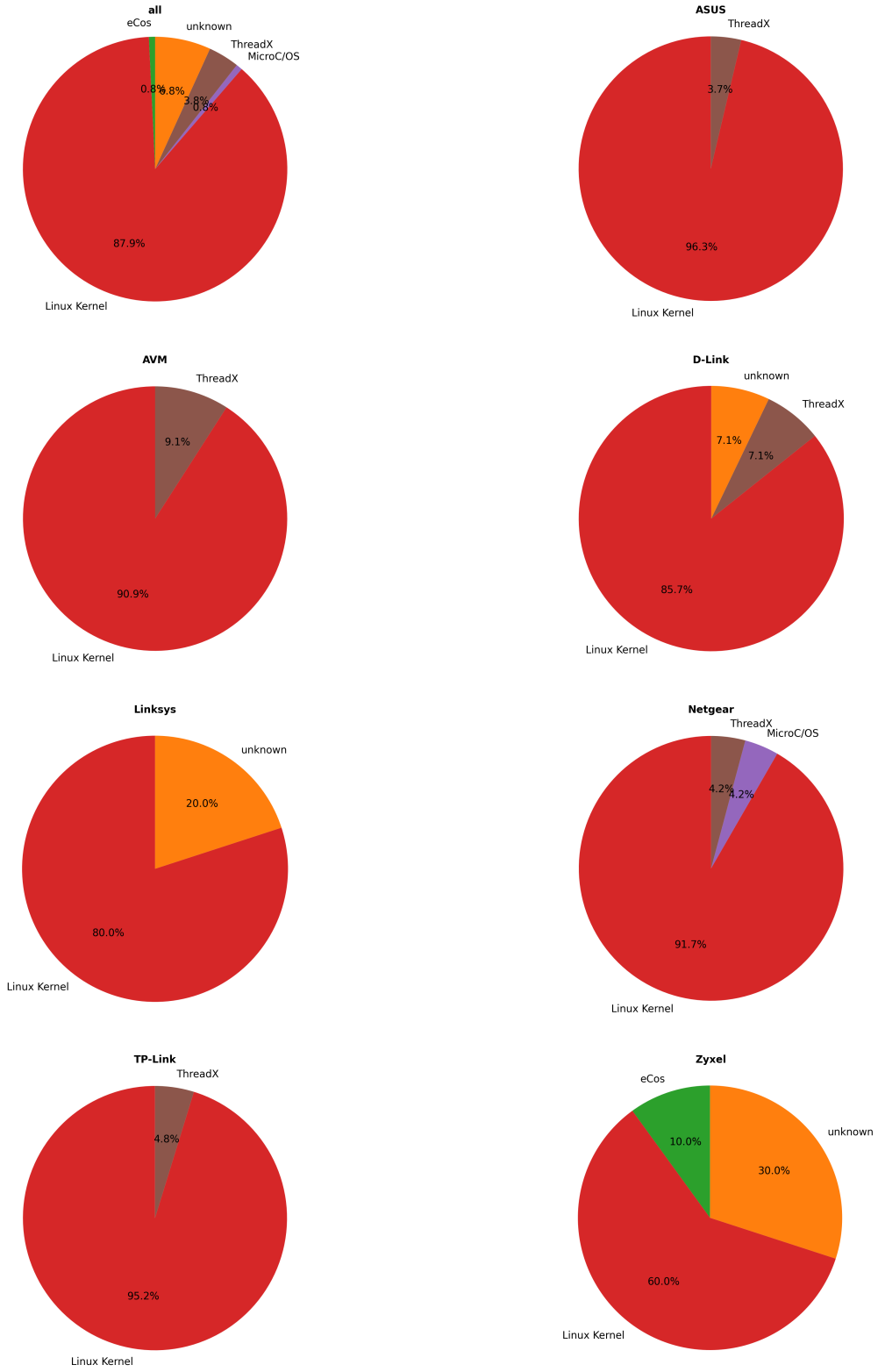

Figure 2.1: Distribution of Operating Systems

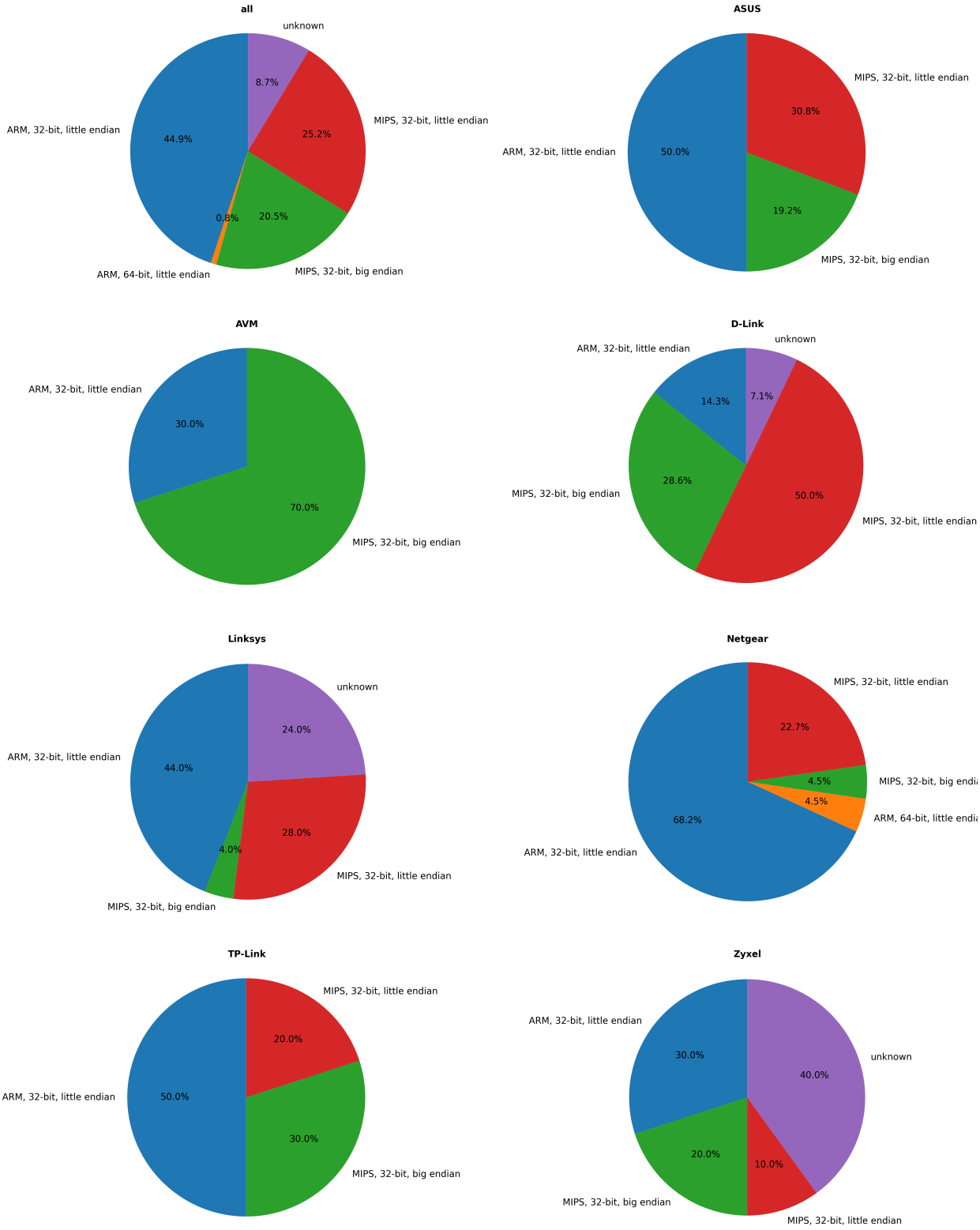

Figure 2.2: Distribution of CPU Architectures

System. The Linksys E2500, E5350, E5400, EA2750, EA6100, EA9500 and Zyxel NBG6515 could not be extracted at all as mentioned above.

3 Evaluation

We used FACT[1] to automatically analyze the firmware images. Afterwards, we utilized the REST API of FACT to extract and prepare the data of interest. We looked at five different aspects of the firmware images and their related questions regarding security.

1. **Days Since Last Firmware Update Release.** Does the vendor maintain all of their products regularly? Or in other words, how often do they fix issues?
2. **Operating System.** How old are the OS versions powering the devices? How many well known critical vulnerabilities do these versions provide?
3. **Exploit Mitigation.** Do the vendors activate exploit mitigation techniques?
4. **Private Cryptographic Key Material.** Do they publish keys that should stay private for security reasons?
5. **Hard-coded Login Credentials.** Are there any hard-coded credentials that might allow unintended access to the device?

The following sections give more detailed information about the aspects and present our findings. The single sections are structured as follows. They start with a **description** of what we look at, how the data is generated and how this relates to security. Afterwards we present and comment our **results**. Finally, we discuss the **reliability** of our results.

3.1 Days Since Last Firmware Update Release

Description

In this section we look at the number of days since the last release before 27th March 2020. If the release notes nor the website stated a release date, we used the creation date of the update file instead. The release date data may indicate if the routers are maintained on a regular basis. As mentioned before, the vendors use third party software and the developers of this software fix vulnerabilities in their software on a regular basis. From there, if a vendor did not update a firmware in a long time, it is for sure that there are several known vulnerabilities in the device. The other way round is not necessarily true. This is further discussed in the «Reliability of Results» section.

Results

81 routers got an updated within the last 365 days before 27th March 2020. However, the average number of days since the last update before 27th March 2020 is 378 days. That means in average devices did not get any security fixes within one year. 22 of 127 devices were not updated within in the last two years. The worst case was not updated since 1969 days, which means more than five years without security patches. However, the boxplot[2] in figure 3.1 shows that there are differences between the vendors. As you can see, ASUS, AVM and Netgear up-

[1] https://github.com/fkie-cad/FACT_core
[2] https://en.wikipedia.org/wiki/Box_plot

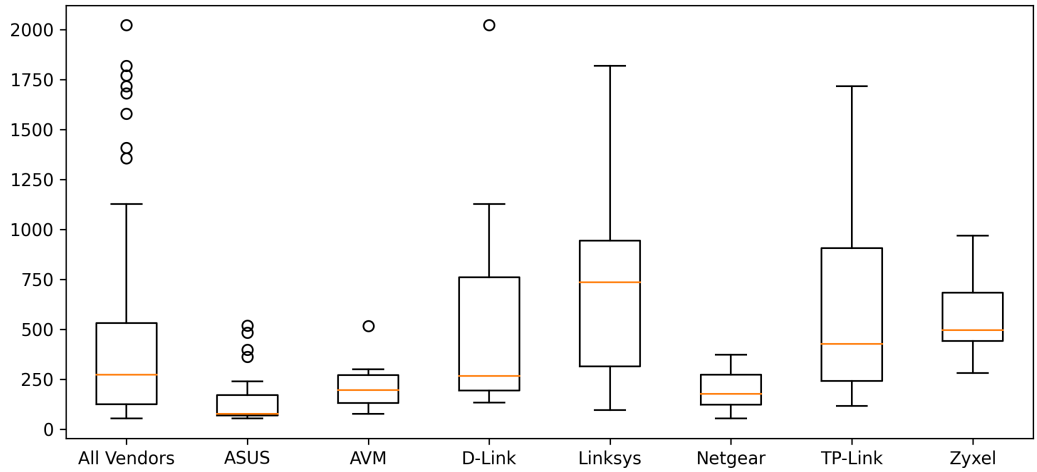

Figure 3.1: Days Since Last Release Before 27th March 2020

dated all of their devices within one and half year. The other vendor's results are more distributed. Nonetheless, these vendors updated at least half of their devices within the last two years.
In conclusion the update policy of router vendors is far behind the standards as we know it from desktop or server operating systems. However, routers are exposed to the internet 24 hours a day leading to an even higher risk of malware infection. As we have seen in section 2.1 most of the devices are powered by Linux and security patches for Linux Kernel and other open source software are released several times a year. This means the vendors could distribute security patches to their devices far more often, but they do not.

Reliability of Results

Our data is a snapshot, so that we cannot tell for sure if AVM, ASUS and Netgear update all their devices on a regular basis. Furthermore, this data does not tell if the updates provided fixes for all the vulnerabilities discovered in the meantime. Additionally, there is a possibility that vendors provide their most recent firmware version just over-the-air. That means the router tries to download firmware updates from a non public source. Therefore, the most recent firmware might not be available on the vendor's website or ftp server.

3.2 Operating System

Description

In this section we have look at the age of the OS versions. If there is an old version, there are many known vulnerabilities in most cases. We just have look at the Linux Kernel versions, because it is the most used OS as presented in section 2.1. FACT detects the Kernel Version by a regular expression and verifies that the file is not a text file[3]. This reduces the chance of false positives because configuration or documentation files are ignored. Finally, we have a look at the number of known vulnerabilities in this kernel versions. FACT maps the software versions to Common Vulnerabilities and Exposures (CVE) listed in the National Vulnerability Database(NVD)[4].

[3] `https://github.com/fkie-cad/FACT_core/blob/master/src/plugins/analysis/software_components/signatures/os.yara`
[4] `https://nvd.nist.gov`

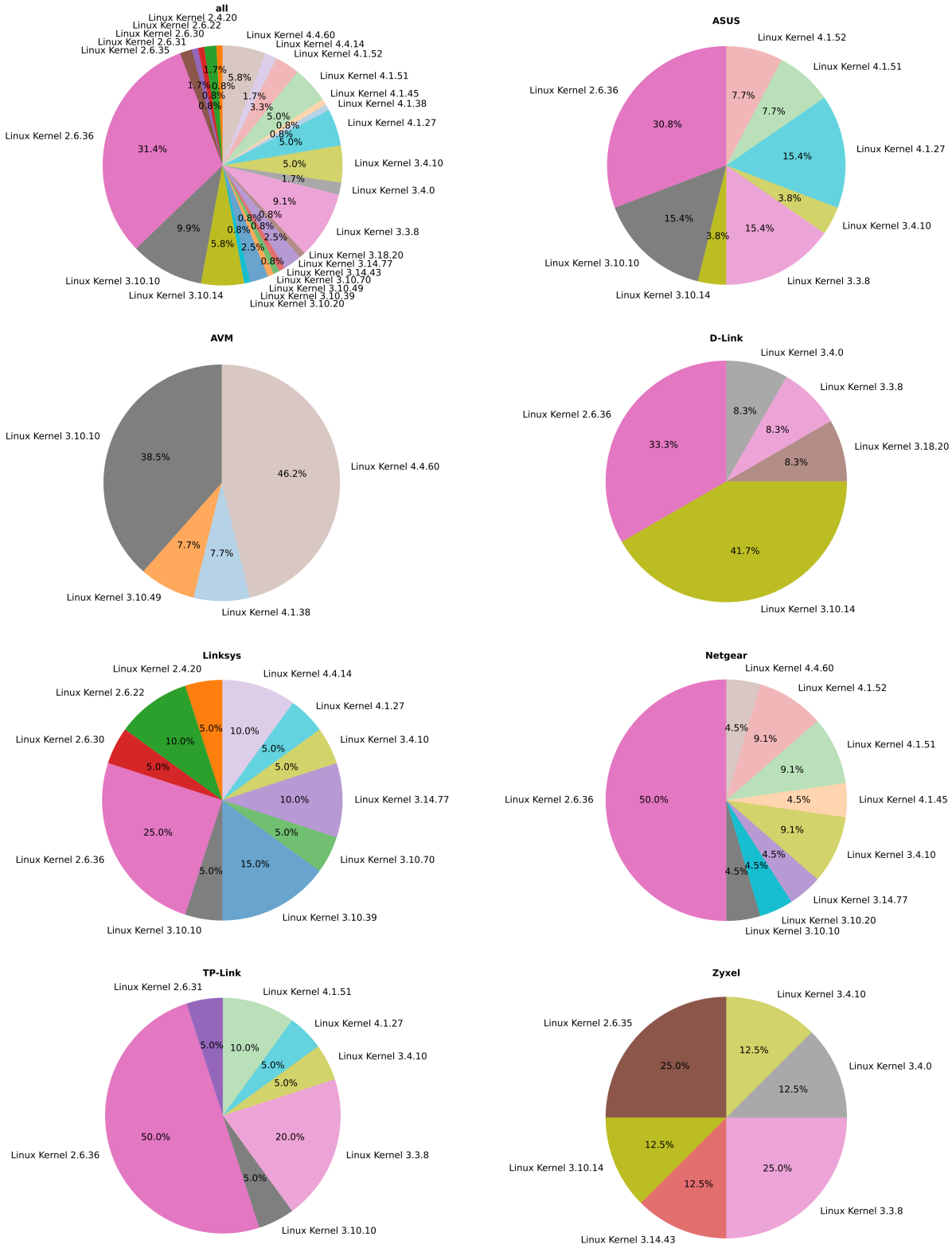

Figure 3.2: Linux Versions

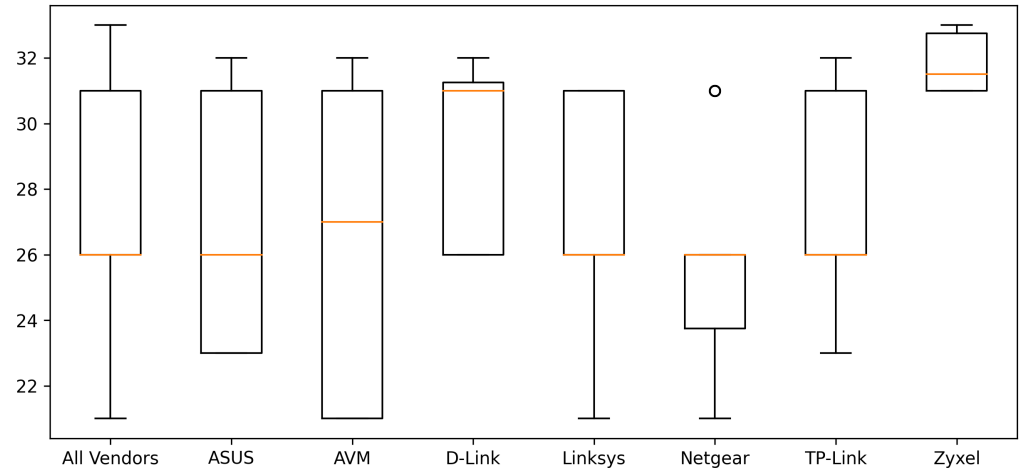

Figure 3.3: Number of Critical Severity CVEs in Linux Kernel per Firmware Image

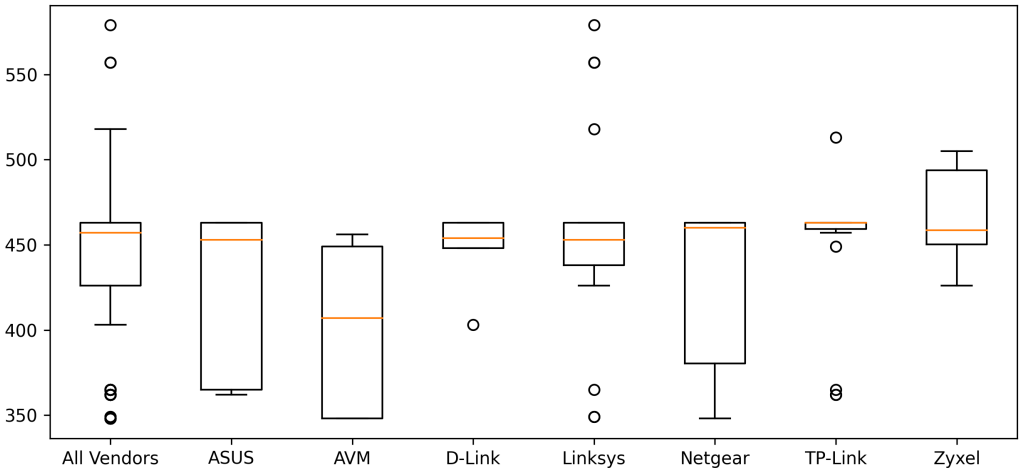

Figure 3.4: Number of High Severity CVEs in Linux Kernel per Firmware Image

At first FACT generates the Common Platform Enumeration (CPE) of the software version. CPE is basically a standard representation of software versions that is used by the NVD. Afterwards FACT queries the NVD utilizing the CPE that is mapped to the corresponding CVEs. Not all vulnerabilities are easy to exploit or their exploitation is not critical at all. Therefore, NVD defines a vulnerability severity rating called Common Vulnerability Scoring System (CVSS)[5]. In this section we are just interested in high or even critical CVEs. Unfortunately, there are two versions of CVSS used in the NVD: CVSS v2.0 and CVSS v3.0. «Critical» is defined for CVSS v3.0 only. But older CVE entries just provide a CVSS v2.0 score where as newer entries often provide CVSS v3.0 score only. That is why we counted the number of CVEs with a critical CVSS v3.0 score and additionally we counted the number of CVEs with a high CVSS v2.0 or CVSS v3.0 score. In addition, we evaluate a non official critical (noc) rating with CVSS v2.0 or CVSS v3.0 greater or equal 9.0. This rating shall close the gap between the two other ratings.

[5] `https://nvd.nist.gov/vuln-metrics/cvss`

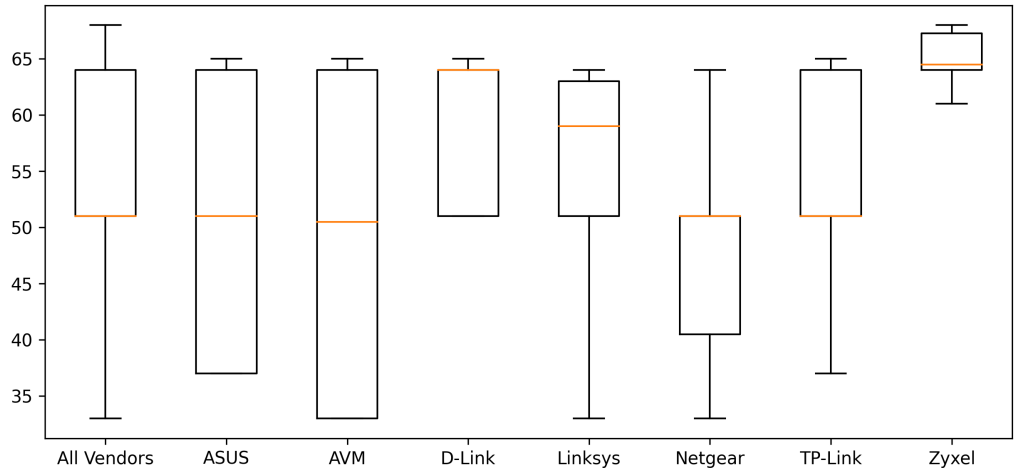

Figure 3.5: Number of NOC Rated CVEs in Linux Kernel per Firmware Image

Results

Figure 3.2 shows the Linux Kernel versions present on the devices. More than 1/3 of the devices use a kernel version 2.6.36 or even older versions. The last security update for 2.6.36 was provided on February 17th 2011 according to Wikipedia[6]. That is 9 years without security patches. Kernel version 2.6.32 was kind of a long term support (lts) kernel and got its last security update in 2016. However, none of the evaluated routers used this lts kernel. The oldest kernel version was found in the Linksys WRT54GL which is powered by a 2.4.20 Linux Kernel released in 2002. AVM is the only vendor that has not a single device running a 2.6 or older kernel. 46% of AVMs devices are powered by 4.4 Linux Kernel, which is maintained until today. Nevertheless, more than half of the AVM devices run kernel versions that are not maintained anymore. Figure 3.2 shows that the other vendors are even worse.
Figures 3.3, 3.4 and 3.5 show the known vulnerabilities perspective on this data. As you can see, even the «best» devices have at least 21 critical or 348 high rated CVEs. The routers are affected by 53 critical CVEs on average according to our noc rating. The worst case regarding high severity CVEs is the Linksys WRT54GL powered by the oldest kernel found in our study as mentioned above. There are 579 high severity CVEs affecting this product. The worst cases according the noc rating are Zyxel NBG6816 and Zyxel NBG6815 affected by 68 critical CVEs.
We refer to our noc rating at this point, because our data shows that critical CVEs according to CVSS v3.0 rating are not a good rating to compare old software with newer software. Figure 3.4 states that the mean (orange line) of critical CVEs per Firmware in AVM devices is higher than the mean of all devices, even if AVM uses more recent kernel versions than others. On the other hand Netgear seems to perform very good, even if half of their devices use a kernel that was not patched in nine years. This does not sound plausible, but it can be explained easily. As mentioned above, the CVSS v2.0 does not define «critical» CVEs and older CVEs do not have a CVSS v3.0 rating. This means even if a vulnerability of an old kernel would be ranked as «critical» in the means of the CVSS v3.0 rating, it is not part of this statistic. If we have look at our noc or the high rating, the data is much more plausible.
In conclusion, there is not a single device without known critical vulnerabilities. Another interesting fact is that even if a router was updated in the last year (compare with section 3.1), software versions known for vulnerabilities were not necessarily replaced by newer ones. Otherwise there would not be so many devices utilizing a 2.6 Linux Kernel. We also learned that just looking at

[6] `https://de.wikipedia.org/wiki/Linux_(Kernel)`

officially critical ranked CVEs does not tell the whole picture.

Reliability of Results

There is a chance that the vendors develop their own kernel patches to fix critical kernel issues. However, we think that this is rarely done in practice if you consider the overhead of developing and testing own kernel patches for a high number of high severity issues.
Additionally, there is a chance of false positive Kernel detection, because FACT uses a quite simple pattern matching approach. FACT detects two Linux Kernels in 4 of the 116 Firmware Images powered by Linux. Nevertheless, the AVM Fritz!Box 5490 and 5491 might consist of two embedded devices, if the integrated glass fiber modem is implemented as a separate embedded device for instance. Their second kernels were found in a firmware file: `/lib/firmware/ath_tgt_fw2.fw`. This might be true for the the Zyxel NBG6815 and NBG6816, too. Their second kernels were found in `/lib/firmware/topaz-linux.lzma.img`. It sounds plausible to us that these are running two Linux systems in one router.
Another Problem is the CPE. There are misspelled entries or even different CPEs regarding the same software version because the NVD CVE entries are generated manually. This means there is a high probability of false negatives. Sanguino and Uetz discuss this issue in detail in their paper[7].

3.3 Exploit Mitigation

Description

There are many exploit mitigation techniques[8] suitable to protect embedded devices. Especially Linux based devices can be protected easily, because the tools are free and tested in practice for years. The idea of exploit mitigation techniques is simple; they implement additional features that shall prevent the exploitation of a vulnerability. These techniques work for known and unknown vulnerabilities. However, they cannot prevent every exploitation attempt. The exploit mitigation techniques that we look at can be enabled or disabled on file basis. Our data shows the percentage of how many executables enable a specific technique per firmware. FACT uses a method similar to the checksec[9] tool to verify if a feature is enabled. Since checksec supports elf-binaries only, this section covers the 116 Linux firmware images of our 127 image evaluation set.
We analyzed the usage of the following exploit mitigation techniques:

- **Non-Executable Bit(NX)** marks regions of the memory as non executable. If an attacker wants to execute own code, he has to store it somewhere in the victims memory. The idea of NX is to mark regions of memory as non executable that should not provide executable code.

- **Position-Independent Executable (PIE)** loads a program at a random location in memory. If mitigation techniques like NX are enabled, an attacker needs to reuse code of legitimate programs and libraries already loaded to memory. This is called return-oriented programming (ROP)[10]. Therefore, the attacker needs to know where to find the code in the victim's memory. Since PIE randomizes these code locations, ROP is much harder to implement.

[7] https://arxiv.org/pdf/1705.05347.pdf
[8] https://7h3ram.github.io/2012/exploit-mitigation-techniques-on-linux.html
[9] https://github.com/slimm609/checksec.sh
[10] https://en.wikipedia.org/wiki/Return-oriented_programming

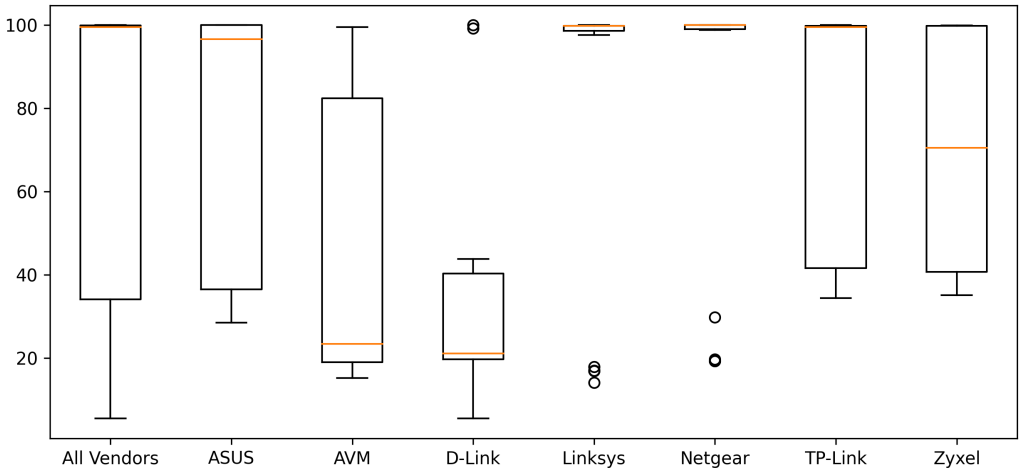

Figure 3.6: Percentage of Executables with NX Enabled per Firmware Image

- **RELocation Read-Only(RELRO)** protects the Global Offset Table (GOT)[11] against manipulations during program runtime. Basically GOT maps memory locations of functions from shared libraries or global variables, so that the program can use them. If an attacker does ROP, he often tries to manipulate this mapping to redirect a legitimate function call to another function. RELRO marks GOT as read only after the program was loaded into memory. Therefore, it cannot be manipulated by an attacker during runtime. There is a partial RELRO[12] mode protecting global variable overwriting only and a full RELRO mode protecting the whole GOT. In our analysis we count partial mode and full mode as enabled.

- **Stack Canaries** mitigate buffer overflow attacks. In C you have to reserve space in memory to store incoming data. If the program does not check if the incoming data is larger than the reserved space (called buffer), the incoming data may overwrite other data in the memory. This can involve data affecting the program execution enabling an attacker to execute own code. The basic idea of Canaries is to store a special byte sequence called Canary on specific positions in memory. These sequences are checked for changes during runtime of the program.

- **FORTIFY_SOURCE** mitigates buffer overflow attacks as well. There are safe string operation functions in C limiting the maximum input data length. This prevents buffer overflow attacks. The idea of FORTIFY_SOURCE is to replace unsafe string operation functions by safe string operation functions if possible. Possible means the compiler is able to estimate the maximum required buffer size. Obviously, this is not possible all the time.

Results

The usage of exploit mitigation techniques is quite rare with few exceptions. NX is used on nearly all devices of all vendors as you can see in figure 3.6. However, there are big differences in adoption percentage; even if devices are developed by the same vendor.
Figure 3.7 shows that PIE is used by AVM on over 90% of executables in most devices. The AVM Fritz!Box 4020 seem to not use PIE at all. But this device was not extracted by FACT correctly. Thus, it might use PIE as well, but we cannot tell for sure. All other vendors use PIE in all their

[11] `https://en.wikipedia.org/wiki/Global_Offset_Table`
[12] `https://ctf101.org/binary-exploitation/relocation-read-only/`

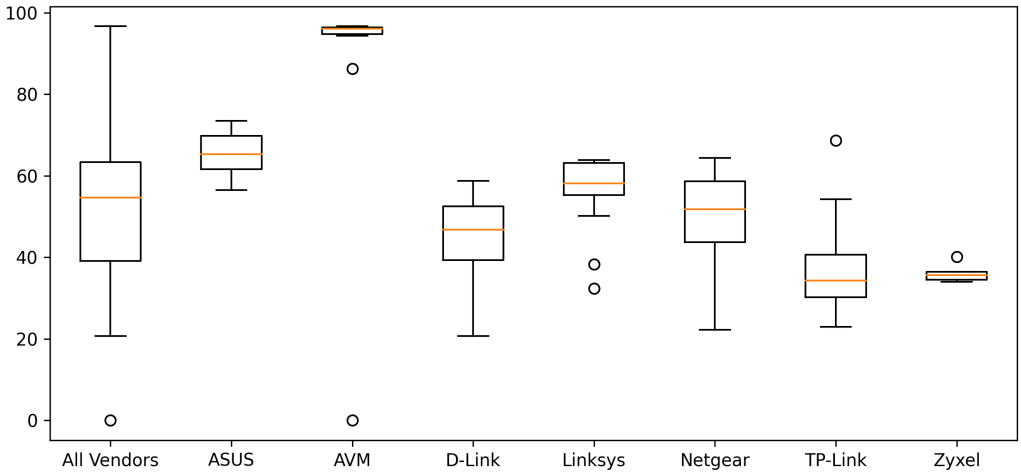

Figure 3.7: Percentage of Executables with PIE Enabled per Firmware Image

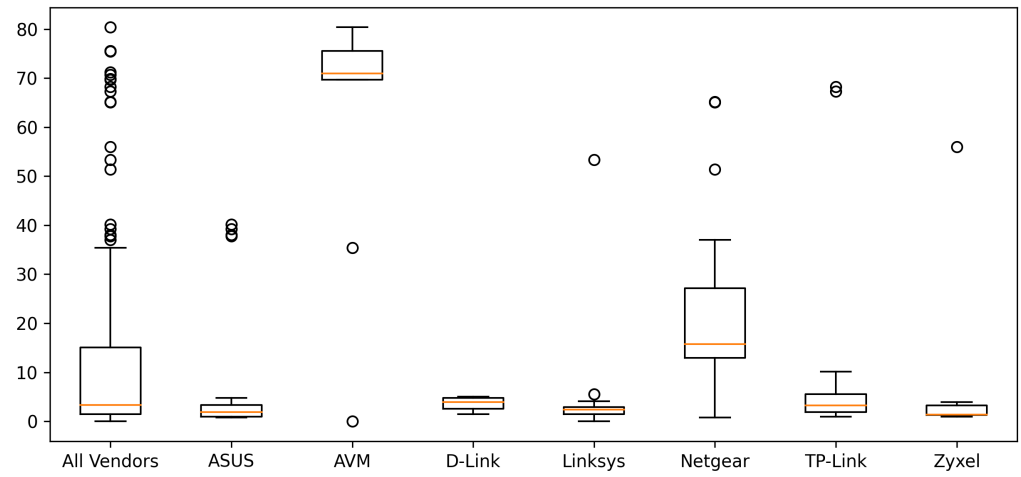

Figure 3.8: Percentage of Executables with RELRO Enabled per Firmware Image

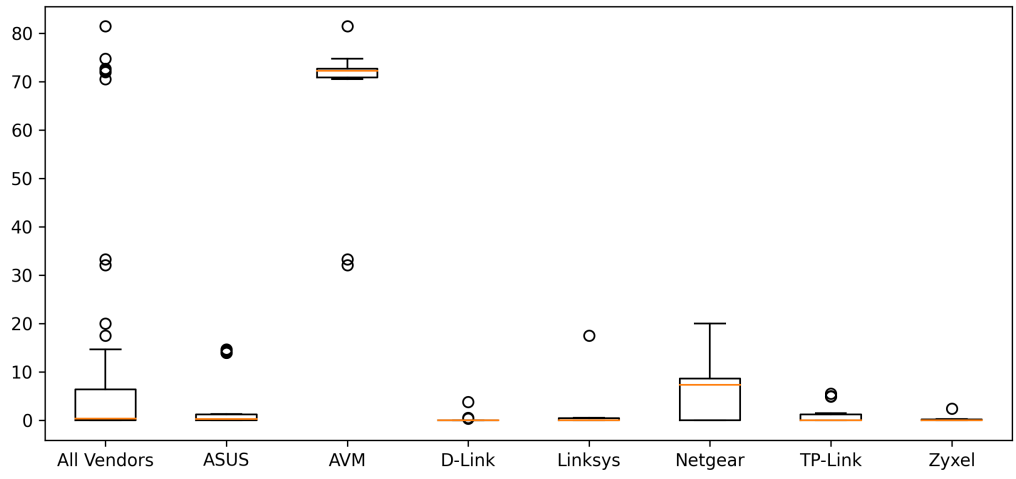

Figure 3.9: Percentage of Executables with Canary Enabled per Firmware Image

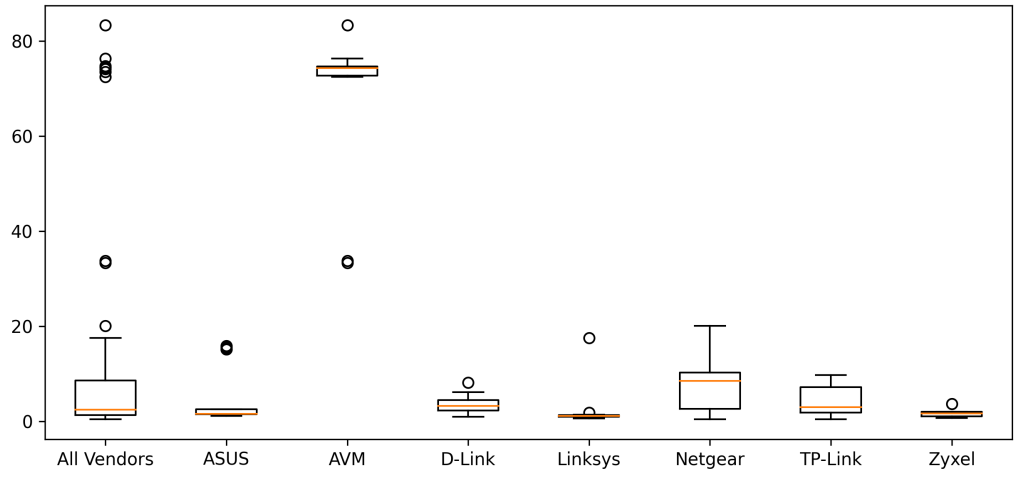

Figure 3.10: Percentage of Executables with FORTIFY_SOURCE Enabled per Firmware Image

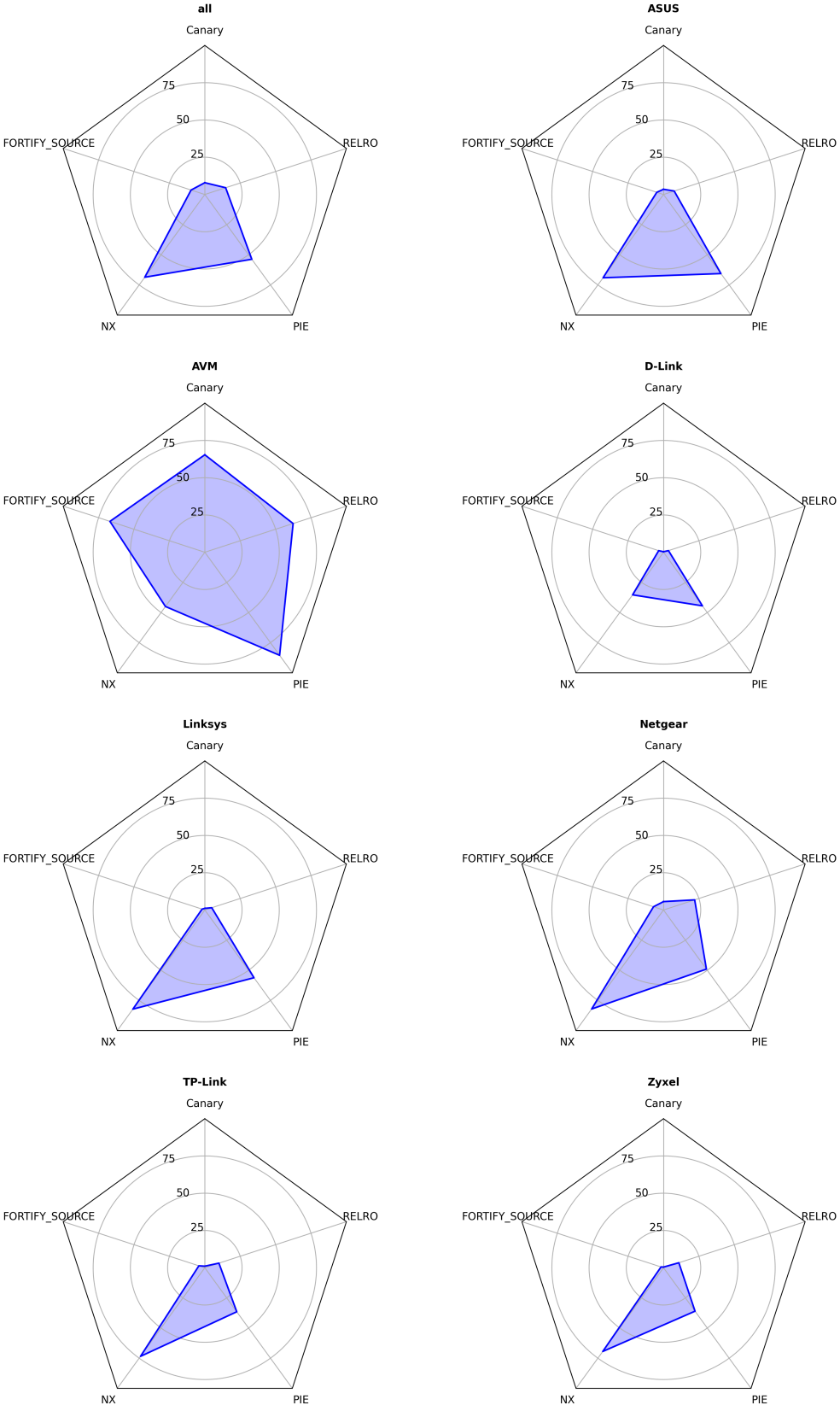

Figure 3.11: Mean of Enabled Exploit Mitigations

products as well. But the percentage of files that enabled this feature is significantly lower.
AVM enabled RELRO on more than 70% of the files in almost all devices. The AVM Fritz!Box 4020 does not seem to use it, which sounds strange because AVM uses RELRO on all other devices. The reason why we did not find anything might be that this device was not extracted correctly as mentioned above. The other vendors enable RELRO to some extend on some devices. Only D-Link does not use it much on any device. The highest value can be found on D-Link DWR-932 F featuring 5% of executables with RELRO enabled.
Figure 3.9 and 3.10 show that Canaries and FORTIFY_SOURCE are widely used in most AVM devices. The other vendors do not use them much.
Figure 3.11 sums up the usage of all exploit mitigation techniques utilizing radar charts[13]. The bigger the pentagon the more security features are enabled on a higher percentage of files. The «all» radar chart shows that NX and PIE are more commonly used on the routers than Canaries, FORTIFY_SOURCE and RELRO. AVM has the largest pentagon since they use Canaries, FORTIFY_SOURCE and RELRO quite much. D-Link has the smallest pentagon.

Reliability of Results

The AVM Fritz!Box 4020 example has shown that if the extraction is not complete the results might be wrong. Furthermore, the technique used by the checksec and the FACT plugin tries to find indicators in the binaries that are typical for a specific mitigation techniques. Since they are just indicators, there is a risk of false positives.

3.4 Private Key Material

Description

Cryptographic private key material is typically used to authenticate a communication partner. There are three typical scenarios:

1. **Secure Communication Channel**
2. **Signing Data**
3. **Login to Remote Services**

If you consider that private keys can be extracted easily from a firmware image, the keys do not provide any security at all if they are part of the firmware. This means any attacker can impersonate the device and do man in the middle attacks for instance. We want to make it absolutely clear that there is no good reason to **publish** a **private** key, because **a published private key does not provide any security at all!** If the functionality requires private keys on the device, the vendors should follow the OWASP Embedded Application Security Best Practices[14]. OWASP suggests to use a hardware security element. When doing so, the key is not part of the firmware and it is even hard to extract the key from the device itself.
In this part of the study we count how many private keys are published in the firmware of a device. Hence, the keys are part of the firmware, these keys are shared with all devices of the same model. This means one private key published in a firmware puts thousands of devices in danger.

[13] https://en.wikipedia.org/wiki/Radar_chart
[14] https://owasp.org/www-project-embedded-application-security/#div-project

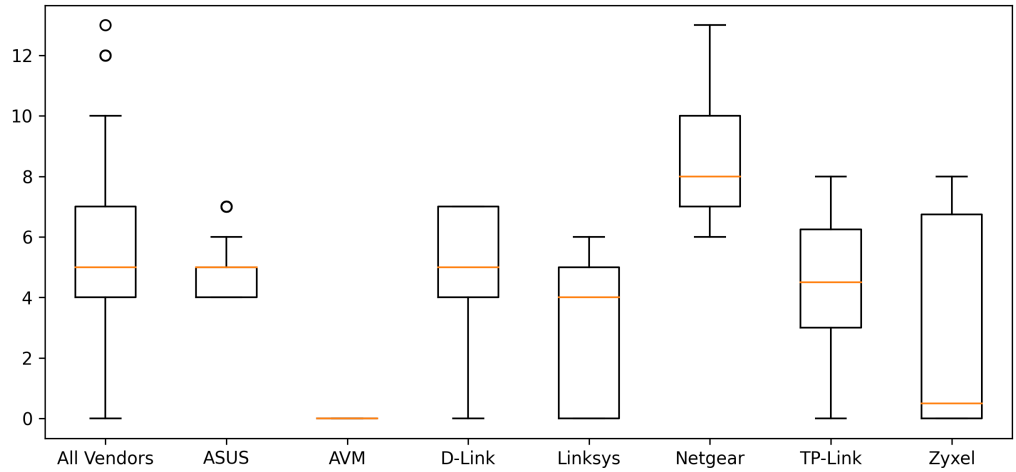

Figure 3.12: Number of Private Keys per Firmware Image

Results

Figure 3.12 shows how many private keys are published in the firmware images. There are almost five private keys published per firmware image in average. The Netgear R6800 provides a total number of 13 private keys in a single device. There are just few firmware images without private keys. As you can see AVM is the only vendor not publishing any private key in its firmware images. Linksys and Zyxel seem to share less private keys than the other vendors, but their low number on some devices is caused by extraction failures mentioned in chapter 2. Therefore, the data can be misleading at this point.
In conclusion the use of a private key authentication mechanism suggests a sense of security. But a lot of private keys are published in almost any router, so that there is no security in using this mechanism on these devices. Do not forget that even a single published private key can be a severe security issue. Just one vendor does a good job at this point.

Reliability of Results

The regular expressions used by FACT to detect private keys are quite unique, so that false positives are unlikely. On the other hand, there might be false negatives, because not all firmware images could be extracted completely as mentioned above. Nevertheless, there might be private keys used during the development, but not used in the final product. However, if these are not used at all, it would be safer to remove them anyway.

3.5 Hard-coded Login Credentials

Description

The Mirai botnet used hard-coded telnet credentials to infect millions of embedded devices and this is one reason the OWASP suggests to not use hard-coded (static) user accounts[15]. In other words, any hard-coded credential in a firmware image puts all devices to danger if these credentials can be used to log on to the device. It gets even worse, if the passwords are public or easy crackable. In a worst case scenario passwords are stored in plain text on the device.

[15] `https://owasp.org/www-project-embedded-application-security/#div-project`

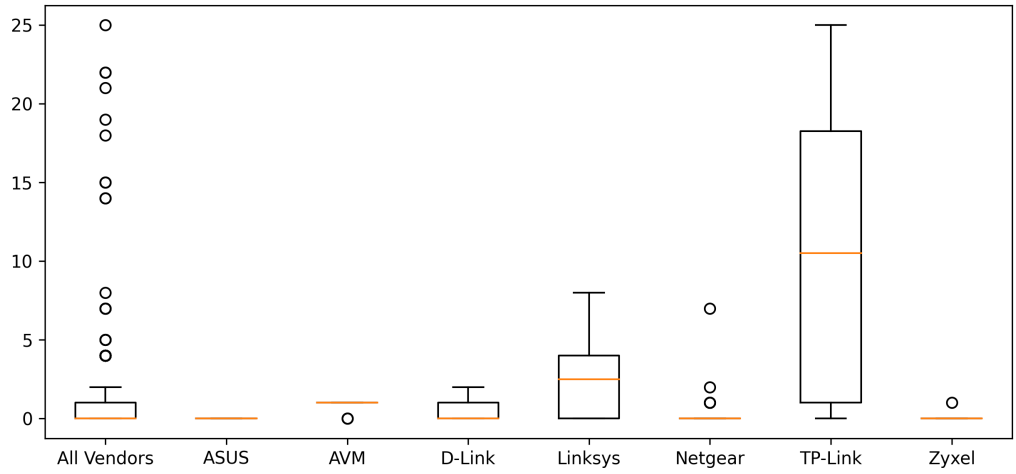

Figure 3.13: Number of Hard-coded Credentials per Firmware Image

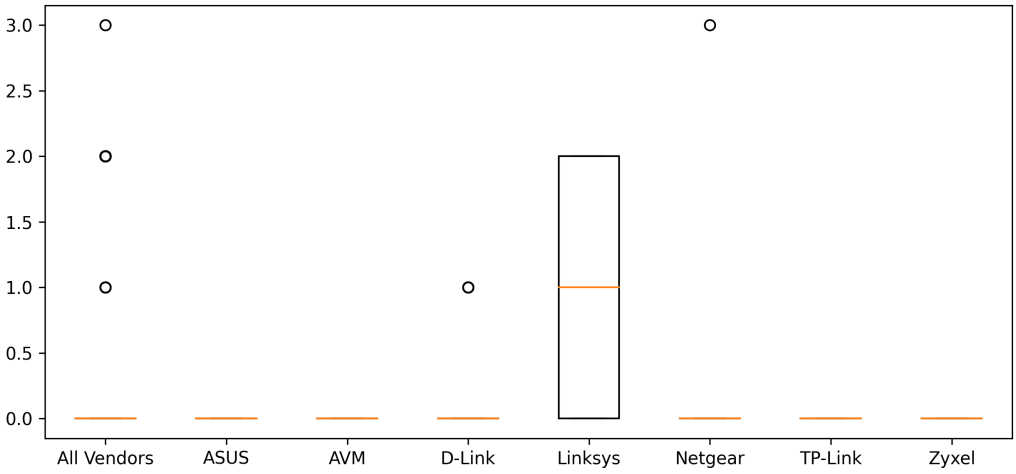

Figure 3.14: Number of Well Known Hard-coded Passwords per Firmware Image

Furthermore, if the user cannot change a password, you might get a feeling that the password is related to a backdoor. In this part of the study we count the hard-coded login credentials in the firmware images. FACT detects all credentials that look like an entry of a shadow or htaccess file. Normally, these passwords are stored as a hash value. If the password is long enough it should be safe, if it is not leaked in any other way. Hence, we check, if the hard-coded password is on the list of the 10k most common passwords[16], as well. Additionally we check, if the password is listed at routerpasswords.com a database for well known router passwords. Since the extracted passwords are hashed, FACT uses a dictionary attack utilizing John the Ripper[17] to check the passwords.

[16] `https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10k-most-common.txt`

[17] `https://www.openwall.com/john/`

Results

Figure 3.13 shows the number of hard-coded passwords present in each firmware image. The good news is that more than 60% of the router firmware images do not have hard-coded login credentials. The bad news is that 50 routers do provide hard-coded credentials. 16 routers have well known or easy crackable credentials. The worst device is the Netgear RAX40 with the following three well-known credentials:

- root:amazon
- nobody:password
- admin:password

However, we do not know, if you can log on to theses accounts remotely.
ASUS is the only vendor not storing any hard-coded credential in its firmware images.

Reliability of Results

As mentioned above, we do not know if any of the found credential is related to a running service connectable remotely. However, it would be more secure, to remove theses accounts, if no one is intended to connect to them. If someone is intended to connect to it, you should ask yourself, who that might be and why? Furthermore, there is high chance of false negatives, since the past has shown that vendors use programs to set default passwords at runtime[18]. These are not found by FACT. Additionally, we did not find any hard-coded passwords in firmware images, that we could not extract. Therefore, there might be even more false negatives.

[18] `https://pierrekim.github.io/blog/2016-09-28-dlink-dwr-932b-lte-routers-vulnerabilities.html`

4 Conclusion

Our analysis showed that Linux is the most used OS running on more than 90% of the devices. However, many routers are powered by very old versions of Linux. Most devices are still powered with a 2.6 Linux kernel, which is no longer maintained for many years. This leads to a high number of critical and high severity CVEs affecting these devices.
Since Linux is the most used OS, exploit mitigation techniques could be enabled very easily. Anyhow, they are used quite rarely by most vendors except the NX feature.
A published private key provides no security at all. Nonetheless, all but one vendor spread several private keys in almost all firmware images.
Mirai used hard-coded login credentials to infect thousands of embedded devices in the last years. However, hard-coded credentials can be found in many of the devices and some of them are well known or at least easy crackable.
However, we can tell for sure that the vendors prioritize security differently. AVM does better job than the other vendors regarding most aspects. ASUS and Netgear do a better job in some aspects than D-Link, Linksys, TP-Link and Zyxel.
Additionally, our evaluation showed that large scale automated security analysis of embedded devices is possible today utilizing just open source software. To sum it up, our analysis shows that there is no router without flaws and there is no vendor who does a perfect job regarding all security aspects. Much more effort is needed to make home routers as secure as current desktop or server systems.

List of Abbreviations

CPU	Central Processing Unit
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
FACT	Firmware Analysis and Comparison Tool
GOT	Global Offset Table
lts	long term support
noc	non official critical
NVD	National Vulnerability Database
NX	Non-Executable Bit
OEM	Original Equipment Manufacturer
OS	Operating System
PIE	Position-Independent Executable
ROP	return-oriented programming