UNIVERSIDADE FEDERAL DE PERNAMBUCO

Centro de Informática
UFPE

Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Ciência da Computação

A Security Analysis of Residential Gateways leased by a Brazilian Internet Service Provider to Customers

Pedro Henrique Tôrres Santos
phts@cin.ufpe.br

ADVISED BY
Prof. Dr. Paulo André da Silva Gonçalves
pasg@cin.ufpe.br

Recife
April 30, 2021

Agradecimentos

Agradeço a meus pais, Pierre e Germana, que sempre me apoiaram e me incentivaram a batalhar e conquistar meus sonhos. Que me criaram com tanto amor e carinho. Que me deram todas as oportunidades para que eu chegasse aqui e me tornasse o profissional que hoje sou.

Agradeço a meus avós, Fernando e Verônica, que foram uma parte muito importante da minha criação e uma fonte de inspiração para a vida. Com quem tanto viajei e que basicamente me adotavam nos meus períodos de férias.

Agradeço a meu namorado, Mário, que tanto amo e que me fez crescer muito como pessoa. Por ter me incentivado a adotar Mingau e Óliver, meus dois gatos que não consigo viver sem. Por estar do meu lado nos bons e maus momentos. Por todas as risadas que demos juntos e memórias felizes eternizadas em minha mente. Por me apoiar em momentos difíceis e me tranquilizar. Por todas as horas de sono perdidas me ajudando tanto nesse trabalho, que não teria conseguido concluir de outra forma.

Agradeço a todos os meus amigos, professores, e coordenadores do Diocesano, que me fizeram aproveitar cada momento do ensino fundamental e médio, um período de muitas boas recordações da minha vida. Por tudo que ai passei, errando e acertando não me arrependo do caminho que trilhei, pois foi ele que me fez chegar aqui.

Agradeço a André, Claudio, Danilo, Edjan, Luca, Val, e Sergio, amigos que estiveram comigo ao longo da graduação e que proporcionaram tantos momentos inesquecíveis, sem vocês não sei o que teria sido esses 5 anos na faculdade. Nunca esquecerei das nossas calouradas, festas, viradas, e saídas juntos, muito menos do Lobo Hamilton.

Agradeço a Dudu e Luiz, que apostaram em mim e me convidaram para a LookMobile. Onde pude dar meus primeiros passos na carreira profissional e aprender bastante.

Agradeço ao PET-Informática e a todos os meus amigos com quem lá trabalhei. Por tudo que conseguimos realizar e pelo impacto que causamos dentro e fora do CIn. Seja com a OPEI, com o HackaPET, com as visitas ao CIn, com a recepção dos calouros, e até com as doações de sangue que organizamos.

Agradeço a todos os meus amigos de Core Engineering na In Loco e especial a Airton, Anabuki, Colossus, Renato, Mendes, Lauro, e Paulo que fizeram e fazem parte comigo do time de SRE. Que pude aprender tanto com cada um e que me fizeram crescer muito profissionalmente. Por todas as risadas no aquário e por proporcionar um ambiente de trabalho em que todos os dias acordava contando as horas para chegar ao office.

List of Figures

List of Tables

Contents

1 Introduction

Upon acquisition of a residential Internet service, it is usual for Internet Service Providers (ISPs) to provide the necessary equipment for the customer to use their services [25, 7, 27, 31]. Depending on whether or not the customer has also subscribed to Voice over IP (VoIP) and/or Internet Protocol Television (IPTV) services, the device may be more complex to deliver telephony and Television (TV) functionality.

The device leased by the ISP generally acts as a residential gateway and it is also known as Customer-Premises Equipment (CPE). The CPE is installed on the subscriber's residence and connects to the ISP via the physical medium negotiated, in Brazil, mostly copper and fiber [1]. With the device targeted to anyone who wants Internet and other services at home, it is desired that the CPE works as a plug'n'play solution with no technical expertise required from its users.

When providing this out-of-the-box solution, the ISP takes responsibility for securing the residential gateway with the default configurations, as most users will never change them. But serious vulnerabilities have been found in the past years on residential gateways from several vendors, and some of them were introduced by ISPs, raising concerns about the security measures put in place in CPEs leased to customers [37].

The authentication and encryption mechanisms for wireless networks also play a major role in securing the data transmitted wirelessly, as anyone may listen to the bits being sent and received. But over the time, some standards have been found to be flawed and are no longer considered safe to be used, even when accurately implemented and properly configured [45, 46].

These problems are aggravated by the discrepancy between how long the manufacturer and/or the ISP are willing to provide security patches for the device and how long customers use the equipment. Maintaining a different range of devices or replacing existing CPEs with newer ones is costly and doesn't benefit the company directly, making it not attractive from a business perspective [17].

1.1 Objectives

This work intends to study CPEs provided to customers of one Brazillian ISP. The ISP was chosen based on its national relevance and on the ability of the researcher to run the proposed actions with the permission of multiple customers on their environment. Due to concerns regarding a direct reference to the provider, the ISP name will not be disclosed.

The main objective of the research is to check if the devices are being properly configured and provide a desired level of security. To better understand the impact of using a standard for authentication and encryption over another on a CPE, the protocols and their security properties are analyzed and known issues and vulnerabilities are studied. If some issue is unveiled, the risks and possible mitigations are discussed.

Tests are performed on the management interfaces of the CPEs to ensure that they are robust enough against attacks that may compromise the network, the devices connected to it, or the device itself.

The ISP servers that may impact on the customer privacy and security are analyzed and the connection from the residential gateway to them is inspected. This work specially aims to study the servers that handle the telephony service with the Session Initiation Protocol (SIP) and the ones that perform remote management of the CPEs with the CPE WAN Management Protocol (CWMP).

It is also investigated if a customer is able to replace the ISP CPE with a piece of commercial equipment, taking the responsibility of maintaining the device from the ISP and leaving to itself. The process of doing so is detailed and the drawbacks are discussed.

Finally, recommendations based on the findings of this research and on the background knowledge are given. Action points that the customer may perform to improve its security are explored for the different pieces of equipment.

1.2 Document Structure

The document is segmented in 9 chapters, this one being the introduction for the research. Chapters 2 and 3 respectively discuss the background knowledge in Wireless Networks and ISP-side Protocols. The CPEs analyzed by the research are specified in Chapter 4, the method used to prepare the devices is explained, and data is collected from them. In Chapter 5, the configuration data collected in each CPE is inspected and the values set are discussed. The security of the Hypertext Transfer Protocol (HTTP) Management Interface of the devices is assessed on Chapter 6. Chapter 7 analyzes the implementation of some ISP-side Services. The possibility of switching the ISP-provided CPE with another equipment is researched on Chapter 8, instructing how to do so. Finally, Chapter 9 exposes the conclusion of this work.

2 Background in Wireless Networks

In this chapter, the origins and the motivation underneath Wireless Fidelity (Wi-Fi) is exposed. The different authentication and encryption protocols are detailed, as defined by the Institute of Electrical and Electronics Engineers (IEEE), and have their security problems explained.

The research covers Open System and Wi-Fi Enhanced Open (WEO) standards, for unauthenticated or open networks; Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, and WPA3 standards, for authenticated or closed networks; and Wi-Fi Protected Setup (WPS) and Wi-Fi Easy Connect (WEC) standards, for quick setup networks.

The knowledge acquired in this chapter is later used for the security assessment of the CPEs and to better understand the impact of the configurations set.

2.1 IEEE 802.11 and Wi-Fi

IEEE 802.11 is a working group, part of the IEEE LAN/MAN Standards Committee (LMSC), that develops and maintains networking standards and recommended practices for Wireless LAN (WLAN) [13]. The first version of its standard was released in 1997 and operated at a 2.4 GHz frequency, allowing data to be transmitted up to 2 Mb/s.

The original standard was quickly superseded by the IEEE 802.11b amendment. Published in 1999, it increased the maximum transfer rate of the standard to 11 Mb/s while staying in the 2.4 GHz band.

In the same year, companies joined forces together to certify interoperability of products implementing the IEEE 802.11 standard and to promote it as the global WLAN standard across all market segments. The Wireless Ethernet Compatibility Alliance (WECA) was born with that mission [48].

For months, WECA worked closely with several vendors of IEEE 802.11b equipment to develop an interoperability testbed for the standard. The work was complete in February 2000 and WECA members were able to submit their products against the test matrix. Upon success, the product would receive the Wi-Fi CERTIFIED^{TM} seal and the company would be allowed to use the Wi-Fi^{TM} logo on the device's advertising and packaging. The seal indicates that the product "met industry-agreed standards for interoperability, security, and a range of application specific protocols" and works with any other devices that also bear the symbol [38]. In 2002, WECA changed its name to Wi-Fi Alliance.

In 2018, the Wi-Fi Alliance introduced a new designation to identify the Wi-Fi generations [40], moving away from the traditional IEEE 802.11 amendment names and adopting a simpler numerical sequence scheme as shown in Table 1.

IEEE 802.11 Amendment	Wi-Fi Generation
802.11b	Wi-Fi 1
802.11a	Wi-Fi 2
802.11g	Wi-Fi 3
802.11n	Wi-Fi 4
802.11ac	Wi-Fi 5
802.11ax	Wi-Fi 6

Table 1: IEEE 802.11 Amendment to Wi-Fi Generation mapping

2.2 Open Network Standards

2.2.1 Open System

The Open System authentication is essentially a null authentication algorithm [14]. Any wireless client may become authenticated if the target of the request is configured to use this kind of authentication, although the recipient is not required to accept the authentication request and may decline it.

2.2.2 Wi-Fi Enhanced Open

The new security standard, was introduced in 2018 and intends to replace the Open System authentication for public networks. Based on the Opportunistic Wireless Encryption (OWE) [10], its main difference is the introduction of a privacy layer that encrypts traffic between the client and the station [41].

2.3 Protected Network Standards

2.3.1 WEP

The WEP algorithm is part of the original IEEE 802.11 standard and was intended to avoid eavesdropping and ensure privacy for those connected to a wireless network [14].

It relies on a secret that is known by the station and has been delivered to the wireless client, presumably via a secure channel, to accept or decline the authentication request. An encrypted challenge is sent to the client, which decrypts it using the shared key and sends the decrypted challenge back to the station, proving that it knows the password, thus should be authenticated successfully.

Algorithm The algorithm [14] uses a key sequence produced by a Pseudo-Random Number Generator (PRNG) to encrypt and decrypt data, establishing a secure channel as shown in Figure 1. The PRNG used by WEP requires a 64-bit or 128-bit seed to initialize its initial state and derive the associated keystream. The WEP Shared Key (SK), either 40 bits (WEP-40) or 104 bits (WEP-104) in size, and the Initialization Vector (IV), 24 bits long, are concatenated and used as the seed for the PRNG. The IV content is random and its

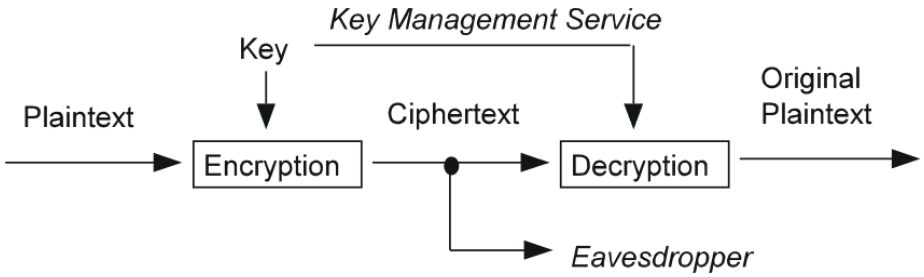


Figure 1: A Confidential Data Channel
Source: [14]

role is to avoid the PRNG to be initialized with the same seed more than once, preventing the data from being encrypted with an already known key sequence.

The Integrity Algorithm (IA) is used to detect if the message sent via the wireless network was corrupted and possibly tempered with on transport, it takes the plaintext as input and produces 32 bits as the Integrity Check Value (ICV).

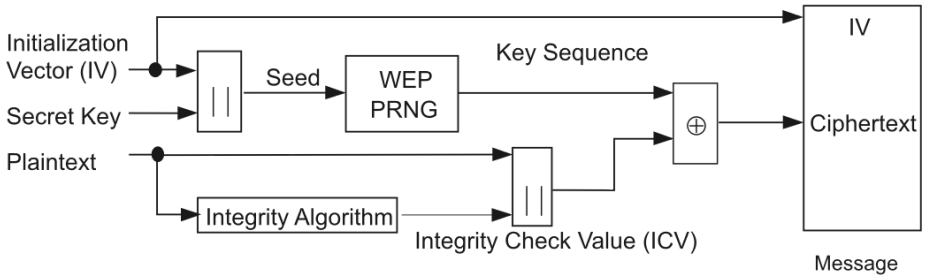
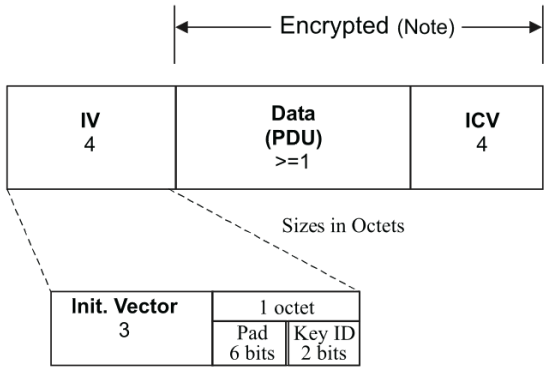


Figure 2: WEP Encipherment Block Diagram
Source: [14]

As illustrated on Figure 2, to encrypt a message, the algorithm requires the SK, the IV, and the plaintext. First, the SK and the IV are fed to the PRNG, and N bits from the keystream are stored in KS, where N is the length of the plaintext added by the size of the ICV. Then the plaintext is fed to the IA and the output is stored in ICV. The plaintext is concatenated with the ICV and XORed with KS, resulting in the ciphertext. Finally, the IV is concatenated with the ciphertext and returned as the output of the algorithm, the MAC PDU (MPDU).

Figure 3 shows the encrypted MPDU as constructed by the WEP algorithm. It is transmitted over the wireless connection and then decrypted by the receiver. To decrypt the MPDU, only the SK is required. So the confidentiality of the data channel established using WEP essentially relies on the secrecy of the shared key.



NOTE-The encipherment process has expanded the original MPDU by 8 octets, 4 for the Initialization Vector (IV) field and 4 for the Integrity Check Value (ICV). The ICV is calculated on the Data field only.

Figure 3: Construction of Expanded WEP MPDU
Source: [14]

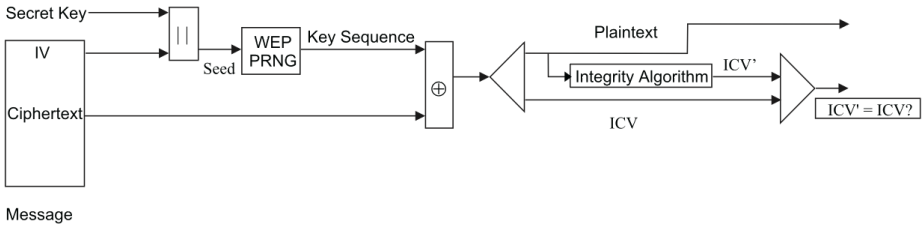


Figure 4: WEP Decipherment Block Diagram
Source: [14]

The decryption algorithm, as represented by Figure 4, receives the SK and the MPDU to be decrypted. First, the IV is extracted from the first 24 bytes of the MPDU and appended to the SK, resulting in the seed for the PRNG. Then N bits are consumed from the keystream and stored in KS, N is the length of the ciphertext and SK the key sequence produced. The ciphertext is XORed with KS, recovering the plaintext concatenated with the ICV. Finally, the plaintext is fed to the IA, calculating ICV'. If the values of the extracted ICV and ICV' are the same, the plaintext is successfully returned as the output of the algorithm. Otherwise, the MPDU was corrupted and an error is sent to the Medium Access Control (MAC) management.

Security The PRNG used by WEP is the Rivest Cipher 4 (RC4) stream cipher [14]. Stream ciphers are susceptible to attacks when a seed is used more than once, generating identical keystreams and effectively encrypting messages with the same key. This makes it possible to recover the seed used on the algorithm by analyzing messages encrypted with equivalent keys.

On WEP, the seed used is the concatenation of the SK, which is constant, and the IV, which is 24-bit long [14]. The maximum number of different seeds that can be used on a WEP network is not high enough, allowing the seed to be easily and rapidly be recovered by eavesdroppers using attacks such as Fluhrer, Mantin, and Shamir (FMS) [8] and Pyshkin, Tews, and Weinmann (PTW) [29], thus compromising the SK and the security of the entire network.

2.3.2 WPA

The WPA is a security protocol based on a draft of the IEEE 802.11i amendment. It was pushed by the Wi-Fi Alliance in 2003 as an intermediate solution to the weaknesses of the WEP algorithm [39], allowing manufacturers to patch existing WEP devices with a safer authentication mechanism while maintaining interoperability. Since its conception, it was meant to be replaced as soon as the IEEE 802.11i amendment was published in its definitive form [43].

WPA moved away from the 40-bit and 104-bit WEP keys and adopted two new authentication methods. WPA-Personal uses a human-readable passphrase as proof of knowledge to authenticate clients. While WPA-Enterprise relies on the IEEE 802.1X standard to authenticate clients on a remote server using one the Extensible Authentication Protocol (EAP) [43]. Both authentication mechanisms result in the Pairwise Master Key (PMK), a key that is used to derive other keys used internally by WPA as foundation for its privacy protocols.

PSK The Pre-Shared Key (PSK) is the authentication protocol of WPA-Personal networks [39]. It derives a 256-bit key DK using the Password-Based Key Derivation Function 2 (PBKDF2) algorithm [15] from a sequence P of 8 to 63 American Standard Code for Information Interchange (ASCII)-encoded characters, between ' ' (32) and '∼' (126) [14]. On PBKDF2 invocation, the Service Set ID (SSID) of the network is used as the salt S, the iteration count c is set to 4096, and the length $dkLen$ of DK is given in bytes.

$$DK = PBKDF2(P, S, c, dkLen)$$

$$\therefore$$

$$PSK = PBKDF2(P, SSID, 4096, 256/8)$$

TKIP The Temporal Key Integrity Protocol (TKIP) is a cipher suite designed to enhance WEP on existing hardware when WPA authentication is configured [14]. TKIP encapsulates the original WEP algorithm to address all security problems known at the time while complying with the first generation stations' constraint of 4 million instructions per second, about 5 instructions per byte processed.

The Message Integrity Code (MIC) was introduced to detect forgery attacks. It is calculated with the Michael algorithm, that due to the computing constraints, couldn't provide the level of effectiveness desired and was complemented with countermeasures on a latter step of TKIP [14]. The Michael

algorithm relies on a key for authentication purposes, either a single 128-bit key is used by both nodes for encryption and decryption or each node has its 64-bit key for encryption, which is shared with the other node for decryption.

The frame order started to be enforced via the TKIP Sequence Counter to avoid replay attacks. Any MPDU received out of sequence is ignored.

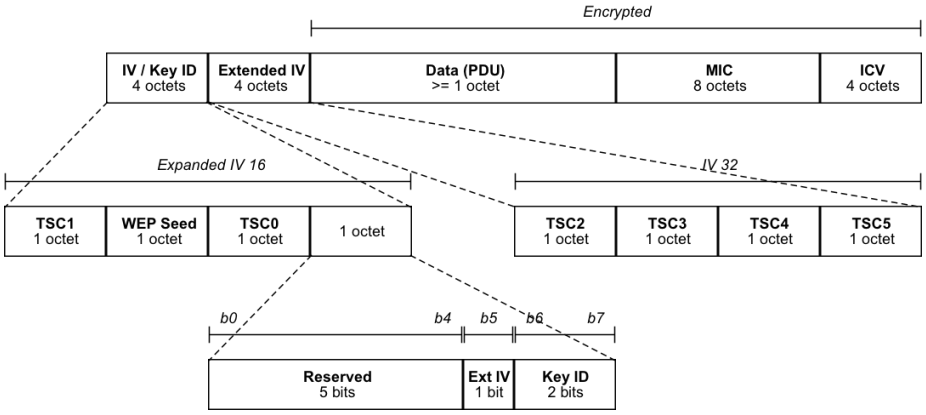


Figure 5: Construction of Expanded TKIP MPDU
Source: [14]

The original WEP MPDU format was extended, as shown in Figure 5, to accommodate additional information required by TKIP, the Extended IV (EIV) and the MIC. The TKIP EIV field is placed right after the original WEP IV field and adds an extra 4 octets to the MPDU. The TKIP MIC increases the MPDU size by 8 octets and is located between the original WEP Protocol Data Unit (PDU) and WEP ICV fields.

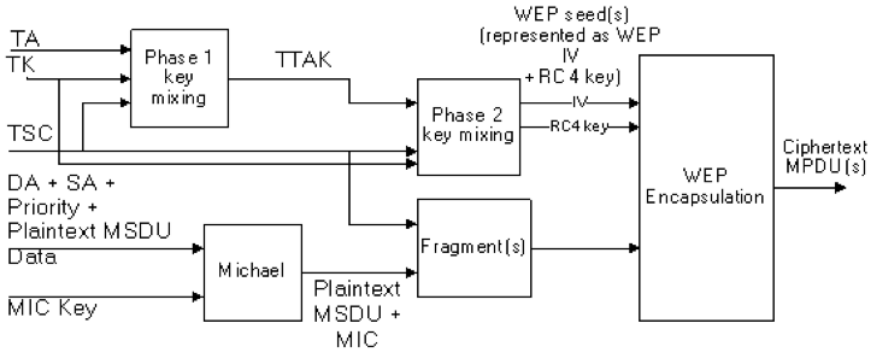


Figure 6: TKIP Encapsulation Block Diagram
Source: [14]

Represented on Figure 6, the encryption process of TKIP combines the Tem-

poral Key (TK), Transmitter Address (TA), and TKIP Sequence Counter (TSC) using cryptographic mixing functions and uses it as the SK and the IV of the WEP algorithm, effectively enhancing the PRNG seed strength by avoiding the reuse of keys. Then TKIP calculates the MIC by executing the Michael algorithm with the MIC key over the MAC Service Data Unit (MSDU) plaintext, priority, Source Address (SA), and Destination Address (DA). The result is fed to the original WEP algorithm, which outputs the MPDU to be transmitted.

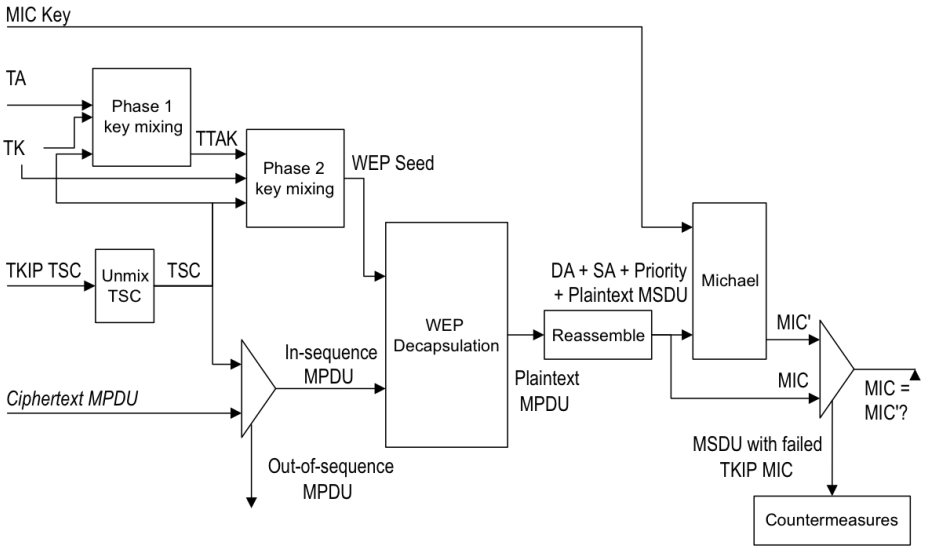


Figure 7: TKIP Decapsulation Block Diagram
Source: [14]

When decrypting, TKIP rejects all MPDUs received out of sequence. Then it calculates the WEP seed using the same mixing algorithm of the encryption process. The WEP algorithm is executed, decrypting the MPDU. Michael is executed with the proper MIC key over the recently recovered data, the result is then compared to the MIC specified in the MPDU. If the MICs don't match, TKIP performs countermeasures procedures to mitigate what looks like an attack. Otherwise, the message is returned by the algorithm, meaning that the MPDU was decrypted and successfully authenticated. The decryption mechanism is illustrated on Figure 7.

Security A compromise was made on the TKIP design to make possible its use on legacy WEP hardware. The forgery attacks were mitigated with the introduction of the Michael algorithm, but, due to computing power constraints, while blocking the forged packets it would cause a denial of service on the network [14].

As TKIP just encapsulates the WEP algorithm, it still relies on the security of the RC4 PRNG. It was found that the keystream generated by RC4 is biased

towards certain sequences and it made practical attacks against WPA-TKIP networks within an hour [32]. The attacker would establish a Transmission Control Protocol (TCP) connection with some victim on the network and would repeatedly send identical packets particularly sized with a well-known content over the connection. Then the wireless traffic was captured and filtered to only what would likely be an attacker's packet. Ciphertext statistics were extracted and plaintext likelihoods calculated using a combination of the Fluhrer-McGrew (FM) and Mantin's Digraph Repetition (AB*S*AB) biases. Finally, the MIC key is derived from one of the candidates with the correct ICV, allowing any other packet of the victim to be fully decrypted.

2.3.3 WPA2

In 2004, the IEEE 802.11i amendment was finalized and published and, as planned, the Wi-Fi Alliance released a new protocol based on the document, WPA2 [39]. Unlike its predecessor, WPA2 wasn't designed to work with older hardware constraints and could provide more robust security mechanisms.

One of the key differences between WPA and WPA2 is its confidentiality and integrity protocol. On WPA it was optional to have the new protocol implemented, while on WPA2 its implementation is mandatory and TKIP should only be used as a fallback for interoperability with older devices.

WPA2 relies on the same authenticators of WPA, PSK for WPA2-Personal and IEEE 802.1X for WPA2-Enterprise.

CCMP The CCM Protocol (CCMP) uses the Advanced Encryption Standard (AES) block cipher operating in CTR with CBC-MAC (CCM) mode. Counter (CTR) provides data confidentiality, while Cipher Block Chaining (CBC)-MAC is used for authentication and integrity. Originally, CCMP supported only 128-bit AES keys (CCMP-128), but later revisions of the algorithm allowed the use of 256-bit keys (CCMP-256) as well.

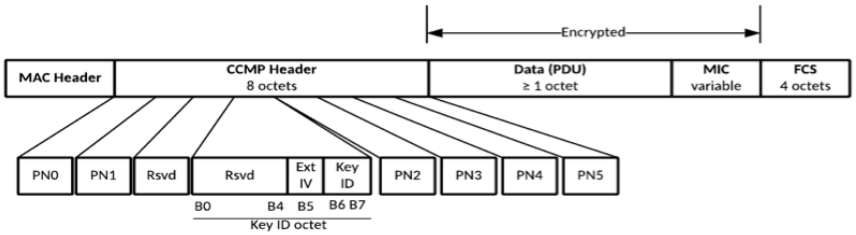


Figure 8: Expanded CCMP MPDU
Source: [14]

Not needing to rely on WEP anymore, CCMP unified the IV field of WEP and EIV field of WPA into the 8-octet CCMP Header field. The WEP ICV

was dropped and the MIC field was turned into a variable size field, 8-octet or 16-octet long for CCMP-128 and CCMP-256 respectively. The resulting MPDU is shown on Figure 8.

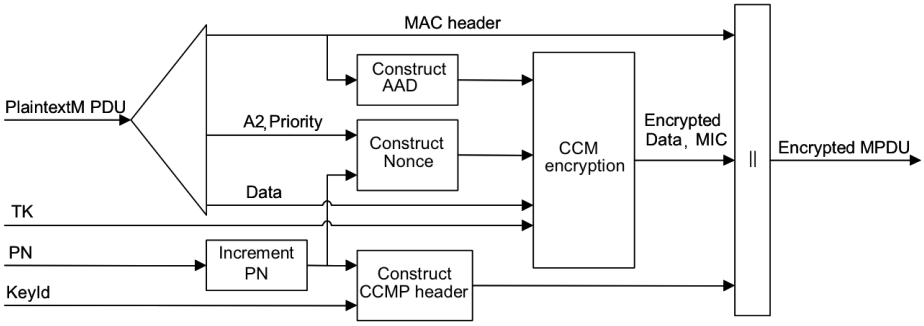


Figure 9: CCMP Encapsulation Block Diagram
Source: [14]

On the encryption process, represented on Figure 9, CCMP maintains a Packet Number (PN) value for each TK, incrementing it sequentially for each MPDU processed. The PN is used along the Key ID (KID) to build the CCMP Header, part of the final MPDU. PN is also used together with the MPDU Address 2 (A2) and Priority of the MPDU being consumed to calculate the Nonce Block, avoiding replay attacks. The MAC Header has its values authenticated by constructing the Additional Authentication Data (AAD), providing integrity protection. Finally, the CCM Originator Processing is invoked with the values of AAD, nonce, plaintext, and TK to form the ciphertext and the MIC. The encrypted MPDU is accordingly assembled and sequentially transmitted.

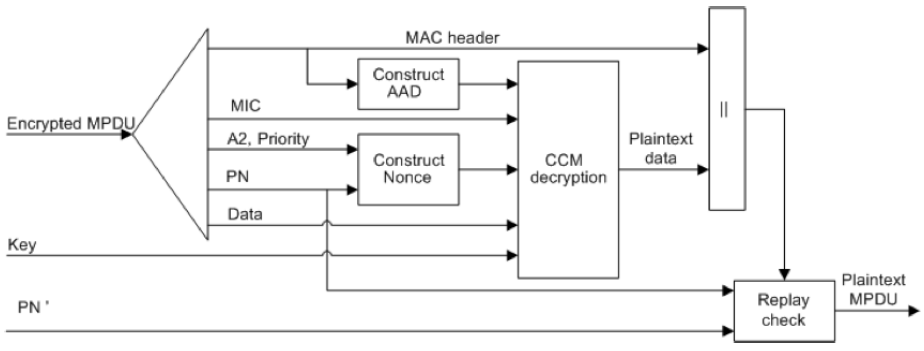


Figure 10: CCMP Decapsulation Block Diagram
Source: [14]

To decrypt the MPDU, the AAD is constructed in the same way as the original MPDU. The Nonce Block uses the A2 and Priority values from MPDU

as usual, but it also gets the PN from there. Then the CCM Recipient Processing is invoked with the TK, the calculated AAD and Nonce Block, and the extracted ciphertext and MIC values. The decrypted MPDU is assembled with the resulting plaintext. Before having the MPDU returned, the replay check is executed, which verifies if the extracted PN value is less than or equal to the local PN value for the TK, implying a frame replay. The decryption process is shown on Figure 10.

Security In 2017, a vulnerability on the 4-way Handshake of WPA was discovered [33]. The 4-way Handshake is designed so that the Station (STA) and the Access Point (AP) can prove to each other that they know the PMK without disclosing it and exchange some information used to derive encryption keys [14]. The vulnerability allowed attackers to reset the value of PN, effectively making CCMP reuse keystreams when encrypting data. Then, if a message encrypted with the reused keystream is known, it is possible to derive the keystream and therefore decrypt any other packet with the same nonce. This problem can be patched, without affecting interoperability with other devices, on the client-side software handling WPA handshakes.

Just like WEP and WPA, WPA2 is vulnerable to offline brute-force attacks. The complexity of the passphrase used is a key aspect of the network's security [39].

AES-related attacks may be performed as well, but their practicality was not demonstrated so far.

2.3.4 WPA3

14 years after the WPA2 release, WPA3 was published. The new protocol is based on the IEEE 802.11-2016 revision and comes with a new handshake process that makes offline dictionary attacks impossible [47]. WPA3 retains interoperability with WPA2 devices via the Transition Mode and must be supported by all Wi-Fi devices certified since July 2020.

SAE The Simultaneous Authentication of Equals (SAE) protocol was originally defined on the IEEE 802.11s amendment for use on mesh networks [14]. It is a variant of the Dragonfly Key Exchange [9], meant to provide an authentication mechanism resistant to offline brute-force attacks. SAE relies on discrete logarithm cryptography to achieve its security properties. The password derivation is made using one of the hash-to-curve functions specified in the protocol.

In the IEEE 802.11-2016 revision, the use of SAE was broadened to act as a replacement for the PSK authenticator of traditional wireless networks, allowing Wi-Fi Alliance to define it as the authentication protocol for WPA3-Personal networks [47].

GCMP The GCM Protocol (GCMP) is a more efficient alternative to CCMP introduced in the IEEE 802.11ac amendment to allow devices to transfer data

at higher speeds with less processing requirements [14]. Just like CCMP, it has a 128-bit (GCMP-128) and a 256-bit (GCMP-256) variants. WPA3 enforces use of GCMP-256 on WPA3-Enterprise networks.

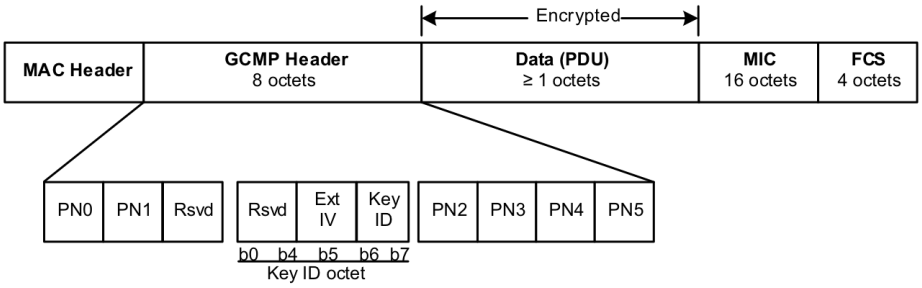


Figure 11: Expanded GCMP MPDU
Source: [14]

Represented by Figure 11, the MPDU of GCMP is very similar to the CCMP one, the MIC size being the only difference between them. Instead of a variant MIC size for 128-bit and 256-bit keys, GCMP's MIC is always 16-octet long.

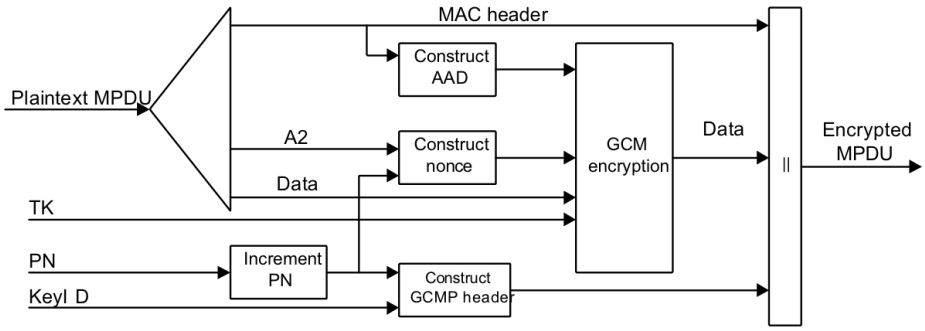


Figure 12: GCMP Encapsulation Block Diagram
Source: [14]

The encapsulation process, shown on Figure 12, differs from CCMP on the Nonce construction. The priority is no longer considered, only A2 and PN still remain. As expected, the CCMP Header construction is replaced with the GCMP Header construction and the CCM Originator Processing with the Galois/Counter Mode (GCM) Originator Processing.

On decapsulation, the CCM Recipient Processing is substituted by the GCM Recipient Processing. The Nonce construction is performed just like in the encapsulation process. The process is shown on Figure 13.

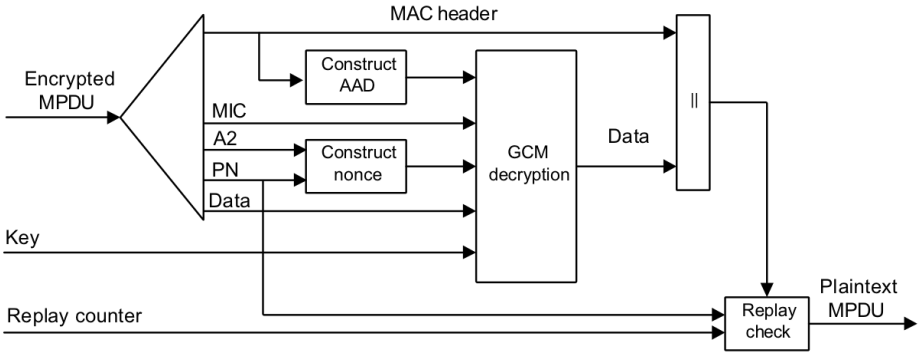


Figure 13: GCMP Decapsulation Block Diagram
Source: [14]

Security The WPA3-SAE Transition Mode was designed to interoperate with WPA2 devices. By doing so, attackers were able to downgrade connections to WPA2 and perform offline dictionary attacks, recovering the network passphrase [34].

In 2019, flaws in the Dragonfly Handshake of WPA3 were found [34]. They allowed offline brute-forcing of passphrases used on WPA3-Personal networks by analyzing the time spent when authenticating with the victim using Brainpool Curves. If the attacker is able to execute user-mode code (e.g. JavaScript on a browser) on a client's device, it is also possible to analyze memory access patterns when the client constructs the commit frame of a Dragonfly Handshake and achieve the same result.

2.4 Network Setup Standards

2.4.1 WPS

Originally called Wi-Fi Simple Config (WSC), the WPS was introduced by Cisco in 2006 as an alternative authentication method for WPA networks. The main goal of this protocol is "to simplify the security setup and management of wireless networks" at home, allowing a client to acquire network credentials with the Push Button Configuration (PBC), by typing a Personal Identification Number (PIN), or via Near Field Communication (NFC) [44].

When WPS is configured to authenticate with a PIN, it uses an 8-digit number. The last digit of the PIN is a checksum of the first 7 digits, so there are 10^7 possible PINs, too many to be brute-forced online. But in 2011 a researcher discovered that WPS reported the correctness of the first and last 4-digits of the PIN individually, making it possible to gain access to the network in, at most, $10^4 + 10^3$ attempts [36]. This flaw was fixed on version 2.0.2 of WPS [44].

In 2014, a problem in the implementation of WPS by several vendors made it feasible to perform an offline brute-force attack [4]. The lack of entropy

allowed the state of the internal PRNG to be derived from information retrieved by partially executing the WPS protocol on the victim, making it possible to compute the E-S1 and E-S2 nonces and subsequently brute-force the PIN. The E-S1 and E-S2 nonces are 128 random bits sent by the AP to the STA that are used to salt the first and second parts of the PIN before hashing it.

2.4.2 WEC

Intended to replace WPS, the WEC protocol uses, mainly, Quick Response (QR) codes or NFC tags to establish the network connection between the client and the station [42]. It was designed with Internet of Things (IoT) devices in mind, standardizing the onboarding process and making it possible to change wireless configurations without having to re-join all devices to the network.

The protocol relies on asymmetric cryptography to create a secure channel between the peers to exchange the credentials. Unlike WPS, there is no PIN and an attacker is not able to take advantage of a legitimate authentication initiation to acquire network credentials.

3 Background in ISP-side Protocols

This chapter explore some protocols that are used by the CPE to communicate with the ISP and the other way around. Two protocols are analyzed: one allows the ISP to manage the CPEs remotely, and the other is used by the CPE to access the voice service of the provider.

3.1 CWMP

CWMP, also known as Technical Report 069 (TR-069), was introduced in 2004 with the intention of defining the communication between a CPE and an Auto-Configuration Server (ACS) [5]. CWMP is commonly used by ISPs to manage residential gateways remotely, without requiring user intervention. The placement of the protocol in the infrastructure is shown on Figure 14.

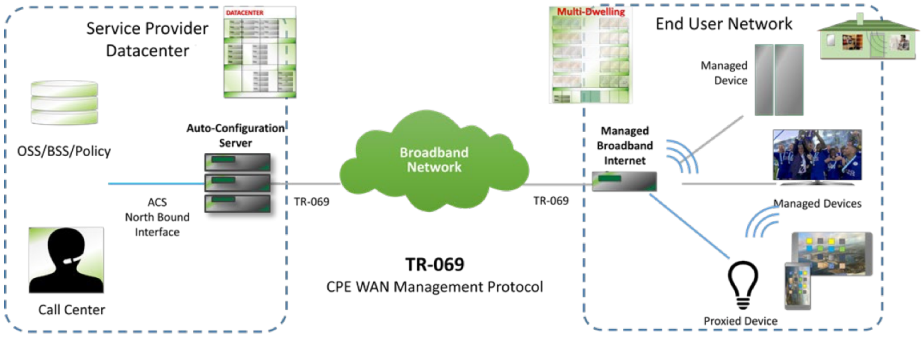


Figure 14: Positioning in the End-to-End Architecture
Source: [5]

CPEs can be provisioned as soon as they connect to the broadband access network and can be re-configured at any time by the ACS. The provides mechanisms for the CPE to download software and firmware updates, self-initiated or ACS-initiated, notifying the ACS upon success or failure. Software can be installed, updated, uninstalled, enabled, or disabled on the CPE via CWMP. The ACS may monitor the CPE's status and performance, also being able to run diagnostic tests and collect the results.

3.2 SIP

SIP is a signaling protocol commonly used by ISPs to establish, maintain, and terminate phone calls over the Internet [23]. Among its responsibilities, it helps routing traffic between the participants and allows the peers to share session descriptions to agree on a set of standards that will be used when transmitting data.

It is designed to work over TCP and User Datagram Protocol (UDP) connections and Transport Layer Security (TLS) can be used to provide a secure

channel for the data. The protocol incorporates several elements from HTTP and its headers are human-readable.

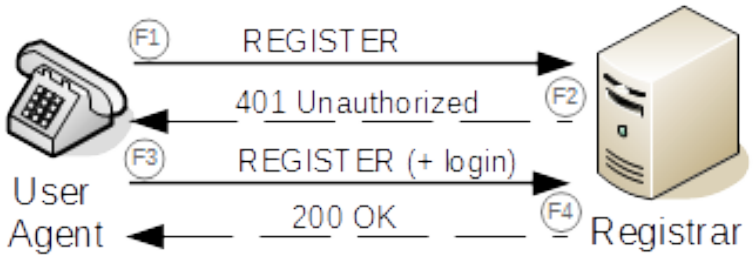


Figure 15: SIP Registration Flow
Source: Korolev Alexandr

Before making and receiving calls, the client must create a session by sending a REGISTER request to the SIP Registrar, shown on Figure 15. The server may request credentials with a challenge-response authentication mechanism, using the Message Digest 5 (MD5) algorithm [22] by default.

With the session established, the client may invite another peer to a call through a SIP Proxy, the receiver is notified and may answer or decline the request. The invite carries all the mechanisms supported to communicate with the caller, leaving the callee to decide which one will be used upon acceptance. The procedure is represented in Figure 16.

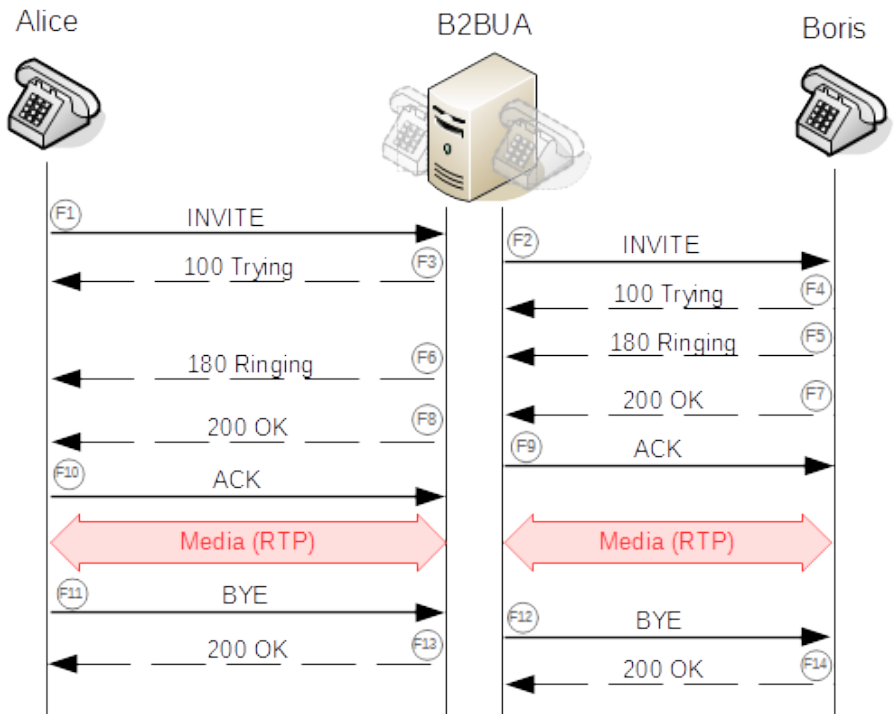


Figure 16: SIP Invitation Flow
Source: Korolev Alexandr

4 CPEs and Research Data

This chapter lists all CPEs being analyzed, including the manufacturer name, the model number, and the transmission medium supported by the device. Each device is assigned an identifier that is used throughout this work.

Then it is explained how each CPE is prepared before any data is collected or any experiment is executed. The state that the CPE is put in resembles its state as soon as the equipment is installed by the ISP on the customer's home.

Finally some data is collected from the device enclosure, the wireless network being broadcasted, the management interfaces, and the configuration file of each CPE. This data is analyzed in the next chapters.

4.1 Subjects

The chosen ISP has two main offerings for their residential Internet service, copper and fiber. For each transmission medium, the ISP deploys different equipment as a residential gateway, that have been majorly manufactured by Askey and MitraStar in recent years.

This study will analyze 8 of those CPEs, as specified on Table 2. An equal number of copper and fiber devices was selected, as well as the number of Askey and MitraStar equipment.

CPE Identifier	Transmission Medium	Manufacturer Name	Model Number
CPE-0	Copper	Askey	`RTA9227W-D112`
CPE-1	Copper	Askey	`RTV9015VW`
CPE-2	Copper	MitraStar	`DSL-100HN-T1-NV`
CPE-3	Copper	MitraStar	`DSL-2401HN2-E1C`
CPE-4	Fiber	Askey	`RTF3505VW-N2`
CPE-5	Fiber	Askey	`RTF8115VW`
CPE-6	Fiber	MitraStar	`GPT-2541GNAC-N1`
CPE-7	Fiber	MitraStar	`GPT-2731GN2A4P`

Table 2: Identifiers of the CPEs

The CPEs analyzed include all models that were leased by the ISP to the customers part of the research. Other models were incorporated into the analysis based on a list of devices on the ISP support page [26].

4.2 Preparation

Before any test was performed, each device had its latest ISP-provided firmware installed and was factory reset. Throughout the research, this is considered the baseline state of each device and represents a CPE that had no intervention since the ISP had it installed on the customer's home.

The firmware for each device was acquired by sending CWMP Inform requests to the ISP ACS on behalf of the CPE with the script of Appendix A. If

the firmware was outdated, then a CWMP Download Firmware Upgrade Image request would be received as the response, as shown in Figure 17.

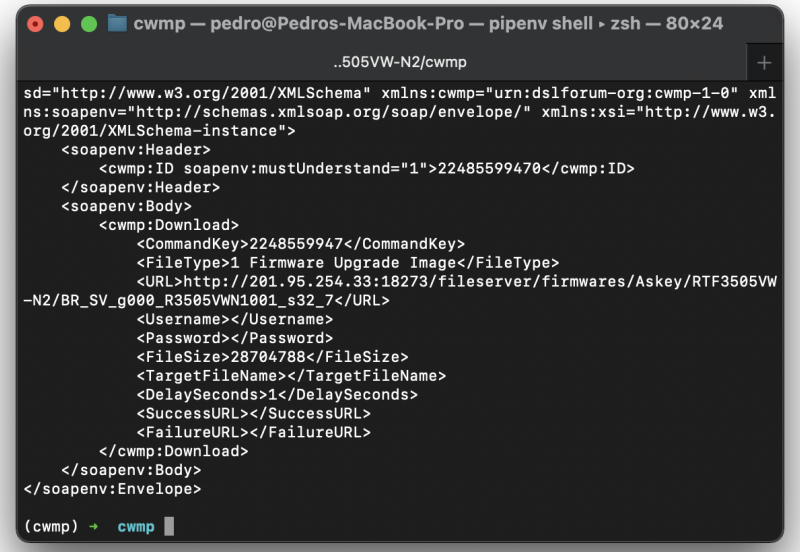


Figure 17: CWMP Download Firmware Upgrade Image

CPE-0 and CPE-1 never received firmware upgrade requests from ACS, no matter what version was set on the CWMP Inform request. Looking into the firmware repository of the ISP, it was found that newer firmware was indeed available for download. But after further inspection, it was verified that both CPEs are not able to have their firmware upgraded.

The experiments on CPE-0 and CPE-1 proceeded with the outdated firmware. All other devices were upgraded via the HTTP management interface with their respective firmware images, as shown on the Table 3. Then, all CPEs were factory reset by holding the reset button, located on the rear of each device, for 10 second.

With the devices set to their baseline, a computer was connected to one of CPE's Local Area Network (LAN) ports via an Ethernet cable and was able to acquire an Internet Protocol (IP) address from the Dynamic Host Configuration Protocol (DHCP) server running on the CPE, as shown in Figure 18. The gateway and Domain Name System (DNS) servers were set to the CPE's IP address, `192.168.15.1` for all CPEs in the research.

CPE Identifier	Firmware Version
CPE-0	BR_SO_RTK_V5.1.8j
CPE-1	BR_SO_RTK_V6.1.8t
CPE-2	BR_SA_113WUK0b8
CPE-3	BR_SA_g003_114WUQ0b18
CPE-4	BR_SV_g000_R3505VWN1001_s32_7
CPE-5	BR_SG_g11.11_RTF_TEF001_V6.54_V014
CPE-6	BR_SV_1.00(VNZ.0)b18
CPE-7	BR_SV_1.11(WVK.0)b18

Table 3: Firmwares of the CPEs

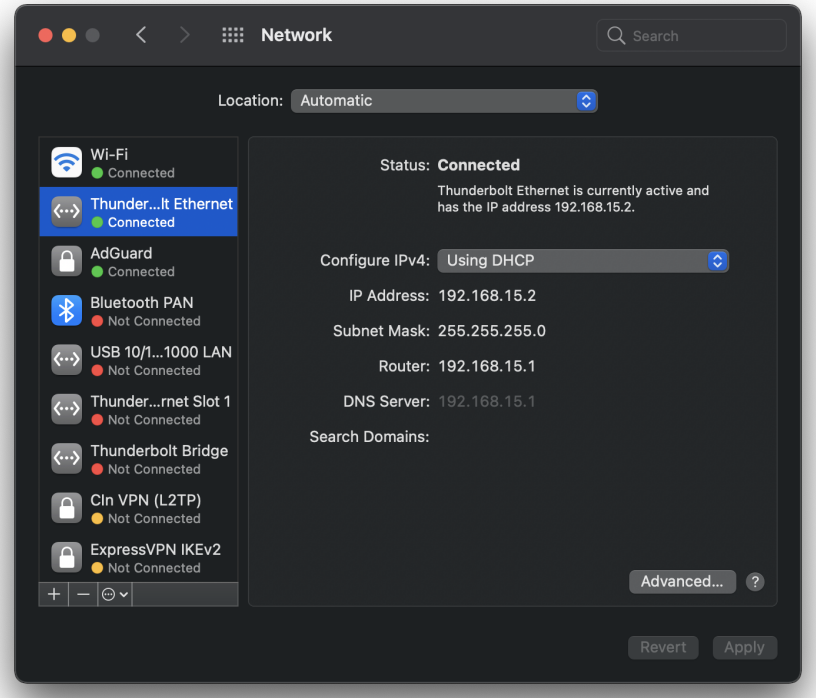


Figure 18: Wired Connection to the CPE

4.3 Operating System

Most of the CPEs run on the 32-bit Microprocessor without Interlocked Pipelined Stages (MIPS) version of the Linux Kernel. The identification process was made by either executing commands on the device's shell, as shown in Figure 19, or by inspecting the firmware image installed. Unfortunately, not all devices had their operating system identified, they most likely don't run Linux.

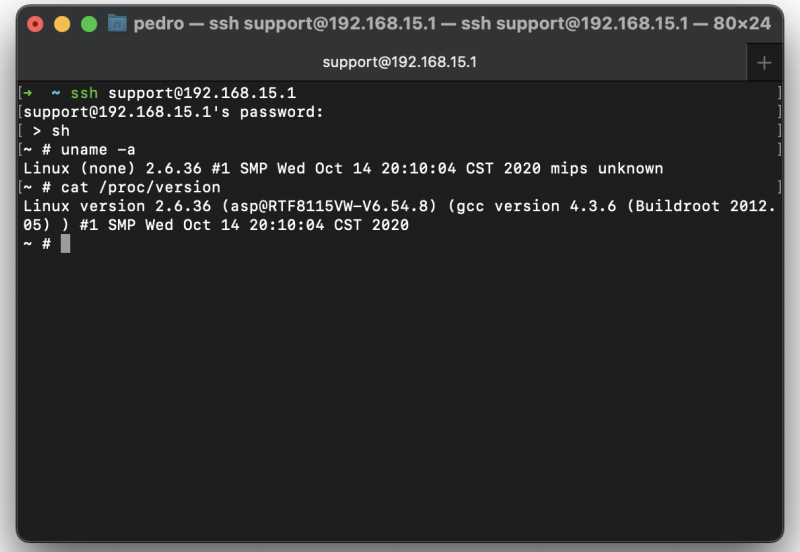


Figure 19: Checking CPE Operating System

Table 4 presents the operating system identified on each CPE.

CPE Identifier	Operating System
CPE-0	unknown
CPE-1	unknown
CPE-2	unknown
CPE-3	unknown
CPE-4	`Linux version 3.4.11-rt19+ (openwrt@sw3-pc) (gcc version 4.6.2 (Buildroot 2011.11) ) #1 SMP PREEMPT Thu Jun 18 13:46:11 CST 2020`
CPE-5	`Linux version 2.6.36 (asp@RTF8115VW-V6.54.8) (gcc version 4.3.6 (Buildroot 2012.05) ) #1 SMP Wed Oct 14 20:10:04 CST 2020`
CPE-6	`Linux version 3.4.11-rt19 (andy@iBuild) (gcc version 4.6.2 (Buildroot 2011.11) ) #1 SMP PREEMPT Mon Dec 11 17:11:19 CST 2017`
CPE-7	`Linux version 2.6.36 (john@iBuild) (gcc version 4.3.6 (Buildroot 2012.05) ) #2 SMP Thu Aug 16 19:00:39 CST 2018`

Table 4: Operating Systems of the CPEs

4.4 Wireless Configuration

While the CPEs were turned on, iwlist and wash tools were used on a computer within range of the networks to inspect their configuration. The information printed on the label of each device was used to associate the Wi-Fi networks with the CPEs.

The information collected is presented on Table 5 and includes the Network Basic SSID (BSSID), the Network Extended SSID (ESSID), the Frequency Band, the Security Protocol, the Pairwise Cipher, the Authentication Suite, the Network Passphrase, the WPS Version, and WPS Authentication Method.

CPE Identifier	Network BSSID	Network ESSID	Frequency Band	Security Protocol	Pairwise Cipher	Authentication Suite	Network Passphrase	WPS Version	WPS Authentication Method
CPE-0	5C:C9:D3:D8:73:1A	VIVO-731B	2.4 GHz	WPA + WPA2	TKIP + CCMP	PSK	C9D3D8731B	2.0	PBC
CPE-1	54:2F:8A:6C:BE:98	VIVO-BE99	2.4 GHz	WPA + WPA2	TKIP + CCMP	PSK	2F8A6CBE99	2.0	PBC
CPE-2	34:57:60:13:11:CC	VIVO-11CC	2.4 GHz	WPA + WPA2	TKIP + CCMP	PSK	57601311CC	2.0	PBC + PIN
CPE-3	C0:3D:D9:53:32:08	VIVO-3208	2.4 GHz	WPA + WPA2	TKIP + CCMP	PSK	t4v4AysytY	2.0	PBC
CPE-4	10:72:23:0D:83:CA	VIVOFIBRA-83CC	2.4 GHz	WPA2	CCMP	PSK	72230D83CC	2.0	PBC
CPE-4	06:72:23:0D:83:D7	VIVOFIBRA-83CC	5 GHz	WPA2	CCMP	PSK	72230D83CC	2.0	PBC
CPE-4	10:72:23:0D:83:D7	VIVOFIBRA-83CC-5G	5 GHz	WPA2	CCMP	PSK	72230D83CC	2.0	PBC
CPE-5	98:7E:CA:44:F8:8F	VIVOFIBRA-F881	2.4 GHz	WPA2	CCMP	PSK	uzrUaxS565	2.0	PBC + PIN
CPE-5	86:7E:CA:44:F8:8E	VIVOFIBRA-F881	5 GHz	WPA2	CCMP	PSK	uzrUaxS565	2.0	PBC + PIN
CPE-5	98:7E:CA:44:F8:8E	VIVOFIBRA-F881-5G	5 GHz	WPA2	CCMP	PSK	uzrUaxS565	2.0	PBC + PIN
CPE-6	AC:C6:62:B3:E3:80	VIVOFIBRA-E37E	2.4 GHz	WPA2	TKIP + CCMP	PSK	c662b3e37e	2.0	PBC
CPE-6	AC:C6:62:B3:E3:87	VIVOFIBRA-E37E-5G	5 GHz	WPA2	CCMP	PSK	c662b3e37e	2.0	PBC
CPE-7	A4:33:D7:53:2C:C8	VIVOFIBRA-2CC8	2.4 GHz	WPA + WPA2	TKIP + CCMP	PSK	6C383DA8BA	2.0	PBC
CPE-7	A4:33:D7:53:2C:CF	VIVOFIBRA-2CC8-5G	5 GHz	WPA2	CCMP	PSK	6C383DA8BA	2.0	PBC

Table 5: Wireless Networks of the CPEs

4.5 Allowed Ingress Traffic

The nmap tool [11] was used to look for open TCP and UDP ports on the CPE under analysis. Additionally, the ping tool [12] was also executed to check if Internet Control Message Protocol (ICMP) traffic was allowed.

First, both tools were used from inside the LAN towards the CPE's private IP, listing which ports were exposed to the local network. The execution of both tools is shown on Figure 20.

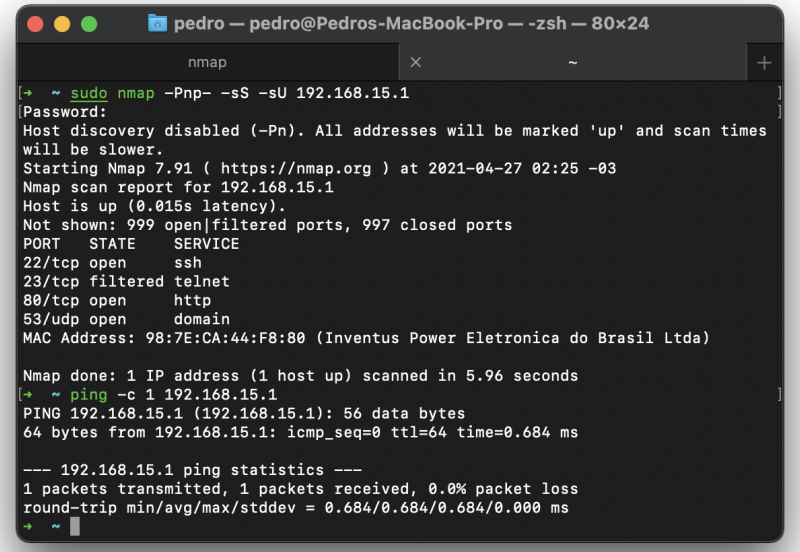


Figure 20: CPE Allowed Ingress Traffic

Then, to detect the ports exposed to the broadband network, the CPE had to be connected to the copper or fiber cable of the ISP to acquire a public IP address. The tools were executed from a device on the Wide Area Network (WAN), directing the scan to the CPE's public IP.

Table 6 presents all ingress traffic allowed by the CPE on both LAN and WAN sides.

CPE Identifier	From LAN-side	From WAN-side
CPE-0	53/udp 80/tcp 443/tcp 5431/tcp ICMP	7547/tcp ICMP
CPE-1	53/udp 80/tcp 443/tcp 5431/tcp ICMP	7547/tcp ICMP
CPE-2	22/tcp 53/udp 80/tcp ICMP	22/tcp 80/tcp 7547/tcp ICMP
CPE-3	22/tcp 53/udp 80/tcp ICMP	22/tcp 80/tcp 7547/tcp ICMP
CPE-4	53/udp 80/tcp ICMP	7547/tcp ICMP
CPE-5	53/udp 80/tcp ICMP	7547/tcp ICMP
CPE-6	80/tcp 443/tcp ICMP	7547/tcp ICMP
CPE-7	53/udp 80/tcp 5555/tcp ICMP	7547/tcp ICMP

Table 6: Allowed Ingress Traffic of the CPEs

4.6 Outgoing Traffic

CPEs 4 and 7 have tcpdump [30] installed, allowing the outgoing traffic to be inspected from inside the device. The tcpdump binary was introduced on CPE 5 via a Trivial File Transfer Protocol (TFTP) server [24], allowing the traffic to be inspected as well.

CPEs 0 to 3 and 6 neither have tcpdump installed nor are able to have it introduced on them by any means. On CPEs 0 and 1, the only data that could be extracted were snapshots of the TCP Translation Table.

With the environment set, the capture is started and the broadband connection is provided to the CPE, allowing it to initiate connections.

4.6.1 CWMP

On all CPEs was observed CWMP traffic to the ISP ACS in a very similar pattern. The default configuration points to an intermediary ACS with hardcoded credentials. Table 7 presents all different configurations found on the CPEs.

ACS Uniform Resource Locator (URL)	ACS Username	ACS Password
http://acs.telesp.net.br:7005/cwmpWeb/WGCPEMgt	mitrastar	telefonica
http://acs.telesp.net.br:7005/cwmpWeb/WGCPEMgt	acsclient	telefonica
https://acs.gvt.net.br	acsclient	acsvivo15sca
https://acs.gvt.net.br	acsclient	acsgvt25sca

Table 7: Hardcoded ACS Credentials of the CPEs

The CPE sends a CWMP Inform request that is answered with an ACS request for changing Connection Request Credentials and enabling Periodic Inform with an interval of 108000 seconds. The new Connection Request username set by the ACS is a combination of CPE Serial Number, CPE Product Class, and CPE Organizationally Unique Identifier (OUI), respectively concatenated with a dash used as a separator. The password is a 128-bit number encoded as a hexadecimal string, likely the MD5 hash of an unknown value. A new password is issued everytime that the CPE registers itself using the default CWMP

settings. Table 8 presents the credentials acquired by each CPE in one of the registrations with the default ACS.

CPE Identifier	Connection Request Username	Connection Request Password
CPE-0	5CC9D3D8731B-RTA9227W-D112-C0D962	4f6a306dc705bab62a07415460231bec
CPE-1	542F8A6CBE99-RTV9015VW-C0D962	b10f38755e844832aa586ccca67572b7
CPE-4	1072230D83CC-RTF3505VW-N2-009096	7f6fc3cb94341ace99154184e71cd0eb
CPE-5	987ECA44F881-RTF8115VW-009096	da9e89b37c32650c44c826db21f54bc8

Table 8: Connection Request Credentials of the CPEs

After performing the requested changes and notifying the ACS about it, the ACS now requests the ACS URL and Credentials to be changed. The final ACS used by all CPEs is the same. It was verified that the username and passwords used to access the ACS are the UNIX Timestamp of the time that the credentials were issued, respectively appended by characters 'u' and 'a', as presented in Table 9.

ACS URL	ACS Username	ACS Password
http://acs.telesp.net.br:7015/cwmpWeb/CPEMgt	${UNIX_TIMESTAMP}u	${UNIX_TIMESTAMP}a

Table 9: Assigned ACS Credentials of the CPEs

A CWMP Inform request is sent to the new ACS. If the device was running an outdated firmware at the time of the request, the ACS will respond if a CWMP Download request with the Firmware Upgrade Image. In all cases, the firmware image was hosted under `http://201.95.254.33:18273/fileserver/`. The server seems to be accessible only from an IP address that belongs to the ISP, other addresses experience timeouts when requesting data.

4.6.2 VoIP

SIP traffic was identified on CPEs that were bound to customers subscribed to the VoIP service of the ISP. The captured traffic shows that TLS is not used and the transport protocol utilized is UDP. The destination of the packets was `192.168.80.1:5060`, which acts as a SIP Outbound Proxy.

By inspecting the content of SIP requests, it was able to verify that `ims3.gvt.net.br` was being used as the SIP registrar and the SIP realm. For the username, the customer's phone number formatted with the E.164 standard was used. The password could not be retrieved in plaintext, as it is sent hashed and would require brute-forcing to reveal its contents.

4.6.3 IPTV

Unfortunately, it wasn't possible to capture IPTV data as the test environment doesn't have a subscription to the ISP IPTV service. But it was possible to

identify that similar to the VoIP service, the IPTV has a dedicated Virtual LAN (VLAN) for it, number 602, on fiber-based CPEs.

4.7 Anonymously Accessing Management Interfaces

All devices have their HTTP Management Interface enabled by default and can be accessed by navigating to `http://192.168.15.1` on a browser. This interface provides information about the CPE and its network.

On the main page, Figure 21, both public and private IP addresses of the device can be viewed together with the gateway and DNS servers configured. The SSID, security protocol, WPS state, and channel of the wireless networks are specified, also showing which LAN ports are in use. A list with the hostname, MAC address, and IP of all wired and wireless clients currently connected to the CPE is present. If VoIP is configured, the telephone number is shown.

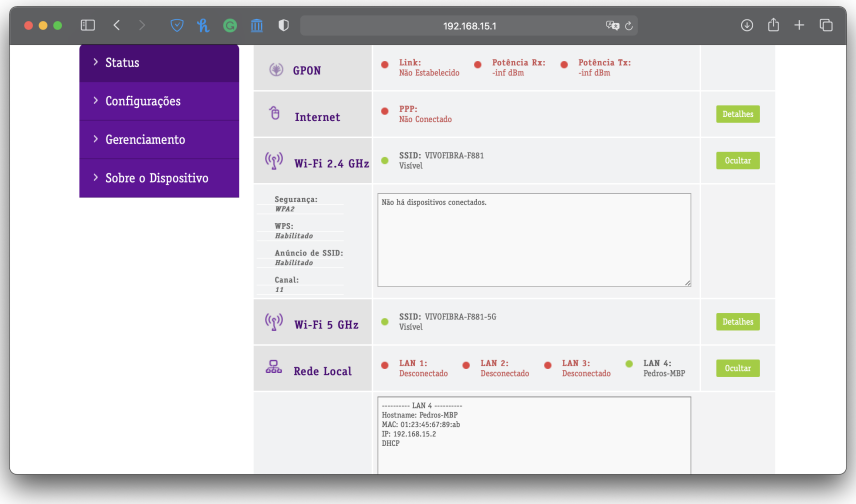


Figure 21: Main Page of the CPE HTTP Management Interface

The about page of the interface, Figure 22, shows the manufacturer, model, firmware version, hardware version, serial number, WAN-side MAC address, and LAN-side MAC address. If the device is fiber-compatible, it also shows the Gigabit Passive Optical Network (GPON) Serial Number (S/N).

The Secure Shell Protocol (SSH) and Telnet interfaces, even when manually enabled, don't expose any data about the device without authenticating, immediately requesting credentials.

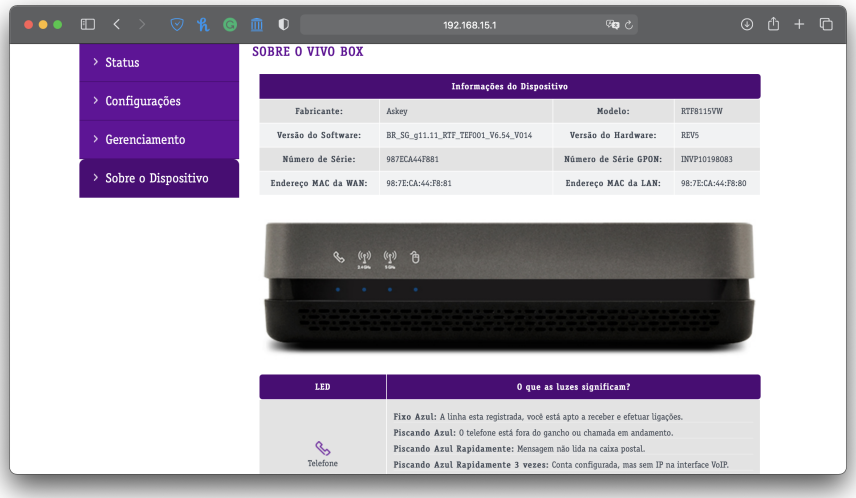


Figure 22: About Page of the CPE HTTP Management Interface

4.8 Configuration Export

None of the CPEs support exporting the configuration file on the standard HTTP Management Interface, even when signing in with the admin user. But a hidden interface is available in all devices and it allows the configuration file to be exported. To access this interface, you must navigate to `http://192.168.15.1/padrao` and log into the support user, using the admin's password.

On CPEs 2 to 7, you can navigate through the menu and export the configuration in the proper page with the support account. But CPEs 0 and 1 don't have the export page indexed in the menu, and you must access `http://192.168.15.1/saveconf.htm` manually, shown in Figure 23.

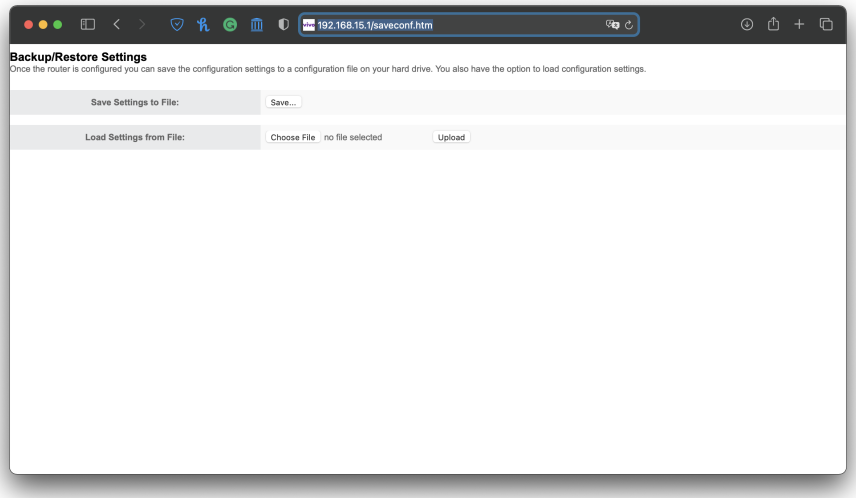


Figure 23: Unindexed CPE Configuration Export Page

4.9 EPROM Extraction

The CPEs don't have any functionality intended to dump their Erasable Programmable Read-Only Memory (EPROM) via software. But it was discovered that by accessing CPEs 4 to 7 via SSH or Telnet, you are either able to read the memory area where the EPROM is mapped or read the Memory Technology Device (MTD) partition with TFTP and dump it to a server. The command bellow can be executed on compatible CPEs to have the first EPROM partition transfered to `192.168.15.2`, a TFTP server.

```
tftp -l /dev/mtd0 -p 192.168.15.2
```

To list all EPROM partitions, the `/proc/mtd` file can be read, as shown in Figure 24.

Although it is possible to read the EPROM of CPE 6 using the TFTP technique, the device's shell sets a small time limit to processes executed from it, making it impossible to complete the transfer before the limit exceeds. To work around this problem, an EPROM programmer was directly attached to the CPE's EPROM, allowing a computer to dump its contents via the programmer.

On CPEs 0 and 1, there is a command that shows the memory addresses where the EPROM is mapped and there is another command is capable of reading the content of an arbitrary memory address. Combining both commands, it is possible to extract the full EPROM content from a machine capable of accessing the CPEs via Telnet. The scripts used are attached on Appendix B.

For CPEs 2 and 3, the same technique of using an EPROM programmer had to be used. No command capable of dumping the EPROM via software

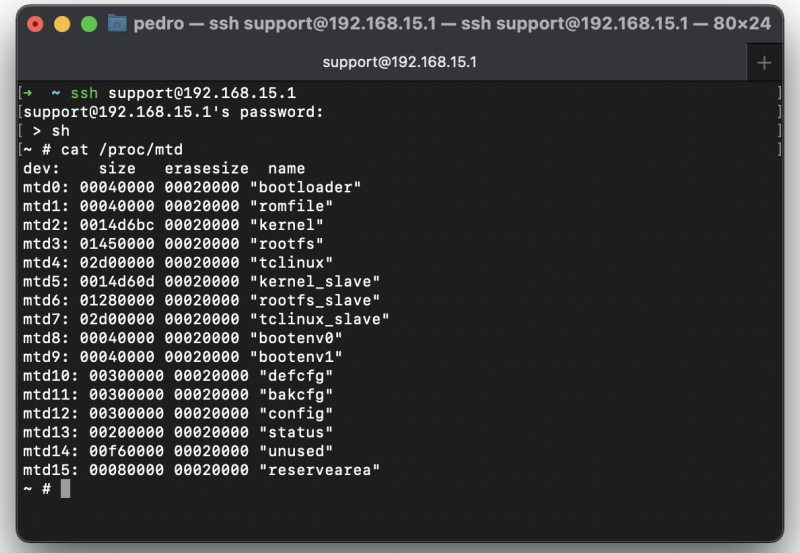


Figure 24: File /proc/mtd of CPE 5

was found.

5 Configuration Analysis

The chapter aims to analyze the configuration set on the CPEs by the ISP. All data was collected as specified in the previous chapter. If some setting poses a threat to the user and its devices, the issue is discussed and possible remediations are explained.

5.1 Configuration File

All devices in the research allow a configuration file to be exported and later imported. These configuration files expose additional settings that are not available through the management interfaces. Most of the CPEs export it as plaintext, but CPEs 3, 5, and 7 encrypt the file. Some fields of the file exported from CPE 2 are also encrypted.

The fields encrypted on CPE 2 configuration file are hex-strings prefixed by `_encrypted_`. When decoding the strings, it always started with the characters `A`, `E`, `S`, and `\0`, indicating that AES is the cipher used to protect the original content. It was detected that the same value was encrypted twice in the configuration file producing distinct ciphertexts, indicating that either different keys are used to encrypt different fields or the key is salted upon encryption. Unfortunately the decryption key was not found and it is unknown.

The encryption used on CPEs 3 and 7 was identified by binwalk [20] to be OpenSSL with a salted password. Just like on CPE 2, the key could not be identified.

CPE 5 was different, the exported configuration file carries plaintext data mixed with ciphertext data. Originally it was speculated that the file was using some non-standard compression algorithm as binwalk wasn't able to identify the file type. But there is a tool that can be used to decode the configuration file when accessing the device via SSH or Telnet, and after decoding the exported file, it became smaller. So the file was indeed encrypted and not compressed. While the tool allowed the file to be modified and restored on the device, the decryption binary cannot be executed outside the device, as it communicates with a character device managed by an unknown kernel module. But by decompiling the binary and their dependencies using Binary Ninja [35] and Ghidra [16], it was able to recover the encryption and decryption algorithm used by the CPE, as shown in Figure 25. By simply adding or removing a 16-byte header from the configuration file and XORing every byte with 8, the file can be encrypted or decrypted. The decryption function, reimplemented in Python, is available on Appendix C.

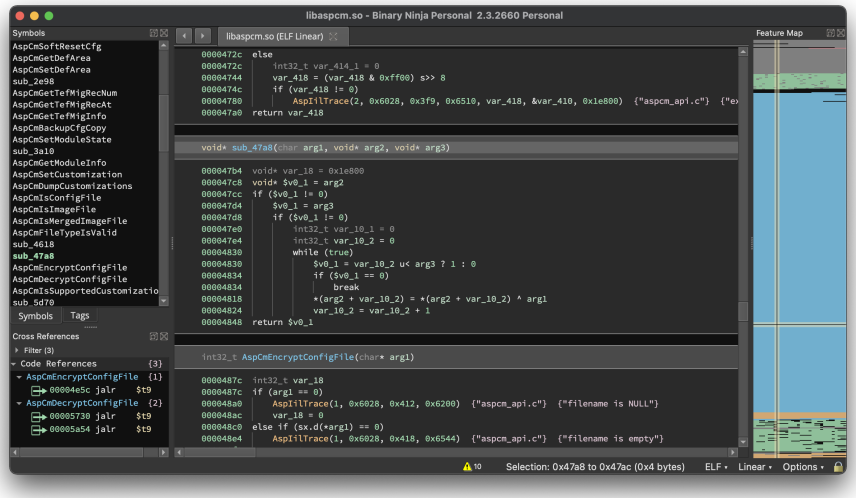


Figure 25: Function `sub_47a8` of `libaspcm.so` Decompiled by Binary Ninja

5.2 Wi-Fi

The CPEs analyzed use a very common technique to assign different ESSIDs to distinct CPEs. The last 4 hexadecimal digits of the MAC address of one of its network interfaces are appended to a fixed prefix, generating a reasonably unique name for the Wi-Fi networks. Devices that support the 5 GHz band add a `-5G` suffix to the generated name and assign it to the ESSID of the faster network, distinguishing it from the 2.4 GHz network. CPEs 4 and 5 also advertise the 5 GHz network without the `-5G` suffix, with intent to upgrade compatible devices on the 2.4 GHz band to the 5 GHz band when appropriate.

Almost all CPEs in the research use a common, but unsafe, derivation method for the Wi-Fi passphrase by using part of the BSSID of the network. On CPEs 2, the password is the last 10 hexadecimal digits of the BSSID. The password for CPEs 0, 1, 4, and 6 is the substring from the third to the tenth hexadecimal digit of the BSSID appended by the last to characters of the 2.4 GHz network ESSID. CPEs 0, 1, 2, and 4 use the uppercase version of the hexadecimal MAC address, while CPE 6 uses the lowercase version.

CPEs 3, 5, and 7 have passphrases that could not be identified to be derived from any other information of the equipment. But, by analyzing other CPEs of the same model, it was confirmed that their values follow a pattern.

Given that none of the CPEs are compatible with WPA3, an offline brute-force attack can be performed. With the patterns found on passwords of CPEs 3, 5, and 7, the attack may succeed in a reasonable time.

Additionally, CPEs 0 to 4 are using WPA as security protocol alongside WPA2. Meaning that even with a strong passphrase, it is not possible to ensure

that the network is safe. CPEs 6 and 7 are susceptible to the same class of problems on their 2.4 GHz network, WPA is enabled on CPE 7 and CPE 6 strangely enables TKIP with WPA2 even though WPA support is disabled.

Table 9 presents the patterns found on the passphrase of each CPE. It also indicates the maximum brute-force time needed to crack the password on a p3.2xlarge Amazon Web Services (AWS) Elastic Compute Cloud (EC2) Instance [2] given the benchmark presented in Figure 26. The CPEs with 0s brute-force time can have their password derived from other information in constant time.

CPE Identifier	Passphrase Pattern	Brute-Force Time
CPE-0	`upper(MAC[2:10] + ESSID[-2:])`	0s
CPE-1	`upper(MAC[2:10] + ESSID[-2:])`	0s
CPE-2	`upper(MAC[2:])`	0s
CPE-3	`[0-9A-Za-z]{10}`	565871h
CPE-4	`upper(MAC[2:10] + ESSID[-2:])`	0s
CPE-5	`[0-9A-Za-z]{10}`	565871h
CPE-6	`lower(MAC[2:10] + ESSID[-2:])`	0s
CPE-7	`[0-9A-F]{10}`	45min

Table 10: Wireless Network Passphrase Pattern of the CPEs

Additionally, all CPEs have WPS enabled by default and allow authentication via PBC and CPEs 2 and 5 also can authenticate via PIN. The CPEs were not found to be vulnerable to a WPS brute-force attack.

The PIN used on CPE 2 is a random number that could not be recognized as a derivation of other information and was found in the backup partition of the EPROM, possibly indicating that it could have been assigned while in the factory. The PIN can be reset to a random value via the HTTP management interface and persists between reboots, but the original value is restored when the device is factory reset. After multiple resets, none of the generated PINs do not seem to relate to previously generated PINs, indicating that the PRNG is not reset to a known state on factory restores.

On CPE 5, there is no indication on its web interface that a WPS PIN can be used to authenticate to the network as well. Furthermore, the PIN is set to 12345670 by default and there is no procedure to regenerate the PIN or disable it via regular means. It was only possible to change WPS settings by exporting the configuration file, decrypting it, modifying the PIN value or disabling the authentication method, encrypting the modified file, and finally importing it back to the CPE.

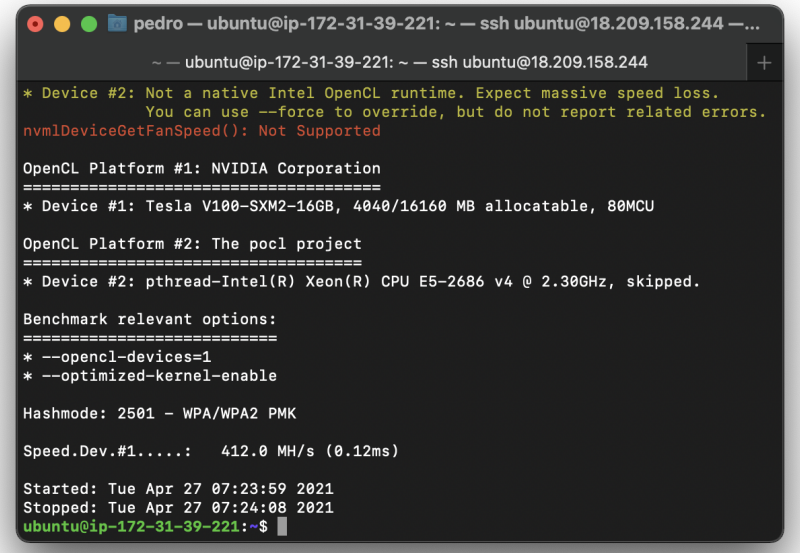


Figure 26: Hashcat WPA Benchmark on a p3.2xlarge AWS EC2 Instance

5.3 Management Interfaces

The CPEs analyzed have two users registered, admin and support. All management interfaces can be accessed using these users, but permissions may differ.

Both users share the same password, that is printed on each device label. Although patterns have been recognized in the password of all CPEs, as presented in Table 10, none of them were identified to be derived from other information.

An online brute-force attack was performed on the HTTP Management Interfaces of the CPEs by writing a C program that opens a socket with the server and sends requests as fast as possible, but the results indicate that it is not feasible to exploit it with this technique. The HTTP server is slow and seems to only process one request at a time, the performance decreased significantly when parallelizing the attack due to the sequential processing of the HTTP server.

Although, if some way of extracting the hashed passwords from the devices is discovered, most of the passwords could be cracked in a reasonable amount of time giving the pattern constraints. The kind of vulnerability that allows the hash extraction is not common, but it has been on some Commercial Residential Gateways (CRGs) in recent years.

CPE Identifier	Password Pattern
CPE-0	`[0-9A-Za-z]{8}`
CPE-1	`[0-9A-Za-z]{8}`
CPE-2	`[0-9a-z]{8}`
CPE-3	`[0-9a-z]{8}`
CPE-4	`[0-9a-f]{8}`
CPE-5	`[0-9a-z]{8}`
CPE-6	`[0-9a-f]{8}`
CPE-7	`[0-9a-f]{8}`

Table 11: Management Interface Password Pattern of the CPEs

5.4 PPPoE

The Internet connection on VLAN 600 uses Point-to-Point Protocol over Ethernet (PPPoE) [6] for tunneling its packages. The protocol requires the user to authenticate and then can provide encryption and compression for the connection. On the CPEs being studied, the authentication mechanism of PPPoE is not really used for client authentication. Each customer is identified based on which port the copper or fiber cable connects to on the ISP side.

The PPPoE credentials are hardcoded on all CPEs analyzed. The HTTP Management Interface explicitly instructs the user to set `cliente@cliente` as the username and `cliente` as the password, as seen in Figure 27.

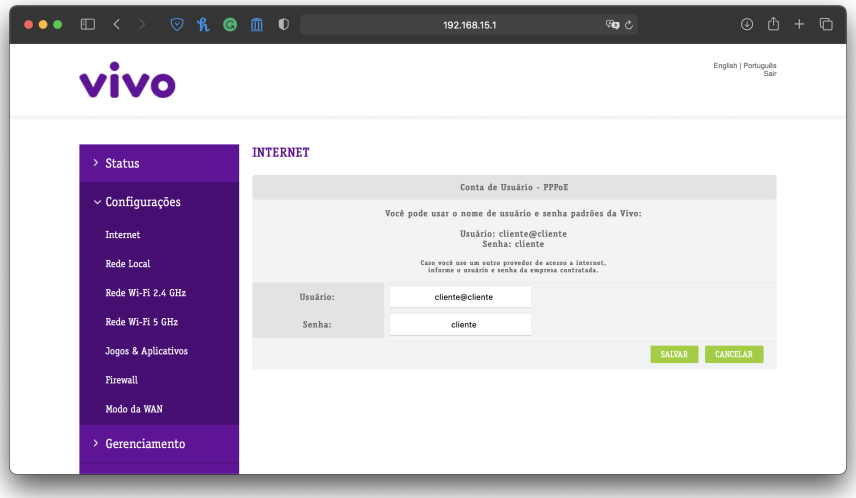


Figure 27: PPPoE Settings on CPE

5.5 SIP

The captured SIP traffic shows that it is unencrypted and the transport protocol used is UDP [19]. Most of the parameters to establish a connection to the VoIP server of the ISP can be extracted from the traffic. Only the password is not available as plaintext and the hashed value is calculated with a nonce that makes it usable only once.

But by inspecting the configuration files and management interfaces of the CPEs, it was possible to retrieve the plaintext SIP password. All the passwords extracted had length 16 and contained only alphanumeric characters. After further analysis, it was found that the last 8 characters of the password are always the first 8 characters with the case swapped. This pattern effectively reduces the strength of the password by a square root, so only 62^8 different passwords exist. A iterator that generates all possible passwords has been implemented and is available in Appendix D.

Running a SIP Digest Authentication benchmark with hashcat on a p3.2xlarge AWS EC2 Instance [2], showed that the machine is able to brute-force the password on the captured SIP packets at a rate of 7.5 GH/s, as show in Figure 28. It means that with an expense of no more than $24.64, in the worst-case scenario, the weakened password can be cracked in 8h4min.

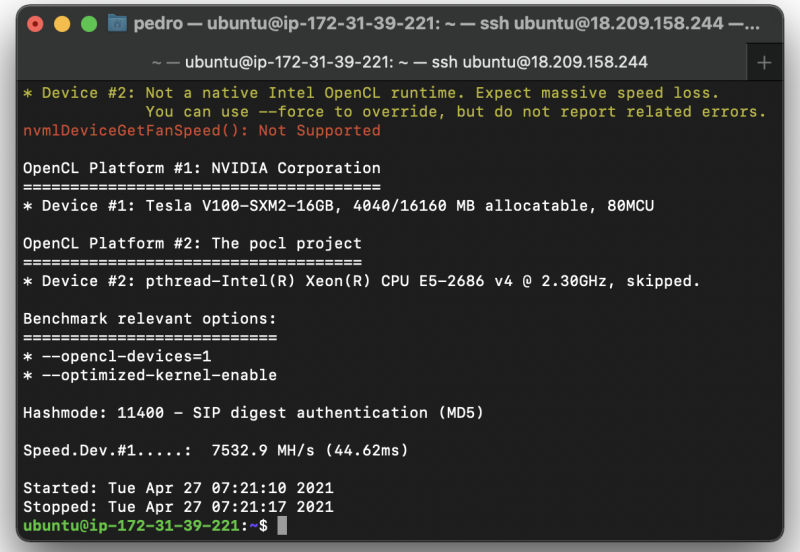


Figure 28: Hashcat SIP Benchmark on a p3.2xlarge AWS EC2 Instance

Enabling TLS [21] would be an option to mitigate the capture of plaintext

SIP requests, but it is not possible because the SIP registrar doesn't support the encrypted protocol. Unfortunately, the customer itself is also unable to change the SIP password, only the ISP can change the password and it is unlikely that it can be set to a custom value or a value without the known pattern. Rotating the password would only mitigate an ongoing attack where the attacker has captured the hashed password but is not able to acquire it anymore. The strength is not increased and if the hash is obtained at a later point in time, it can easily be cracked again.

6 HTTP Management Interface

All CPEs researched have an HTTP Management Interface and it is enabled by default. The interface is a web service and as such, it is susceptible to a wide range of attacks if not properly implemented. Additionally, the management interface is accessible from the WAN-side on some devices. These factors make it an excellent attack vector.

This chapter analyses the different security aspects of the HTTP Management Interface as implemented on each CPE. The problems are assessed and the impact discussed.

6.1 Session

The CPE needs to keep the user authenticated while navigating through the management interface, this is generally done with HTTP Cookies [3].

By analyzing the requests made by the browser when accessing the interface of CPEs 4, 6, and 7, it was found that the cookies contain only 8 to 10 characters, as shown on Figure 29. It seems to be risky, but given that the sessions are short lived and that the HTTP server is slow, hardly a brute-force attack would succeed.

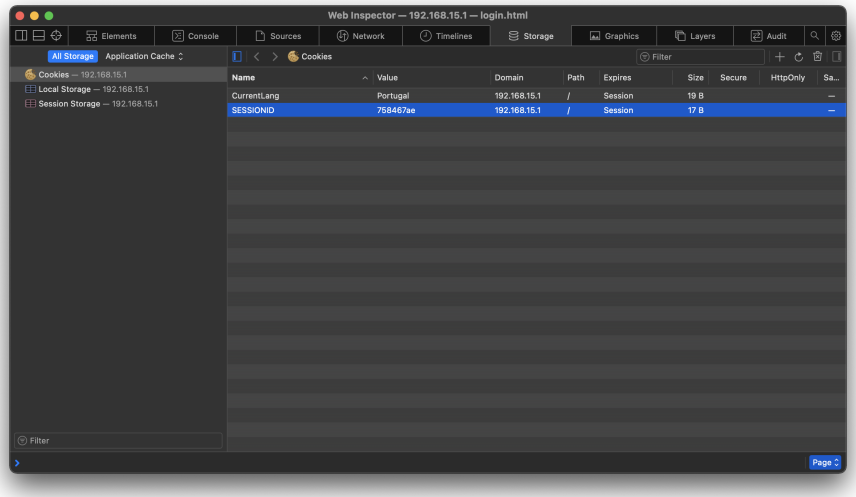


Figure 29: Cookies of the CWMP HTTP Management Interface

Strangely, no cookie, authorization token, or any other method of keeping the user's session was found on the captured requests of CPEs 0 and 1. After performing some tests, it was identified that the CPE authenticates the client based on its IP. If another client on the network is able to hijack the IP address of an authenticated user, it may gain access over the CPE's management interface.

6.2 Unauthenticated Routes

On all CPEs, at least two pages don't require authentication for their contents to be viewed, the status and about pages. No other pages that can be accessed through the menu seem to work without proper authentication, but it is known that some pages of the web interface are not indexed and can only be accessed directly with their URL.

It is possible to make some hidden pages visible on CPEs 0 and 1 by exporting the configuration file as previously mentioned, changing `PADRAO_WIFI_ONLY` from `0x0` to `0x1`, and reimporting the file. But to list all pages available on the HTTP Management Interface with certainty, the firmware image of each CPE was expanded and inspected.

To perform this operation, the binwalk tool was used. It takes a binary file and looks for different magic numbers of different file types, detecting which each segment of the file is. Then it is able to extract the individual segments and expand them.

By feeding the firmware images to binwalk, it was verified that CPE 5 also carries the management interface of subsidiaries of the ISP in other countries. A configuration flag indicates which firmware should be used by the CPE upon start. As the other management are not accessible without tampering with the device, their files were not inspected.

When testing the different URLs by using the file names, surprisingly, there was a serious finding on CPE 5. The page responsible for importing and exporting configuration files doesn't require authentication, as shown on Figure 30. Meaning that anyone that is able to send requests to the port 80/tcp is able to change any setting on the device, including the password of the management console, making null and void any kind of security on the management interfaces of the device.

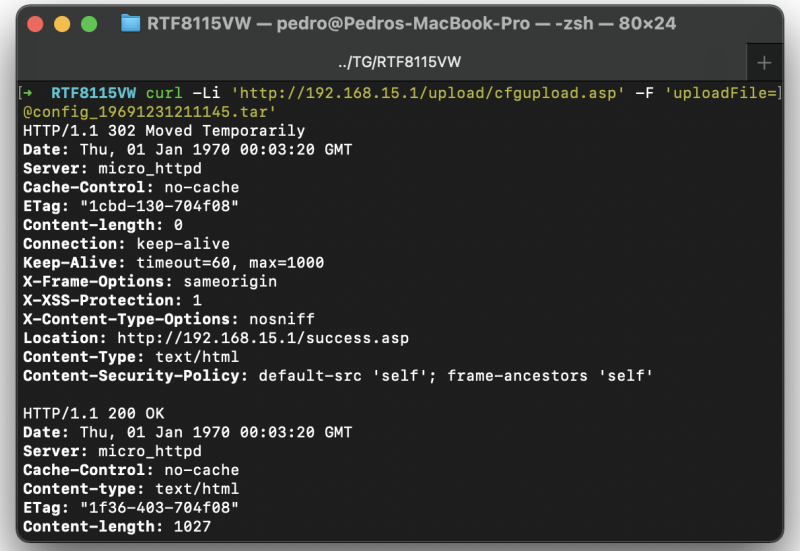


Figure 30: Unauthenticated CWMP Configuration Import

6.3 Symbolic Links

When exporting the configuration file, it was observed that CPE 5 simply writes it in a path on the filesystem. Then the user can download it just by knowing its name, this is also true for any file under the specified directory.

Recently, it was discovered that some residential gateways have a vulnerability that allowed an attacker to access all files of the device's file system by plugging a flash drive with a symbolic link to root on it [28]. This allowed the configuration files and password hashes to be recovered by accessing its sharing.

Similarly, a symbolic link to root was placed in the same folder as the configuration exports are stored. The result was that the HTTP server follows the symbolic link and allows every file on the equipment to be downloaded, as shown on Figure 31.

Although no way of injecting the malicious symbolic link on the device without requiring proper credentials was found, another vulnerability can be discovered in the future that would make this possible. Allowing an attacker to exploit it and gain access to all files on the device, including plaintext credentials.

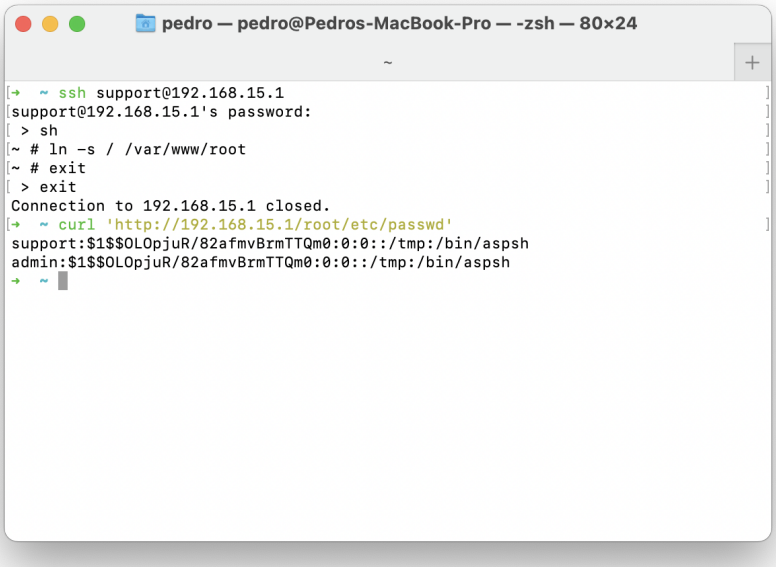


Figure 31: CPE HTTP Management Interface Following Symbolic Link

7 ISP-side Services Analysis

The safety of the service provided by the ISP not solely relies on the CPE used and how it is configured. Some services need to be protected on the ISP-side and the customer has to believe that the provider does it properly.

In this chapter, the services that the customer has to blindly rely on are inspected the their security assessed.

7.1 ACS

When emulating CPE requests to the ACS, it was discovered that the server is an instance of the Nokia Alcatel-Lucent Motive Home Device Manager (HDM) [18]. Ill-formed requests sent to the ACS resulted in an exception being returned to the user, as shown in Figure 32. The exception contained a Java package that could be traced back to the HDM software. It was further confirmed by the favicon used when accessing the ACS URL via a browser, `http://acs.telesp.net.br:7005/favicon.ico`.

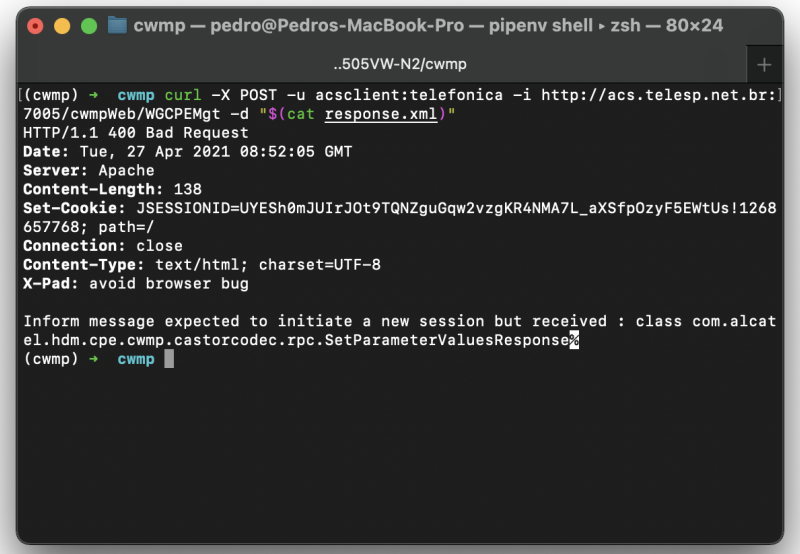


Figure 32: Exception Returned by the ACS

No information regarding the HDM version was able to be extracted. But it was found that the HDM is not exposed directly to the Internet and there is an Apache server, working as a reverse proxy, in front of it.

7.2 SIP

As previously noted, the SIP traffic is not protected by TLS. The lack of encryption means that an attacker could intercept all calls made and received by the customer and also capture the hashed SIP password to brute-force it.

The problem is aggravated by the discovery that any customer is able to authenticate to the SIP server with any phone number as long it has the correct credentials. The server doesn't check the origin of the request to forbid other customers from having access to numbers that don't belong to them. In the experiment, shown in Figure 33, a customer that isn't subscribed to the ISP VoIP service and has a copper-based connection, was able to authenticate with a phone number belonging to another customer that has a fiber-based connection and lives 120km apart from where the connection was initiated.

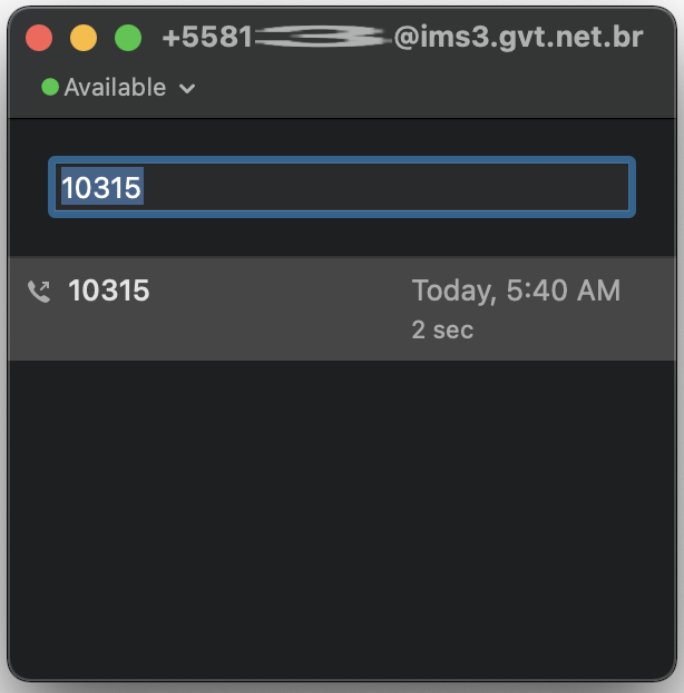


Figure 33: Call Made Using the Phone Number of Another Customer

It was verified that the SIP server has another two problems. It allows enumeration of phone numbers that it is able to authenticate, and there is no mechanism in place that prevents online brute-forcing. Using the PJSUA library to open a single connection, it was able to enumerate phones at a speed of 5.67 per second and brute-force passwords at a speed of 1.27 per second. Both attacks can be parallelized as long as the SIP server is able to handle the throughput of requests.

Enumeration was also found to be possible by trying to register a random number without providing any authorization token. If the phone number is valid, the client receives a 100 Trying response followed by an 401 Unauthorized response, Figure 34. On the other hand, if the phone doesn't exist on their database, the response is a 100 Trying followed by a 403 Forbidden, Figure 35.

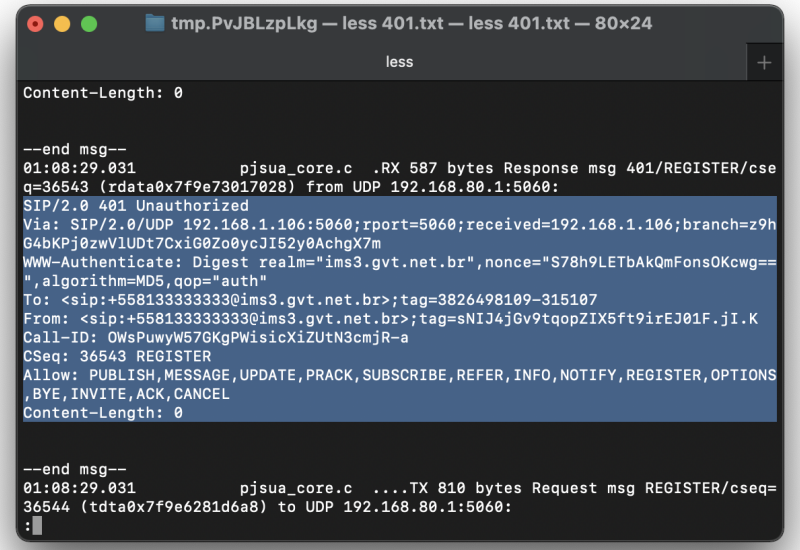


Figure 34: Unauthorized SIP Response for Existent Phone Number

In regards to online brute-forcing, an attacker is able to guess the password an unlimited number of tries without triggering any protection mechanism on the server. The attack can be distributed and run concurrently without any penalty.

Another critical mechanism that could be exploited by an attacker is the registration process of SIP credentials on the CPE. Unfortunately, none of the experiments performed were able to detect this procedure. The devices don't requests the credentials to the ISP and the ISP doesn't send the credentials

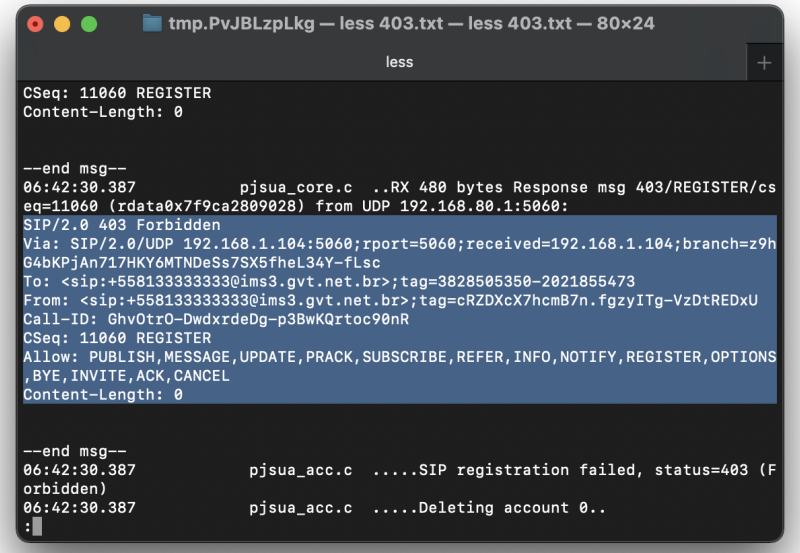


Figure 35: Forbidden SIP Response for Nonexistent Phone Number

proactively.

It is speculated that either the CPE is installed on the customer's home with the credentials already set, or that the technician is able to trigger some process that makes the ISP configure the credentials on CPE remotely. It also unknown if customer support is able to reconfigure the credentials, what would allow an attacker to impersonate the customer and induce the support to reset the password while the attacker is sniffing the physical transmission medium.

8 Substituting the ISP CPE

The chapter evaluates the possibility of replacing the ISP CPE with another device, allowing the customer not to rely on the ISP properly configuring the CPEs and instead using a CRG that allows full access to its settings.

The substitution will be considered successful if the acquired services work as expected without requiring the ISP CPE anymore.

8.1 Devices

To cover both ISP offerings, copper-based and fiber-based devices were used on the test, as presented on Table 12. The devices are manufactured by TP-Link and are designed for residential use. They have Registered Jack 11 (RJ-11) ports that allow analog telephones to be connected to the VoIP service of the ISP.

CRG Identifier	Transmission Medium	Manufacturer Name	Model Number
CRG-0	Copper	TP-Link	`Archer VR600v`
CRG-1	Fiber	TP-Link	`Archer XR500v`

Table 12: Identifiers of the CRGs

The configuration applied on the devices is based on the data collected from the ISP CPEs. Further analysis may be necessary to determine the proper setup for the new equipment.

8.2 Internet

In both cases, a PPPoE connection must be established with the proper credentials. The VLAN Identifier (ID) must be set to 600 and IPv6 can be enabled with Stateless Address Autoconfiguration (SLAAC).

When setting up the devices for the first time, or after a factory reset, a Quick Setup page, shown in Figure 36, is presented when the HTTP Management Interface is accessed.

In the case of CRG-0, the setup page contains a list of ISPs that the device can be automatically configured to work with. Although the ISP being analyzed shows on the list, the autoconfiguration doesn't work as expected and the residential gateway is not able to connect to the Internet. But also on the list, is an ISP that was acquired and incorporated by the researched provider and selecting it for autoconfiguration makes the equipment connect to the network.

CRG-1 doesn't have a predefined list of ISPs like CRG-0, but it can be easily configured with the required information. This process can also be performed on CRG-0 with almost no difference.

In the advanced section of the device's web interface, there is a page that allows Internet Connections to be manually configured. By clicking on the add button, the VLAN ID must be set to 600 and the Connection Type to PPPoE,

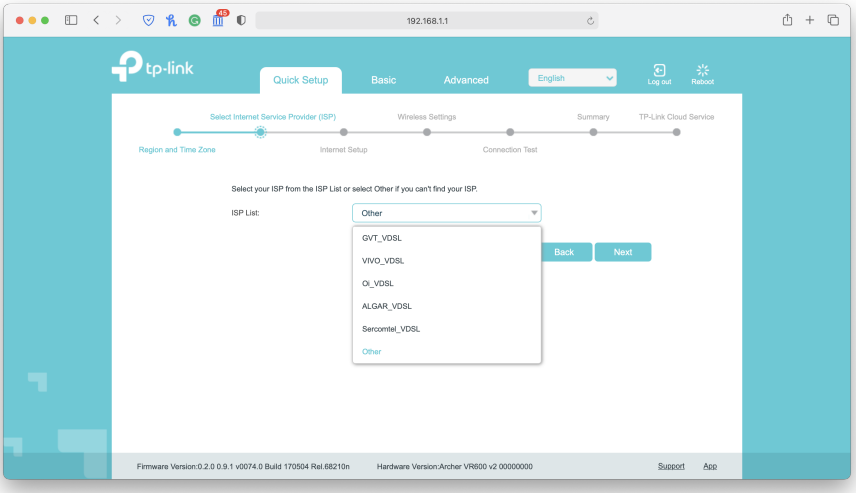


Figure 36: Quick Setup of the CRGs

as shown in Figure 37. The credentials for the PPPoE connection are the same found hard-coded on all CPEs inspected.

In fiber-based devices, you are also required to enter the GPON S/N, that could be found on the about page on the HTTP Management Interface of all fiber-based CPEs analyzed. While in copper-based devices, the Digital Subscriber Line (DSL) Modulation Type must be set to Very High-Speed DSL (VDSL).

Optionally, IPv6 can be enabled with SLAAC, but it depends on regional availability to work. The experiment showed that IPv6 works on devices located in the Metropolitan Area of the state, but it does not on the Agreste Region.

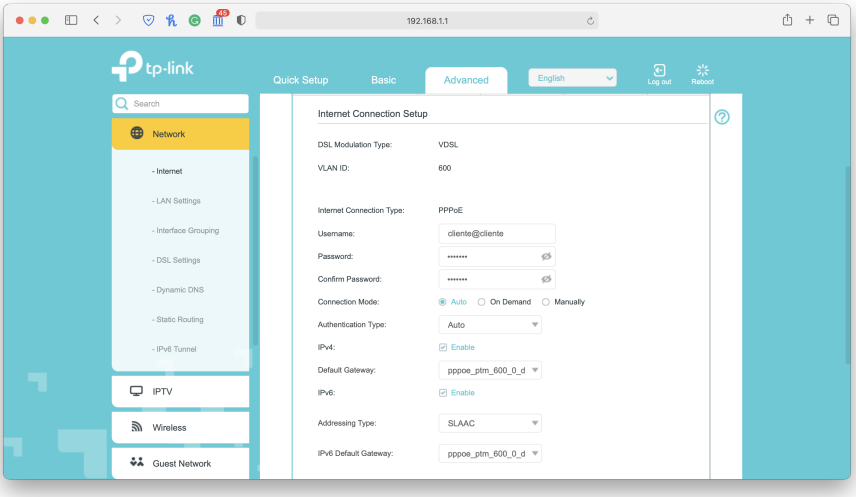


Figure 37: VLAN 600 Settings of the CRGs

8.3 VoIP

To configure the VoIP service on the CRGs, a similar process to the manual Internet setup is followed on both equipment. There is no difference in this process on copper-based and fiber-based devices.

The SIP configuration collected from all CPEs show that an outbound proxy is always used. This proxy is located at `192.168.80.1` and is only accessible on a network with VLAN ID different from the one used for the Internet Connection.

Then, the Internet Configuration page must be accessed and a new connection created. The VLAN ID must be set to 601 and the Connection Type to Dynamic IP, as shown in Figure 38. The default gateway must not be changed and IPv6 must not be enabled.

After creating this new connection, an IP address is automatically acquired. The joined network is part of the `10.0.0.0/8` IPv4 range and doesn't provide access to the Internet.

As the default gateway is set to the Internet Connection previously configured, a static route must be configured to make connections to the SIP Proxy be routed to the new network. This can also be done on the advanced section of the management interface. The Network Destination and Subnet Mask must be respectively set to `192.168.80.1` and `255.255.255.255`. The gateway field must be filled with the IP address of the gateway assigned to the device on the network with VLAN ID 601, and the interface must point to the same network as well, as shown in Figure 39. The experiments show that even after 6 months and reboots, the gateway assigned never changes, allowing the static route not be reconfigured when a new connection is established.

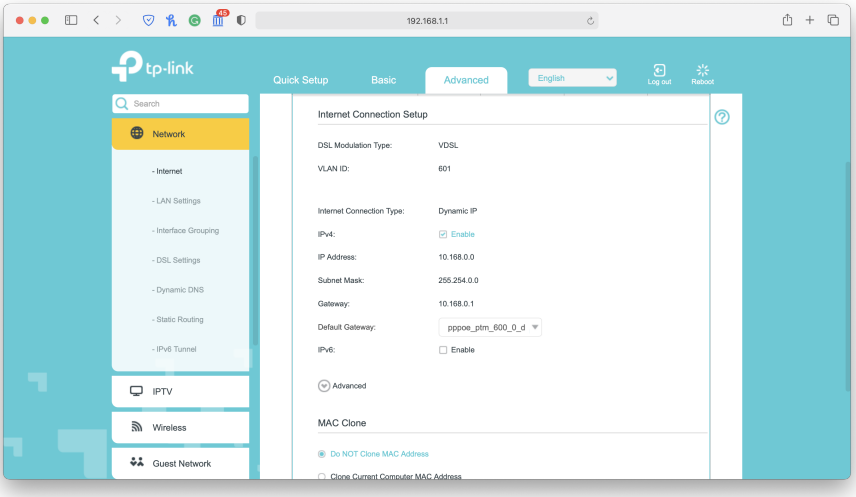


Figure 38: VLAN 601 Settings of the CRGs

With the VoIP Connection configured, a SIP client can be configured on the CRG itself, allowing analog telephones to work when connected to the RJ-11 ports. On the basic section of the HTTP Management Interface, telephone numbers can be registered. The Phone Number and Authentication ID must be set to the customer's phone number formatted following the E.164 standard. The Registrar Address and Password are set to the values acquired from the ISP CPE originally used by the customer. Finally, the Outbound Proxy is set to `192.168.80.1`, keeping the Outbound Proxy Port value of 5060 and the Register via Outbound Proxy checkbox enabled. The settings are shown in Figure 40.

After the phone number was properly registered, the connection status showed a blue check. An analog phone was connected to the RJ-11 port and a number was dialed. Unfortunately the connection was unreliable, sometimes the call was successful, but most of the time there was a problem. The dialer could listen but could not be heard, the opposite also happened. After hanging a call, the number could not receive or make calls for some minutes. The phone rang indefinitely, even when out of the hook.

To diagnose the problem, a SIP client was used on a computer instead of using the CRG one. This should work fine, because differently from the ISP CPEs, the CRG is not configured to block connections going to the SIP proxy. The Telephone app was used as SIP client, as it was free and open source. But after setting the connection on it, similar problems arose.

As it was unlikely that both SIP clients were not properly implemented, the SIP traffic of the CRG the the client running on the computer were captured and compared with the traffic previously captured from the SIP CPE. Everything looked very similar but with one subtle difference, the SIP clients from the CRG

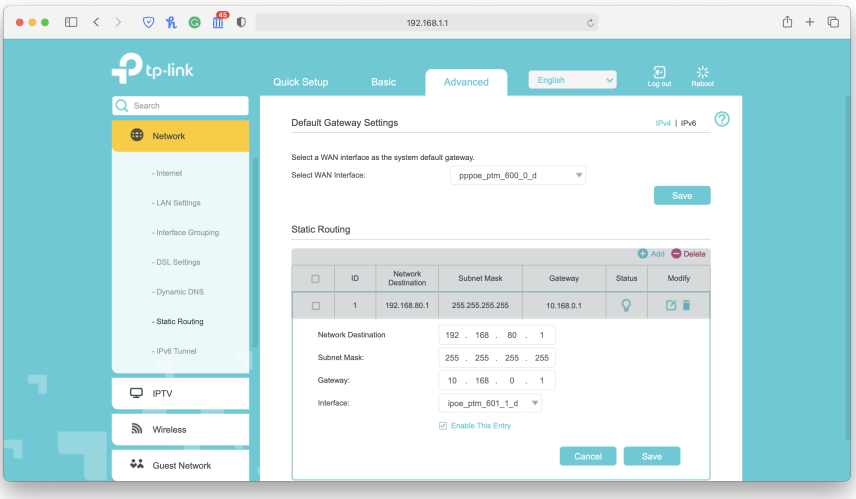


Figure 39: Static Routing Settings of the CRGs

and the computer were sending keep-alive requests to the SIP server, while the CPE wasn't sending them.

Looking at the implementation of the computer and CRG clients, it was found that they use the PJSIP to establish and manage the SIP connection. The CRG used the PJSIP Command Line Interface (CLI), while the computer client used the PJSIP library. Both are configured by default to send Carriage Return (CR)-Line Feed (LF) as keep-alive packets every 15 seconds.

The Telephone app is recompiled with a configuration that prevents those keep-alive requests from being sent. After doing that, the SIP client started to work perfectly, just like the SIP CPE, confirming the problem was indeed the keep-alive packets.

Unfortunately the CRGs have no configuration that allows keep-alive to be disabled. Additionally, the PJSUA CLI doesn't have this flag either, so the PJSUA library should be used instead to allow this configuration to be changed. The firmware of the devices are closed-source, so it is not easy to build a new firmware with the changes required.

While looking for alternatives, it was found that both CRGs have iptables installed on them and have the necessary modules installed to filter the keep-alive packages going to the SIP proxy. The devices were restarted without the copper or fiber cables connected to them, preventing any connection from being established. Then the SSH Management Interface was accessed and iptables rule was created, dropping the SIP keep-alive packets. The copper and fiber cables were reconnected and the connection established. From that point, although the SIP client was still sending the problematic packets, the SIP server never received them and the service started working just like the modified Telephone

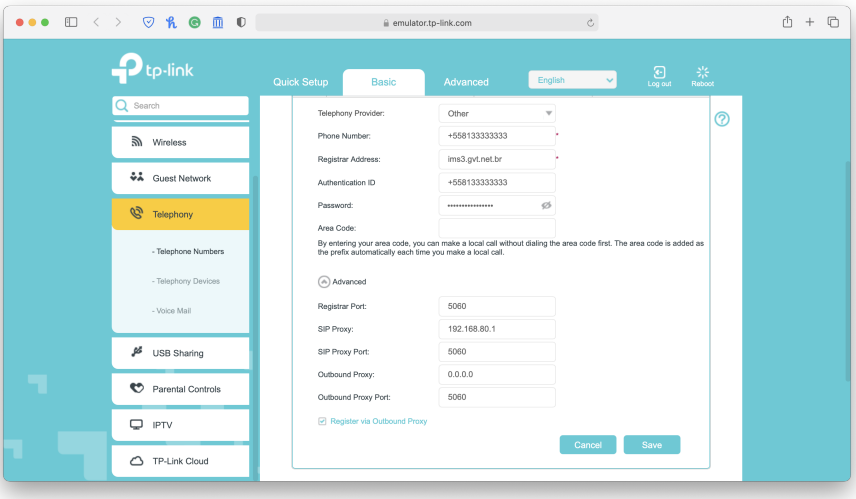


Figure 40: Telephone Numbers Settings of the CRGs

app.

```
1  iptables -I OUTPUT -p udp -j DROP -d 192.168.80.1 -m
       udp --dport 5060 -m string --hex-string '|0d0a|'
       --algo bm --from 28 --to 30
```

The solution found is untethered and the processed needs to be executed every time the CRG reboots. But it is feasible to use if the device is not rebooted frequently, or if a computer that is always connected to it via a wired connection is able to automatically execute the process every time the connection is lost and then reestablished. Another possibility would be having the SIP proxy point to a device that drops the keep-alive packets and then forward to the real SIP proxy.

It is important to notice that the CRG doesn't need to support VoIP if a SIP client running on another device can be used in the desired scenario, it just needs to be able to establish a connection with the proper VLAN ID and Connection Type. This also means that consumers can acquire cheaper CRGs that don't have VoIP built-in and use other solutions that fit in the use-case, like an external Analog Telephone Adapter (ATA).

8.4 IPTV

As previously mentioned, the test environments didn't have an IPTV subscription of the ISP, but information the data collected indicates how it could be properly configured. Although problems like the SIP keep-alive packets wouldn't be detected.

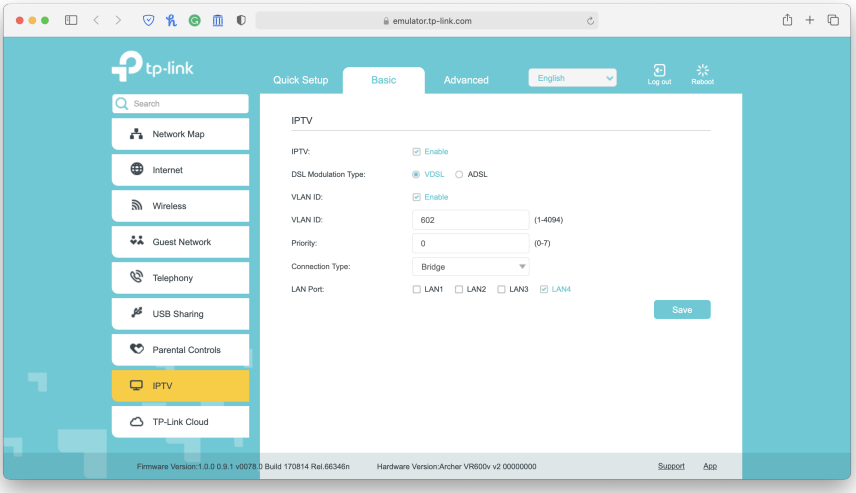


Figure 41: IPTV Settings of the CRGs

The IPTV traffic is routed through a third network. This can be configured on the basic section of the HTTP Management Interface, as shown on Figure 41. IPTV is enabled, the VLAN ID is set 602, and the Connection Type kept as Bridge. The set-top box provided by the ISP is connected to a LAN port of the residential gateway through an Ethernet cable. That specific LAN port must be checked on the configuration page of IPTV, allowing the Set-Top Box (STB) to access the correct VLAN.

This should be enough to have the IPTV service properly working, unfortunately it wasn't verifiable and additional tweaking may be required.

9 Conclusion

The certification programs developed by the Wi-Fi Alliance have successfully ensured interoperability between devices and have set a minimal level of security on the certified devices. But when flaws were discovered on some authentication or encryption mechanism, the interoperability and security goals conflicted and the solutions proposed were controversial. The superseding standards tried to maintain a compatibility layer with the previous protocol that showed to cause security problems.

All the assessed CPEs fulfill their job as a plug'n'play solution that provides access to all services acquired from the ISP without requiring customer intervention. But the different pieces of equipment provide their frictionless experience with different levels of security. Some come configured with settings that increase interoperability while others focus on security.

Although there was no evidence that the CPEs proactively forward information about the local network and devices connected to it, anyone who accesses their HTTP Management Interface is able to gather the information without authenticating, including the customer phone number and the device's GPON S/N. The situation is aggravated in CPEs 2 and 3, as they allow the management interfaces to be accessed from WAN.

Problems like the use of old security standard and weak passwords could be easily mitigated by changing the default configurations set by the ISP. But sometimes it is not as easy as it should be, some settings are purposely hidden from the consumer management interface and are only accessible when signing in with the support user in an unindexed page. Even worse, on CPEs 0 and 1, the support management interface is incomplete and some settings can only be changed by exporting the configuration file, modifying its content, and finally importing the file back to the device. On CPE 5, the configuration file also needs to be decrypted after download and encrypted before upload.

But flaws in the implementation of management interfaces of some devices pose a serious threat to the security and privacy of consumers, these can only be patched with a firmware upgrade. CPEs 0 and 1 don't rely on cookies and/or tokens to maintain the user session on the HTTP Management Interface, instead, any request reaching the interface from an specific IP is considered authenticated after a user logs in. Even worse, these CPEs don't support firmware upgrades, so the problems are unfixable. But the most concerning problem is the unauthenticated configuration import of CPE 5, making null and void almost any kind of security measure the device has in place, as any configuration can be overwritten by anyone able to send requests to its HTTP Management Interface.

The act of remote managing of CPEs by the ISP is not a problem by itself. But the connection with the ISP ACS is not encrypted and may be tampered with. Connections to the file server that hosts the firmware images for the ISP CPEs are also not encrypted and pose a similar threat. Unfortunately there is no way to reconfigure CWMP in a more secure fashion, as the ISP unreasonably downgrades the connection of CPEs using HTTPs to HTTP. The only approach

to the problem would be to disable CWMP completely.

Out of the customer control, the SIP server of the ISP should use TLS even if the traffic doesn't go through the broadband network; the password should be more complex and be completely random, not following any pattern; the origin of registration requests should be verified and only the customer that owns the telephone number should be able to authenticate with it; and mechanisms to decrease the feasibility of online brute-force attacks should be placed.

Using a CRG as replacement for CPEs proved to be possible and Internet configuration is straightforward. Although the VoIP configuration may be troublesome, it is feasible to make it work reliably. It is important to notice that while the CRG allows customers to freely configure the device and not have to rely on the ISP anymore, it also makes the consumer responsible for properly maintaining the device. If a CRG is not properly configured and regularly updated, it might not be better than an out-of-the-box CPE.

Regardless of the device used as a residential gateway, there are some recommendations that the customer should follow to improve the security of its residencial network. Changing the default passwords of the management interfaces and wireless networks is a must do, strong passwords should be used. The use of WPS is disencourage and should be disabled. If the wireless devices connected to the network support newer security protocols, interoperability with old standards should be disabled on the residential gateway. Management interfaces must not be accessible via WAN and, if not used, non-HTTP interfaces should be completely disabled. Finally, it is important to keep all devices on the network up to date, not only the residential gateway.

A CWMP Device Emulation

A.1 main.py

```
1  import requests
2  import time
3  import xml.etree.ElementTree
4
5  def main():
6      url =
           'http://acs.telesp.net.br:7005/cwmpWeb/WGCPEMgt'
7
8      s = requests.Session()
9      s.auth = ('acsclient', 'telefonica')
10
11     with open('inform.xml', 'rb') as f:
12         s.post(url, data=f.read())
13
14     time.sleep(1)
15
16     r = s.post(url)
17     print(r.text)
18
19     with open('response.xml', 'rb') as f:
20         s.post(url, data=f.read())
21
22     with open('get.xml', 'rb') as f:
23         s.post(url, data=f.read())
24
25     with open('inform2.xml', 'rb') as f:
26         s.post(url, data=f.read())
27
28     time.sleep(1)
29
30     r = s.post(url)
31     print(r.text)
32
33     with open('response.xml', 'rb') as f:
34         s.post(url, data=f.read())
35
36     spv = xml.etree.ElementTree.fromstring(r.content)
37     for pvs in
          spv.findall("./{http://schemas.xmlsoap.org/soap/envelope/}Body/{urn:d
38         if pvs.find('Name').text ==
               'InternetGatewayDevice.ManagementServer.URL':
39             url = pvs.find('Value').text
```

```
40          elif pvs.find('Name').text ==
               'InternetGatewayDevice.ManagementServer.Username':
41              username = pvs.find('Value').text
42          elif pvs.find('Name').text ==
               'InternetGatewayDevice.ManagementServer.Password':
43              password = pvs.find('Value').text
44
45      s = requests.Session()
46      s.auth = (username, password)
47
48      with open('inform3.xml', 'rb') as f:
49          s.post(url, data=f.read())
50
51      time.sleep(4)
52
53      r = s.post(url)
54      print(r.text)
55
56      with open('response.xml', 'rb') as f:
57          s.post(url, data=f.read())
58
59  if __name__ == '__main__':
60      main()
```

A.2 get.xml

```
1  <SOAP-ENV:Envelope
2    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
3    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
4    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6    xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
7  <SOAP-ENV:Header>
8  <cwmp:ID SOAP-ENV:mustUnderstand="1">null1</cwmp:ID>
9  </SOAP-ENV:Header>
10 <SOAP-ENV:Body>
11 <cwmp:GetParameterValuesResponse>
12 <ParameterList xsi:type="SOAP-ENC:Array"
      SOAP-ENC:arrayType="cwmp:ParameterValueStruct[1]">
13 <ParameterValueStruct>
14 <Name>InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANPPPConnecti
15 <Value xsi:type="xsd:string">10:72:23:0D:83:CC</Value>
16 </ParameterValueStruct>
17 </ParameterList>
18 </cwmp:GetParameterValuesResponse>
19 </SOAP-ENV:Body>
```

```
20 </SOAP-ENV:Envelope>
```

A.3 inform.xml

```
 1 <SOAP-ENV:Envelope
 2 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 3 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 4 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 5 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 6 xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
 7 <SOAP-ENV:Header>
 8 <cwmp:ID
      SOAP-ENV:mustUnderstand="1">1681692777</cwmp:ID>
 9 </SOAP-ENV:Header>
10 <SOAP-ENV:Body>
11 <cwmp:Inform>
12 <DeviceId>
13 <Manufacturer>Askey</Manufacturer>
14 <OUI>009096</OUI>
15 <ProductClass>RTF3505VW-N2</ProductClass>
16 <SerialNumber>1072230D83CC</SerialNumber>
17 </DeviceId>
18 <Event SOAP-ENC:arrayType="cwmp:EventStruct[2]">
19 <EventStruct>
20 <EventCode>0 BOOTSTRAP</EventCode>
21 <CommandKey></CommandKey>
22 </EventStruct>
23 <EventStruct>
24 <EventCode>1 BOOT</EventCode>
25 <CommandKey></CommandKey>
26 </EventStruct>
27 </Event>
28 <MaxEnvelopes>1</MaxEnvelopes>
29 <CurrentTime>2021-04-13T20:57:35</CurrentTime>
30 <RetryCount>0</RetryCount>
31 <ParameterList
      SOAP-ENC:arrayType="cwmp:ParameterValueStruct[7]">
32 <!-- <ParameterValueStruct>
33 <Name>InternetGatewayDevice.DeviceSummary</Name>
34 <Value
      xsi:type="xsd:string">InternetGatewayDevice:1.4[](Baseline:2,
      EthernetLAN:2, Time:2, IPPing:1,
      DeviceAssociation:2, QoS:2, WiFiLAN:2,Download:1,
      Upload:1, DownloadTCP:1, UploadTCP:1, UDPEcho:1,
      UDPEchoPlus:1, CaptivePortal:1,Bridging:2,
      EthernetWAN:1, WiFiWMM:1, WiFiWPS:1,
```

```
      DHCPCondServing:1,
      DHCPOption:1,X_TELEFONICA-ES_IGMP:1,
      X_TELEFONICA-ES_NATType:1,X_TELEFONICA-ES_Firewall:1,
      X_TELEFONICA-ES_SystemState:1),
      VoiceService:1.0[1](Endpoint:1,
      SIPEndpoint:1)</Value>
35 </ParameterValueStruct> -->
36 <ParameterValueStruct>
37 <Name>InternetGatewayDevice.DeviceInfo.SpecVersion</Name>
38 <Value xsi:type="xsd:string">1.0</Value>
39 </ParameterValueStruct>
40 <ParameterValueStruct>
41 <Name>InternetGatewayDevice.DeviceInfo.HardwareVersion</Name>
42 <Value xsi:type="xsd:string">REV4_B4</Value>
43 </ParameterValueStruct>
44 <ParameterValueStruct>
45 <Name>InternetGatewayDevice.DeviceInfo.SoftwareVersion</Name>
46 <Value
      xsi:type="xsd:string">BR_SG_s00.00_g000_3505019</Value>
47 </ParameterValueStruct>
48 <ParameterValueStruct>
49 <Name>InternetGatewayDevice.DeviceInfo.ProvisioningCode</Name>
50 <Value xsi:type="xsd:string">PROV</Value>
51 </ParameterValueStruct>
52 <ParameterValueStruct>
53 <Name>InternetGatewayDevice.ManagementServer.ConnectionRequestURL</Name>
54 <Value
      xsi:type="xsd:string">http://127.0.0.1:7547/</Value>
55 </ParameterValueStruct>
56 <ParameterValueStruct>
57 <Name>InternetGatewayDevice.ManagementServer.ParameterKey</Name>
58 <Value xsi:type="xsd:string"></Value>
59 </ParameterValueStruct>
60 <ParameterValueStruct>
61 <Name>InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANPPPConnecti
62 <Value xsi:type="xsd:string">127.0.0.1</Value>
63 </ParameterValueStruct>
64 </ParameterList>
65 </cwmp:Inform>
66 </SOAP-ENV:Body>
67 </SOAP-ENV:Envelope>
```

A.4 inform2.xml

```
1 <SOAP-ENV:Envelope
2 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

```
3  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
4  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6  xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
7  <SOAP-ENV:Header>
8  <cwmp:ID
      SOAP-ENV:mustUnderstand="1">1681692777</cwmp:ID>
9  </SOAP-ENV:Header>
10 <SOAP-ENV:Body>
11 <cwmp:Inform>
12 <DeviceId>
13 <Manufacturer>Askey</Manufacturer>
14 <OUI>009096</OUI>
15 <ProductClass>RTF3505VW-N2</ProductClass>
16 <SerialNumber>1072230D83CC</SerialNumber>
17 </DeviceId>
18 <Event SOAP-ENC:arrayType="cwmp:EventStruct[1]">
19 <EventStruct>
20 <EventCode>6 CONNECTION REQUEST</EventCode>
21 <CommandKey></CommandKey>
22 </EventStruct>
23 </Event>
24 <MaxEnvelopes>1</MaxEnvelopes>
25 <CurrentTime>2021-04-13T20:57:35</CurrentTime>
26 <RetryCount>0</RetryCount>
27 <ParameterList
      SOAP-ENC:arrayType="cwmp:ParameterValueStruct[7]">
28 <ParameterValueStruct>
29 <Name>InternetGatewayDevice.DeviceInfo.SpecVersion</Name>
30 <Value xsi:type="xsd:string">1.0</Value>
31 </ParameterValueStruct>
32 <ParameterValueStruct>
33 <Name>InternetGatewayDevice.DeviceInfo.HardwareVersion</Name>
34 <Value xsi:type="xsd:string">REV4_B4</Value>
35 </ParameterValueStruct>
36 <ParameterValueStruct>
37 <Name>InternetGatewayDevice.DeviceInfo.SoftwareVersion</Name>
38 <Value
      xsi:type="xsd:string">BR_SG_s00.00_g000_3505019</Value>
39 </ParameterValueStruct>
40 <ParameterValueStruct>
41 <Name>InternetGatewayDevice.DeviceInfo.ProvisioningCode</Name>
42 <Value xsi:type="xsd:string">PROV</Value>
43 </ParameterValueStruct>
44 <ParameterValueStruct>
45 <Name>InternetGatewayDevice.ManagementServer.ConnectionRequestURL</Name>
```

```
<Value
   xsi:type="xsd:string">http://127.0.0.1:7547/</Value>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>InternetGatewayDevice.ManagementServer.ParameterKey</Name>
<Value xsi:type="xsd:string"></Value>
</ParameterValueStruct>
<ParameterValueStruct>
<Name>InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANPPPConnecti
<Value xsi:type="xsd:string">127.0.0.1</Value>
</ParameterValueStruct>
</ParameterList>
</cwmp:Inform>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A.5 inform3.xml

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
<SOAP-ENV:Header>
<cwmp:ID
   SOAP-ENV:mustUnderstand="1">1681692777</cwmp:ID>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<cwmp:Inform>
<DeviceId>
<Manufacturer>Askey</Manufacturer>
<OUI>009096</OUI>
<ProductClass>RTF3505VW-N2</ProductClass>
<SerialNumber>1072230D83CC</SerialNumber>
</DeviceId>
<Event SOAP-ENC:arrayType="cwmp:EventStruct[1]">
<EventStruct>
<EventCode>0 BOOTSTRAP</EventCode>
<CommandKey></CommandKey>
</EventStruct>
</Event>
<MaxEnvelopes>1</MaxEnvelopes>
<CurrentTime>2021-04-13T20:57:35</CurrentTime>
<RetryCount>0</RetryCount>
<ParameterList
```

```
      SOAP-ENC:arrayType="cwmp:ParameterValueStruct[7]">
28 <ParameterValueStruct>
29 <Name>InternetGatewayDevice.DeviceInfo.SpecVersion</Name>
30 <Value xsi:type="xsd:string">1.0</Value>
31 </ParameterValueStruct>
32 <ParameterValueStruct>
33 <Name>InternetGatewayDevice.DeviceInfo.HardwareVersion</Name>
34 <Value xsi:type="xsd:string">REV4_B4</Value>
35 </ParameterValueStruct>
36 <ParameterValueStruct>
37 <Name>InternetGatewayDevice.DeviceInfo.SoftwareVersion</Name>
38 <Value
      xsi:type="xsd:string">BR_SG_s00.00_g000_3505019</Value>
39 </ParameterValueStruct>
40 <ParameterValueStruct>
41 <Name>InternetGatewayDevice.DeviceInfo.ProvisioningCode</Name>
42 <Value xsi:type="xsd:string">PROV</Value>
43 </ParameterValueStruct>
44 <ParameterValueStruct>
45 <Name>InternetGatewayDevice.ManagementServer.ConnectionRequestURL</Name>
46 <Value
      xsi:type="xsd:string">http://127.0.0.1:7547/</Value>
47 </ParameterValueStruct>
48 <ParameterValueStruct>
49 <Name>InternetGatewayDevice.ManagementServer.ParameterKey</Name>
50 <Value xsi:type="xsd:string"></Value>
51 </ParameterValueStruct>
52 <ParameterValueStruct>
53 <Name>InternetGatewayDevice.WANDevice.1.WANConnectionDevice.1.WANPPPConnecti
54 <Value xsi:type="xsd:string">127.0.0.1</Value>
55 </ParameterValueStruct>
56 </ParameterList>
57 </cwmp:Inform>
58 </SOAP-ENV:Body>
59 </SOAP-ENV:Envelope>
```

B CPE EPROM Extraction

B.1 dump.exp

```
#!/usr/bin/env expect
set timeout -1
log_file log.txt

spawn telnet 192.168.15.1
expect "Username:"
send "support\r"
expect "Password:"
send "pzxTPx6F\r"
expect "$$"
send "sh\r"
expect "XDSL>"
send "show system flashlayout\r"
expect "XDSL>"
send "memory show 0xb4000000 length 0x800000\r"
expect "XDSL>"
send "exit\r"
```

B.2 convert.py

```
def main():
    with open('hex.txt', 'r') as r:
        with open('bin', 'wb') as wb:
            line = r.readline()
            while line:
                wb.write(bytearray(map(lambda e:
                   int(e, 16), line[12:59].split('␣
                   '))))
                line = r.readline()

if __name__ == '__main__':
    main()
```

C CPE Configuration File Decryption

C.1 main.py

```
1  import sys
2
3  # libaspcm.so
4  def sub_47a8(buffer):
5      for i in range(len(buffer)):
6          buffer[i] ^= 8
7
8  def main():
9      b = bytearray(sys.stdin.read()[16:])
10     sub_47a8(b)
11     sys.stdout.write(b)
12
13 if __name__ == '__main__':
14     main()
```

D SIP Password Iterator

D.1 main.c

```
1  #include <stdio.h>
2
3  #define ABC (('9' - '0' + 1) + ('Z' - 'A' + 1) + ('z'
       - 'a' + 1))
4  #define LEN (ABC + 1 + 1)
5
6  int fwd[LEN];
7  int bwd[128];
8
9  void init_idx() {
10     int i = 0;
11     for (char c = '0'; c <= '9'; ++c) {
12         fwd[i] = c;
13         bwd[c] = i;
14         ++i;
15     }
16     for (char c = 'A'; c <= 'Z'; ++c) {
17         fwd[i] = c;
18         bwd[c] = i;
19         ++i;
20     }
21     for (char c = 'a'; c <= 'z'; ++c) {
22         fwd[i] = c;
23         bwd[c] = i;
24         ++i;
25     }
26     fwd[i] = '\0';
27     bwd['\0'] = i;
28 }
29
30 char pwd[16 + 1];
31 void next_pwd() {
32     for (int i = 7; i >= 0; --i) {
33         pwd[i] = fwd[bwd[pwd[i]] + 1];
34         if (pwd[i] != '\0')
35             break;
36         pwd[i] = fwd[0];
37     }
38     for (int i = 7; i >= 0; --i) {
39         char c = pwd[i + 8] = pwd[i];
40         if (c >= 'A' && c <= 'Z')
41             pwd[i + 8] += 'a' - 'A';
```

```
42          if (c >= 'a' && c <= 'z')
43              pwd[i + 8] += 'A' - 'a';
44      }
45  }
46
47  int main() {
48      init_idx();
49      for (int i = 0; i < ABC * ABC * ABC * ABC; ++i) {
50          for (int j = 0; j < ABC * ABC * ABC * ABC;
                ++j) {
51              next_pwd();
52              printf("%s\n", pwd);
53          }
54      }
55      return 0;
56  }
```

References

[1] Agência Nacional de Telecomunicações. "Painéis de Dados - Acessos - Banda Larga Fixa". https://informacoes.anatel.gov.br/paineis/acessos/banda-larga-fixa.

[2] Amazon Web Services. "Amazon EC2 P3 Instances". https://aws.amazon.com/ec2/instance-types/p3/.

[3] A. Barth. "HTTP State Management Mechanism", Apr. 2011. https://rfc-editor.org/rfc/rfc6265.txt.

[4] D. Bongard. "Offline bruteforce attack on WiFi Protected Setup", Oct. 2014. http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf.

[5] Broadband Forum. "TR-069 CPE WAN Management Protocol", June 2020. https://www.broadband-forum.org/technical/download/TR-069.pdf.

[6] D. Carrel, J. Evarts, K. Lidl, L. A. Mamakos, D. Simone, and R. Wheeler. "A Method for Transmitting PPP Over Ethernet (PPPoE)", Feb. 1999. https://rfc-editor.org/rfc/rfc2516.txt.

[7] CLARO S.A. "CONTRATO DE PRESTAÇÃO DE SERVIÇO DE COMUNICAÇÃO MULTIMÍDIA (SCM)". https://www.claro.com.br/documento/2019/06/13/contrato-net-virtua-claro-s.a-sao-paulo-clausula-tcooperacao-1374092328048.pdf.

[8] S. Fluhrer, I. Mantin, and A. Shamir. "Weaknesses in the Key Scheduling Algorithm of RC4". In *"Selected Areas in Cryptography"*, pages 1–24. Springer Berlin Heidelberg, Dec. 2001. https://doi.org/10.1007/3-540-45537-X_1.

[9] D. Harkins. "Dragonfly Key Exchange", Nov. 2015. https://rfc-editor.org/rfc/rfc7664.txt.

[10] D. Harkins and W. A. Kumari. "Opportunistic Wireless Encryption", Mar. 2017. https://rfc-editor.org/rfc/rfc8110.txt.

[11] Insecure.Org. "Nmap: the Network Mapper - Free Security Scanner". https://nmap.org.

[12] A. Kuznetsov. "ping - send ICMP ECHO_REQUEST to network hosts". https://linux.die.net/man/8/ping.

[13] LAN/MAN Standards Committee of the IEEE Computer Society. "ABOUT IEEE P802.11 AND HOW TO PARTICIPATE". https://www.ieee802.org/11/abt80211.html.

[14] LAN/MAN Standards Committee of the IEEE Computer Society. "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, Feb. 2021. https://doi.org/10.1109/IEEESTD.2021.9363693.

[15] K. Moriarty, B. Kaliski, and A. Rusch. "PKCS #5: Password-Based Cryptography Specification Version 2.1", Jan. 2017. https://rfc-editor.org/rfc/rfc8018.txt.

[16] National Security Agency. "Ghidra". https://ghidra-sre.org.

[17] NETGEAR. "Security Advisory for Multiple Vulnerabilities on Some Routers, Mobile Routers, Modems, Gateways, and Extenders". https://kb.netgear.com/000061982.

[18] Nokia Corporation. "Motive Home Device Manager (HDM)". https://supportstage.alcatel-lucent.com/portal/web/support/product-result?entryId=1-0000000002312.

[19] J. Postel. "User Datagram Protocol", Aug. 1980. https://rfc-editor.org/rfc/rfc768.txt.

[20] ReFirm Labs. "Binwalk - Firmware Analysis Tool". https://github.com/ReFirmLabs/binwalk.

[21] E. Rescorla. "The Transport Layer Security (TLS) Protocol Version 1.3", Aug. 2018. https://rfc-editor.org/rfc/rfc8446.txt.

[22] R. L. Rivest. "The MD5 Message-Digest Algorithm", Apr. 1992. https://rfc-editor.org/rfc/rfc1321.txt.

[23] E. Schooler, J. Rosenberg, H. Schulzrinne, A. Johnston, G. Camarillo, J. Peterson, R. Sparks, and M. J. Handley. "SIP: Session Initiation Protocol", July 2002. https://rfc-editor.org/rfc/rfc3261.txt.

[24] D. K. R. Sollins. "The TFTP Protocol (Revision 2)", July 1992. https://rfc-editor.org/rfc/rfc1350.txt.

[25] TELEFÔNICA BRASIL S.A. "CONTRATO DE ADESÃO DE PRESTAÇÃO DO SERVIÇO TELEFÔNICO FIXO COMUTADO (STFC), DO SERVIÇO DE COMUNICAÇÃO MULTIMÍDIA (SCM) E DO SERVIÇO DE ACESSO CONDICIONADO (TV POR ASSINATURA - SeAC).". http://promovivofibra.clientes.ananke.com.br/planos-em-vigor/corpo_planos.php?codigo_item=ADESAO&uf=PE&cidade=TODOS&produto=internet_b2c.

[26] TELEFÔNICA BRASIL S.A. "Vivo Guru Tutoriais — Como Configurar Aparelhos e Aplicativos". https://configuraraparelhos.vivo.com.br/?pagina=device/internet-fixa.

[27] TELEMAR NORTE LESTE S.A. "CONTRATO DE ADESÃO À BANDA LARGA DA OI CATEGORIA RESIDENCIAL". https://www.oi.com.br/ArquivosEstaticos/oi/docs/pdf/oivelox_regs/contrato-de-adesao-banda-larga-da-oi-residencial-r1-b2c.pdf.

[28] Tenable Network Security, Inc. "CVE-2020-5795", Nov. 2020. https://nvd.nist.gov/vuln/detail/CVE-2020-5795.

[29] E. Tews, R.-P. Weinmann, and A. Pyshkin. "Breaking 104 Bit WEP in Less Than 60 Seconds". In *"Information Security Applications"*, pages 188–202. Springer Berlin Heidelberg, Dec. 2007. https://doi.org/10.1007/978-3-540-77535-5_14.

[30] The Tcpdump Group. "TCPDUMP/LIBPCAP public repository". https://www.tcpdump.org.

[31] TIM S/A. "CONTRATO DE PRESTAÇÃO DE SERVIÇOS". https://www.tim.com.br/Portal_Conteudo/_staticfiles/dpmFiles/pdf/Contrato%20de%20Presta%C3%A7%C3%A3o%20de%20Servi%C3%A7os%20Live%20TIM.pdf.

[32] M. Vanhoef and F. Piessens. "All Your Biases Belong to Us: Breaking RC4 in WPA-TKIP and TLS". In *"24th USENIX Security Symposium (USENIX Security 15)"*, pages 97–112. USENIX Association, Aug. 2015. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/vanhoef.

[33] M. Vanhoef and F. Piessens. "Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2". In *"Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security"*, pages 1313–1328. Association for Computing Machinery, 2017. https://doi.org/10.1145/3133956.3134027.

[34] M. Vanhoef and E. Ronen. "Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd". In *"2020 IEEE Symposium on Security and Privacy (SP)"*, pages 517–533. IEEE, May 2020. https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00031.

[35] Vector 35. "Binary Ninja". https://binary.ninja.

[36] S. Viehböck. "Brute forcing Wi-Fi Protected Setup", Dec. 2011. https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.

[37] P. Weidenbach and J. vom Dorp. "Home Router Security Report 2020", June 2020. https://www.fkie.fraunhofer.de/content/dam/fkie/de/documents/HomeRouter/HomeRouterSecurity_2020_Bericht.pdf.

[38] Wi-Fi Alliance. "Certification". https://www.wi-fi.org/certification.

[39] Wi-Fi Alliance. "The State of Wi-Fi Security". https://www.wi-fi.org/download.php?file=/sites/default/files/private/20120229_State_of_Wi-Fi_Security_09May2012_updated_cert.pdf.

[40] Wi-Fi Alliance. "Wi-Fi Alliance introduces Wi-Fi 6". https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-6.

[41] Wi-Fi Alliance. "Wi-Fi CERTIFIED Enhanced Open delivers data protection in open Wi-Fi networks". https://www.wi-fi.org/news-events/newsroom/wi-fi-certified-enhanced-open-delivers-data-protection-in-open-wi-fi-networks.

[42] Wi-Fi Alliance. "Wi-Fi Easy Connect Specification". https://www.wi-fi.org/download.php?file=/sites/default/files/private/Wi-Fi_Easy_Connect_Specification_v2.0.pdf.

[43] Wi-Fi Alliance. "Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks". https://web.archive.org/web/20051108045221/https://www.wi-fi.org/OpenSection/pdf/Whitepaper_Wi-Fi_Security4-29-03.pdf.

[44] Wi-Fi Alliance. "Wi-Fi Protected Setup Protocol and Usability Best Practices". https://www.wi-fi.org/download.php?file=/sites/default/files/private/Wi-Fi_Protected_Setup_Best_Practices_v2.0.2.pdf.

[45] Wi-Fi Alliance. "Removal of TKIP from Wi-Fi Devices", Mar. 2015. https://www.wi-fi.org/download.php?file=/sites/default/files/private/Wi-Fi_Alliance_Technical_Note_TKIP_v1.0.pdf.

[46] Wi-Fi Alliance. "WPA3 Security Considerations", Nov. 2019. https://www.wi-fi.org/download.php?file=/sites/default/files/private/WPA3_Security_Considerations_201911.pdf.

[47] Wi-Fi Alliance. "WPA3 Specification", Dec. 2020. https://www.wi-fi.org/download.php?file=/sites/default/files/private/WPA3_Specification_v3.0.pdf.

[48] Wireless Ethernet Compatibility Alliance. "Mission Statement". https://web.archive.org/web/20000817222927/http://www.wi-fi.org/mission.asp.

Acronyms

A2 MPDU Address 2 16, 18

AAD Additional Authentication Data 16, 17

AB*S*AB Mantin's Digraph Repetition 15

ACS Auto-Configuration Server 2, 3, 21, 24, 25, 31, 32, 48, 59

AES Advanced Encryption Standard 15, 17, 37

AP Access Point 17, 20

ASCII American Standard Code for Information Interchange 12

ATA Analog Telephone Adapter 57

AWS Amazon Web Services 2, 39, 40, 42

BSSID Basic SSID 29, 38

CBC Cipher Block Chaining 15, 76

CCM CTR with CBC-MAC 15–18, 76

CCMP CCM Protocol 2, 15–18, 29

CLI Command Line Interface 56

CPE Customer-Premises Equipment 2, 3, 6–8, 21, 24–42, 44–48, 50–56, 59, 60, 76

CR Carriage Return 56

CRG Commercial Residential Gateway 2, 3, 40, 52–58, 60

CTR Counter 15, 76

CWMP CPE WAN Management Protocol 2, 7, 21, 24, 25, 31, 32, 44, 46, 59, 60

DA Destination Address 14

DHCP Dynamic Host Configuration Protocol 25

DNS Domain Name System 25, 33

DSL Digital Subscriber Line 53, 79

EAP Extensible Authentication Protocol 12

EC2 Elastic Compute Cloud 2, 39, 40, 42

EIV Extended IV 13, 15

EPROM Erasable Programmable Read-Only Memory 35, 39

ESSID Extended SSID 29, 38, 39

FM Fluhrer-McGrew 15

FMS Fluhrer, Mantin, and Shamir 12

GCM Galois/Counter Mode 17, 18, 77

GCMP GCM Protocol 2, 17–19

GPON Gigabit Passive Optical Network 33, 53, 59

HDM Home Device Manager 48

HTTP Hypertext Transfer Protocol 2, 7, 22, 25, 33, 34, 39–41, 44–47, 52, 53, 55, 58–60

IA Integrity Algorithm 10, 11

ICMP Internet Control Message Protocol 30, 31

ICV Integrity Check Value 10, 11, 13, 15

ID Identifier 52, 54, 55, 57, 58

IEEE Institute of Electrical and Electronics Engineers 3, 8, 9, 12, 15, 17

IoT Internet of Things 20

IP Internet Protocol 6, 25, 30, 32, 33, 44, 52–54, 59, 79

IPTV Internet Protocol Television 2, 6, 32, 33, 57, 58

ISP Internet Service Provider 6, 7, 21, 24, 25, 30–32, 37, 41–43, 45, 48–52, 55, 57–60

IV Initialization Vector 9–15, 77

KID Key ID 16

LAN Local Area Network 8, 25, 30, 31, 33, 58, 77, 79, 80

LF Line Feed 56

LMSC LAN/Metropolitan Area Network (MAN) Standards Committee 8

MAC Medium Access Control 10, 11, 14–16, 33, 38, 39, 76, 78

MAN Metropolitan Area Network 8, 77

MD5 Message Digest 5 22, 31

MIC Message Integrity Code 12–18

MIPS Microprocessor without Interlocked Pipelined Stages 27

MPDU MAC PDU 2, 10, 11, 13–18, 76

MSDU MAC Service Data Unit 14

MTD Memory Technology Device 35

NFC Near Field Communication 19, 20

OUI Organizationally Unique Identifier 31

OWE Opportunistic Wireless Encryption 9

PBC Push Button Configuration 19, 29, 39

PBKDF2 Password-Based Key Derivation Function 2 12

PDU Protocol Data Unit 10, 13, 78

PIN Personal Identification Number 19, 20, 29, 39

PMK Pairwise Master Key 12, 17

PN Packet Number 16–18

PPPoE Point-to-Point Protocol over Ethernet 2, 41, 52, 53

PRNG Pseudo-Random Number Generator 9–11, 14, 20, 39

PSK Pre-Shared Key 12, 15, 17, 29

PTW Pyshkin, Tews, and Weinmann 12

QR Quick Response 20

RC4 Rivest Cipher 4 11, 14

RJ-11 Registered Jack 11 52, 55

S/N Serial Number 33, 53, 59

SA Source Address 14

SAE Simultaneous Authentication of Equals 17, 19

SIP Session Initiation Protocol 2, 7, 21–23, 32, 42, 43, 49–51, 54–57, 60

SK Shared Key 9–12, 14

SLAAC Stateless Address Autoconfiguration 52, 53

SSH Secure Shell Protocol 33, 35, 37, 56

SSID Service Set ID 12, 29, 33, 76, 77

STA Station 17, 20

STB Set-Top Box 58

TA Transmitter Address 14

TCP Transmission Control Protocol 15, 21, 30, 31

TFTP Trivial File Transfer Protocol 31, 35

TK Temporal Key 13, 16, 17

TKIP Temporal Key Integrity Protocol 2, 12–15, 29, 39, 79

TLS Transport Layer Security 21, 32, 42, 49, 60

TR-069 Technical Report 069 21

TSC TKIP Sequence Counter 14

TV Television 6

UDP User Datagram Protocol 21, 30, 32, 42

URL Uniform Resource Locator 31, 32, 45, 48

VDSL Very High-Speed DSL 53

VLAN Virtual LAN 2, 33, 41, 52, 54, 55, 57, 58

VoIP Voice over IP 6, 32, 33, 42, 49, 52, 54, 55, 57, 60

WAN Wide Area Network 7, 30, 31, 33, 44, 59, 60, 76

WEC Wi-Fi Easy Connect 8, 20

WECA Wireless Ethernet Compatibility Alliance 8

WEO Wi-Fi Enhanced Open 8

WEP Wired Equivalent Privacy 2, 8–15, 17

Wi-Fi Wireless Fidelity 3, 8, 9, 12, 15, 17, 19, 29, 38, 59, 79, 80

WLAN Wireless LAN 8

WPA Wi-Fi Protected Access 2, 8, 12, 15, 17–19, 29, 38–40

WPS Wi-Fi Protected Setup 8, 19, 20, 29, 33, 39, 60

WSC Wi-Fi Simple Config 19

Symbols

b bit 8

c iteration count, a positive integer 12

DK derived key, an octet string 12

$dkLen$ length in octets of derived key, a positive integer 12

G giga, 10^9 8, 29, 38, 39, 42

H hash 42

Hz hertz 8, 29, 38, 39

KS key stream 10, 11

M mega, 10^6 8

N an integer 10, 11

P password, an octet string 12

$/s$ per second 8, 42

S salt, an octet string 12