Exposed by Default: A Security Analysis of Home Router Default Settings and Beyond

Junjian Ye, Xavier de Carné de Carnavalet, Lianying Zhao, Mengyuan Zhang, Lifa Wu, and Wei Zhang

***Abstract*—With the popularity of the internet, home routers have become crucial for the security of home networks. However, according to the results of our user survey, home routers are often deployed with minimal changes to the factory default settings, which may pose risks to user security and privacy. To systematically evaluate potential risks, we designed a threat-model-based framework and conducted a comprehensive analysis of 40 commercial off-the-shelf home routers from 14 brands. We found a variety of security issues, among which incorrect implementation of TLS is the most common. To improve the efficiency of manually detecting TLS certificate validation vulnerabilities without real routers, we proposed a heuristic method that can narrow down the search scope in firmware and proved its effectiveness with 30 available firmware images of the routers we purchased. Moreover, we evaluated the security of custom remote management protocols and found several cryptographic misuses. Finally, we proposed several recommendations for extending the analysis framework and discussed our ideas about automatically detecting security issues to highlight the need for heightened scrutiny of default settings and inspire other researchers.**

***Index Terms*—Home router security, default settings, manual analysis, TLS misconfiguration.**

I. Introduction

As of April 2024, there were 5.44 billion internet users worldwide, which amounted to 67.1 percent of the global population [1]. A modem together with a router has become a common method to access the internet at home. All traffic from devices at home (e.g., laptops, smartphones, and various IoT devices) goes through home routers, so routers play an important role in home network security. However, to meet the needs of consumers, home routers come with various features and customizable settings, which also brings security risks.

Numerous attacks target home routers, including botnet malware infections [2]–[4] and attacks on wireless protocol implementations [5], [6]. A body of research also aims at automating the discovery of vulnerabilities in routers from firmware images to uncover authentication issues [7], privilege escalation [8], command injection [9], and information disclosure [10]. However, a less studied issue stems from the quality of the settings being applied to the routers. For instance, home routers used to support the deprecated WEP (Wired Equivalent Privacy) protocol years after a powerful attack offering to recover the shared key was published [11], [12].

Given that routers offer a range of customizable settings, we first established through a user survey whether users tend to retain the default configurations. The results indicate that home routers are often deployed with minimal changes to the factory default settings, which is also consistent with the results of a user survey conducted in the UK [13]. We consider the resulting configuration as the *initial default settings*. This will very likely be what the users end up with when using the router until a future event brings attention to the router again. Therefore, any insecurity left in such initial default settings will possibly affect a large population. For instance, users might believe their WiFi network is secure because they must enter a "security key" to access it. However, if the router defaults to a weak security protocol, even though it supports stronger ones, an attacker could still intercept or eavesdrop on sensitive data. This could be even worse than unprotected connection as the user is unaware and may take decisions based on wrong assumptions (e.g., otherwise sensitive transactions could have been avoided).

Concerns also arise from initially disabled features that, when activated by a single click, impose a multitude of additional default settings, which we term *deep default settings*. Similarly, as these settings become functional immediately upon activation, users are less likely to modify them. If overlooked, such settings could obscure a comprehensive security analysis of home router default settings. As an example, a typical feature which is usually disabled by default is guest network (to give temporary network access to a visitor). This disabled feature might not catch the attention of a security analysis but once enabled by the user (e.g., a friend drops by), guest network may default to improper settings and issues from the initial defaults would suddenly apply.

Next, we seek the answers to two questions: "*Will the default settings of home routers pose any security risks?*" (initial defaults) and "*Will the default settings of initially-deactivated features of home routers pose any security risks once activated?*" ("deep" defaults), we designed a comprehensive router default settings security analysis framework based on our defined threat model of home routers, and analyzed 40

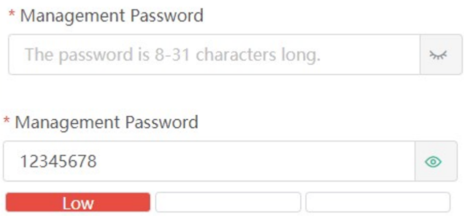
Manuscript received August 24, 2024; revised November 4, 2024; accepted November 16, 2024. This work was supported by the National Key Research and Development Program of China with 2019YFB2101704. *(Corresponding author: Lifa Wu.)*

Junjian Ye, Lifa Wu and Wei Zhang are with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: jjy470742953@gmail.com; wulifa@njupt.edu.cn; zhangw@njupt.edu.cn).

Xavier de Carné de Carnavalet is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: xdecarne@polyu.edu.hk).

Lianying Zhao is with the School of Computer Science, Carleton University, Ottawa K1S 5B6, Canada (e-mail: lianying.zhao@carleton.ca).

Mengyuan Zhang is with Computer Systems, Vrije Universiteit Amsterdam, Amsterdam 1081 HV, Netherlands (e-mail: m.zhang@vu.nl).



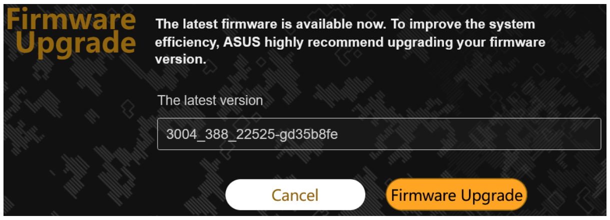
* Management Password

The password is 8-31 characters long.

* Management Password

12345678

Fig. 1. A screenshot of the immersive question about choosing passwords.



The latest firmware is available now. To improve the system efficiency, ASUS highly recommend upgrading your firmware version.

The latest version

3004_388_22525-gd35b8fe

Fig. 2. A screenshot of the immersive question about firmware update.

home routers available at flagship stores on popular shopping platforms that still receive firmware updates after 2018.

This work is an extension of an AsiaCCS 2024 publication [14]. In addition to our user survey in Section II-B, we tested several new features (DDNS services) and behaviors (e.g., against ARP spoofing and CSRF attacks), leading to an increased number of exploitable security issues from 30 to 89. Moreover, we investigated the implementation of custom remote management protocols across several brands of routers and evaluated their security in Section VI. To inspire researchers interested in this work, we also proposed several recommendations for extending the analysis framework in Section III-D and discussed our additional efforts and ideas about automatically detecting security issues in Section VII.

According to our analysis results, we found insecure default settings and behaviors related to Wi-Fi security protocols, setup wizard, guest network, WPS, IPv6, TLS, and router reset. Among them, incorrect implementation of TLS is the most common security risk for home routers due to the lack of TLS certificate validation. Therefore, as a part of this extended work, we proposed a heuristic method to detect these vulnerabilities in router firmware and evaluated it with 30 available firmware images of the routers we purchased in Section V. The results show that our method can significantly narrow down the search scope of TLS certificate validation vulnerabilities and improve the efficiency of manual analysis without real routers.

Our main contributions are as follows:

- We introduce a security evaluation framework for default settings in home routers, with a focus on deep settings, while factoring in four types of adversaries.
- We uncover numerous security issues and shed light on weak default security protocols, incorrect implementation of TLS, improper configuration guidance for users, unencrypted firmware updates, IPv6 without NAT and default firewall, and concealed WPS support. We are currently reporting 89 exploitable security issues to the vendors.
- We propose, implement and evaluate a heuristic method to narrow down the search scope of TLS certificate validation flaws and improve the efficiency of manual analysis without real routers.
- We explore the implementation of custom remote management protocols and discuss security implications.
- We discuss our additional efforts and ideas based on our findings to lay the path for future work on router security.

II. Motivation

This section explains why we decided to analyze the security of home routers' default settings with the results of our user survey.

A. Setup Wizard

Home routers may not be fully operational after initial power-up; users may be prompted to visit the router's management page manually or get automatically redirected there by their browser. The setup wizard then guides users to configure the network, set Wi-Fi and admin passwords, and update firmware, as shown in Fig. 3.

The setup wizard is typically the only stage that mandates users to check and modify settings in the entire life-cycle of a home router. Worse, for the sake of convenience, a feature called "plug-and-play" that allows users to start enjoying their routers without the setup wizard is becoming increasingly common in home routers. Additionally, technicians of Internet Service Providers (ISPs) may help users complete the setup wizard on their behalf, especially for routers borrowed/bought from the ISP.

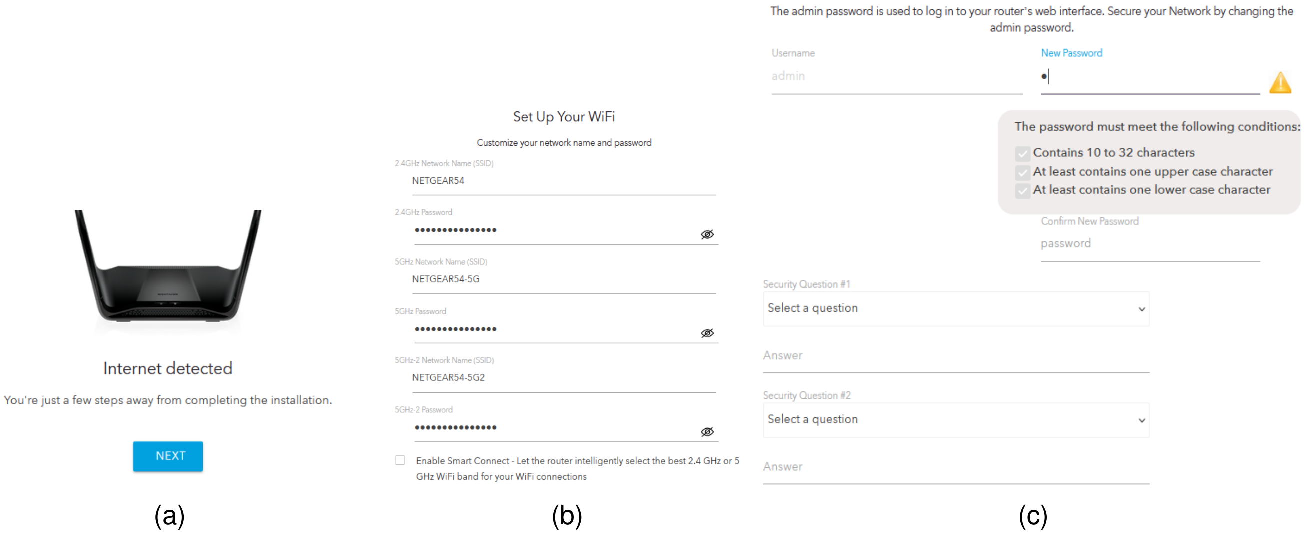
Therefore, we find that checking and modifying the default settings of home routers is usually not mandatory.

B. User Survey

It is not surprising that users tend not to change the default settings of their home routers, but only a small-scale user survey [13] conducted in the UK can support this hypothesis. To understand the configuration habits and attitudes of home router users in depth, we conducted a more comprehensive user survey by spreading an online questionnaire in several countries and regions, especially in China.

The survey consists of four parts: Firstly, we collect the demographics of respondents, including age, gender, education level, occupation, region, familiarity with information technology (IT), and whether they subscribed to an internet plan. Then, we build questions to understand whether and how the participants configure their home routers. Next, we focus on whether participants might change a Wi-Fi or admin password when immersed into a relevant prompt (see Fig. 1) and ask about their password selection strategy. Finally, we provide an immersive question with the prompt on Fig. 2 to study the user behavior in terms of firmware update during the setup wizard.

We hosted the Chinese version of this questionnaire with Wenjuanxing [15] and the English version with SurveyMonkey [16]. The questions and options in both versions are equivalent. Those two online surveys were first distributed among



Internet detected

You're just a few steps away from completing the installation.

NEXT

Set Up Your WiFi

Customize your network name and password

2.4GHz Network Name (SSID)
NETGEAR54

2.4GHz Password
•••••••••••••••

5GHz Network Name (SSID)
NETGEAR54-5G

5GHz Password
•••••••••••••••

5GHz-2 Network Name (SSID)
NETGEAR54-5G2

5GHz-2 Password
•••••••••••••••

□ Enable Smart Connect - Let the router intelligently select the best 2.4 GHz or 5 GHz WiFi band for your WiFi connections

The admin password is used to log in to your router's web interface. Secure your Network by changing the admin password.

Username
admin

New Password
•

The password must meet the following conditions:
☑ Contains 10 to 32 characters
☑ At least contains one upper case character
☑ At least contains one lower case character

Confirm New Password
password

Security Question #1
Select a question

Answer

Security Question #2
Select a question

Answer

Fig. 3. An example of setup wizard. (a) Configure network. (b) Set Wi-Fi password. (c) Set administration password.

the participants we have personal connections with. Then, those participants forwarded the survey to their acquaintances or in their social groups. We spread the links of two surveys out from August to September 2023. The surveys reached China (Mainland), China (HKG, Macao, or Taiwan), North America, and Europe. Our participants span a wide range of ages, from 18 to 60 and above. In the end, we collected a total of 393 responses (322 Chinese and 71 English).

Survey Findings. 94.7% (372/393) of participants subscribe to a home internet plan (DSL, cable, fiber) with an ISP and deploy a router at home, which means that home routers have been widely used. Only 34.7% (129/372) of participants completed the setup wizard of their home routers themselves. Most of other routers are configured by technicians of ISPs. Only 28% (104/372) of participants tried to understand the questions and settings during the setup wizard, and have configured at least one feature on the management page. 45.4% (169/372) of participants retain the default Wi-Fi passwords or admin passwords. 31.6% (124/393) of participants prefer choosing a password that only meets the minimum password strength requirements, and only 17.6% (69/393) tend to generate a random password rather than choosing a simple or personal-information-based password. 56% (220/393) of participants tend to click “Firmware Upgrade” when seeing Fig. 2.

These results not only confirm the assumption that home routers are widely used and their users tend not to modify default settings, but also motivate us to further investigate the security of routers in default configurations and provide inspiration for the design of our analysis framework.

III. Analysis Design

In this section, we first introduce our threat model of home routers and security analysis environment. Based on them, we then describe our methodology to extract default settings from real routers and measure their behaviors. Finally, we propose several recommendations about how to continuously improve this framework in the future.

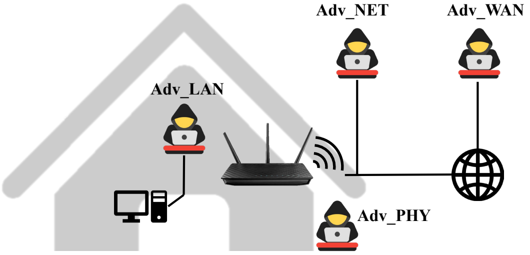


Fig. 4. Threat model.

A. Threat Model

When a router is working and connected to the internet, it may face potential threats from various adversaries. To better study the security of various default settings in home routers, we divide adversaries into the following four types:

- Adv_WAN: Adversaries in the Wide Area Network (WAN) e.g. the Internet. As the WAN interface of routers is exposed on the internet, adversaries can discover open services (notably through specialized search engines e.g., Shodan [17], Zoomeye [18]) and try to exploit them.
- Adv_LAN: Adversaries in the Local Area Network (LAN). Compromised user devices (e.g., by malware), or simply curious guests that are already connected to the router's local network have access to certain of the router's features. Also, adversaries who have managed to get connected to Wi-Fi (LAN) are included.
- Adv_PHY: Adversaries physically close to the router. The Wi-Fi signal coverage of the router is usually sufficient for adversaries to mount attacks to exploit wireless protocols from outside the user's home. For instance, adversaries in close proximity can attempt to break the security protocol to connect to Wi-Fi [19].
- Adv_NET: Adversaries on the network path. Adversaries such as the ISP can passively eavesdrop on the outgoing traffic and mount man-in-the-middle attacks to tamper

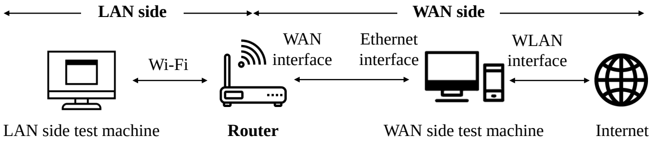


Fig. 5. Environment setup in the analysis framework.

with traffic. Such adversaries are in line with the Dolev-Yao model [20], where they are capable of accessing arbitrary messages sent on the network, only limited by cryptographic capabilities, e.g., can't break encryption with a high-entropy key.

Curious router manufacturers/vendors may also threaten users' security and privacy e.g., by implementing backdoors; we do not consider them in our threat model.

B. Analysis Environment

To conduct a comprehensive analysis of both the LAN and WAN sides of the router, we built an analysis environment based on real routers. We first interpose the WAN interface of the router to observe outgoing traffic, provide a realistic network configuration to the router (especially with regard to IPv6), and conduct port scanning. Our position is thus representative of Adv_WAN and Adv_NET. Then, we connect to the Wi-Fi network of the home router to access the web management page, complete the setup wizard, and conduct further analysis from the perspective of Adv_LAN and Adv_PHY; see illustration in Fig. 5.

C. Analysis Framework

As shown in Section II-B, users tend to keep the default settings of home routers, which makes us realize the importance of default settings. To evaluate their security, we designed a comprehensive security analysis framework based on our threat model and analysis environment. This framework can be divided into two stages: initial analysis and deep analysis, which not only collects the settings shown on the web management page, but also verifies whether critical settings are properly applied. In doing so, we also launch active tests such as port scanning. The sensitive features considered in our framework are listed in Table I.

TABLE I
CATEGORIES OF SETTINGS CONSIDERED AND RELATED THREATS

Categories of settings considered	Adv_LAN	Adv_PHY	Adv_NET	Adv_WAN	Common issues/vulnerabilities
Initial Defaults					
Wi-Fi and admin pass-phrase/word	✓	✓			Weak hard-coded default passphrases, lack of proper guidance for setting strong passphrases
Wi-Fi security protocol		✓			Support for outdated protocols (e.g. WEP, WPA, WPA2-TKIP)
WPS		✓			Hard-coded WPS PINs, absence of attempt rate-limiting
Local web access	✓			✓	Absence of TLS, login rate-limiting/CAPTCHAs or countermeasures to ARP spoofing and CSRF attacks
Firmware update mechanisms			✓		Absence or incorrect implementation of TLS or integrity verification
Exposed services	✓			✓	Concealed sensitive services (e.g. Telnet, FTP, UPnP, HTTP)
Deep Defaults					
Guest network		✓			The same issues as Wi-Fi, absence of isolation from main network
Remote web access			✓	✓	The same issues as local web access, self-signed TLS certificates
Telnet/SSH	✓			✓	Weak hard-coded default passphrases
IPv6				✓	Lack of IPv6 NAT or firewall
DDNS			✓		Absence or incorrect implementation of TLS
Cloud account			✓		Absence or incorrect implementation of TLS, lack of proper guidance for setting strong passphrases
App			✓		Absence or incorrect implementation of TLS
UPnP	✓			✓	Vulnerable port mapping function
External storage	✓		✓	✓	Weak hard-coded default passphrases, absence or incorrect implementation of FTP over TLS
Reset	special case				Incomplete reset

1) Setup Wizard: As introduced in Section II-A, the setup wizard guides users to configure the network, set the Wi-Fi and admin passwords, and update firmware. We first evaluate whether the router can be plug-and-play, disregarding instances where routers that could be plug-and-play include a "quick setup" guide that, despite recommendations, is not necessary for completing the setup process. If the setup wizard is necessary, according to the result of the user survey in Section II-B, users may try to avoid making decisions and simply click "next" or "skip" when possible, especially when the setup wizard is completed by technicians of ISPs. When users without necessary knowledge have to make decisions, visual hierarchical design [21] can affect their decisions. Therefore, when a page of the setup wizard requires to take an action, we can infer the buttons that manufacturers want users to click on and the behavior of most real users based on the principle of visual hierarchy. For example, users tend to click "Firmware Upgrade" when seeing illustrations in Fig. 2 (as shown in Section II-B) because it is yellow and conspicuous. Our goal is to complete the setup wizard as the average user and record default settings and behaviors.

2) Initial Analysis: The initial analysis is concerned about obtaining the initial defaults of the router. The sensitive settings that need to be recorded are as follows.

Wi-Fi and admin passwords/passphrases. Wi-Fi password is the password required for users to connect to the Wi-Fi of the router, while admin password is required to log in to the management page of the router. As shown in Section II-B, many users retain default passwords and prefer choosing a password that only meets the minimum password strength requirements when setting their own passwords. Therefore, we first record the default password/passphrase in each router (if any, and usually provided on the label) to evaluate their strength. Then, we record whether changing the default password is required and try to set the shortest and simplest password that can be accepted to test the minimum strength requirements. If Wi-Fi passwords are predictable for Adv_PHY or admin passwords for Adv_LAN, adversaries may be able to access the LAN or control the router.

Wi-Fi security protocol. Wi-Fi is a Wireless Local Area Network (WLAN) technology based on the IEEE 802.11 standard [22]. Due to the open nature of wireless communications, wireless devices communicate through encryption protocols such as Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, and more recently WPA3 (standardized in 2018) to prevent Adv_PHY from accessing the LAN. We first record the default Wi-Fi security protocol shown on the management page. Sometimes, a hybrid mode is supported to provide backward compatibility with older client devices. We also measure the effective protocols supported by the router by examining beacon frames sent by the router. Beacons carry information about the supported WPA version ciphersuite (TKIP or CCMP). We check both the main and guest networks as well as all frequency bands (2.4 and 5GHz).

WPS. Wi-Fi Protected Setup (WPS) is a simplified Wi-Fi connection method. Typically, users can either enter an 8-digit PIN displayed by the router or press a WPS button physically or in a management page (PBC) to connect to Wi-Fi without entering the passphrase. However, the PIN method is vulnerable to brute force attacks [5]. We can extract the status of WPS from the beacon frames advertised by the router to verify its support for WPS PIN and whether the status is "Configured" (functional) or "Locked" (non-functional) [23]. In the former case, we attempt to record and test the given PIN (printed on the label or available in the relevant configuration page) or launch a brute-force attack leveraging `reaver` [24] when no PIN is known to check the usability of WPS. We also detect whether a rate-limiting mechanism is in place, imposing restrictions on brute-force attacks.

Local web access. A router's management webpage is normally accessible by anyone on the LAN side. Adv_LAN may conduct ARP spoofing [25] to eavesdrop on the traffic between other users and the router and learn sensitive information such as the admin password. Therefore, we first leverage `arpspoof` [26] on the LAN side to check whether the router is vulnerable to ARP spoofing. Then, we record the protocol employed to access the management page (HTTP or HTTPS). If HTTP is the default protocol, we analyze the login packets to check whether admin passwords are transmitted in plaintext. Similar to WPS, we also measure countermeasures to brute-force login attempts, through rate-limiting or CAPTCHAs. Additionally, Cross-Site Request Forgery (CSRF) attack is also a potential risk. By conducting CSRF attacks, Adv_WAN can force users to execute unwanted actions (e.g., enable remote web access) on the management page in which the user has authenticated [27]. To check whether the router is vulnerable to CSRF attacks, we build a malicious website that can force visitors to change router settings on the management page and access it from the LAN side after we have logged in to the management page. We capture and analyze the traffic on the LAN side to check whether the router will accept the router setting modification request.

Firmware update mechanisms. Routers receive firmware updates that can fix security vulnerabilities and improve security. There are three ways to update router firmware: automatic update, user-initiated update and manual update. Automatic update can help users check and automatically update firmware to the latest version at regular intervals, so we check whether it is enabled by default on the management page. User-initiated update means that users can click on the "check firmware version" or "update firmware" buttons on the management page to trigger the same process. Both of the above methods need to interact with servers on the internet. We run `Wireshark` and `mitmproxy` [28] to capture firmware update packets and check whether TLS is correctly implemented. If HTTPS is not employed or the TLS certificate is not properly validated, Adv_NET may downgrade firmware or tamper with the update files through man-in-the-middle (MITM) attacks [29], [30].

Exposed services. Each exposed service increases the attack surface [31] of home routers. In particular, a service exposed on the WAN interface will be indexed by search engines such as Shodan. Adv_LAN and Adv_WAN can discover the open ports on routers' LAN and WAN interfaces, respectively, and exploit vulnerabilities to attack routers [32], especially when the exposed service is hidden (not mentioned on the management page) and cannot be disabled. To discover hidden services, we leverage `Nmap` [33] to scan for the router's open ports and corresponding services and their versions on the WAN and LAN interfaces. To speed up scanning of 65,535 ports, we run `Nmap` in parallel on smaller port ranges.[1]

3) Deep Analysis: The deep analysis focuses on the sensitive features supported by the router, after we enable them. The security analysis methods for these features are as follows.

Guest network. A guest network is a separate wireless SSID advertised by the router with limited functionalities and targeted for user's guests. The network usually provides isolation from other clients of the main network and prevents guests from accessing the management page [34]. There are two types of guest networks: open networks without Wi-Fi password that instead leverage a captive portal to ask for the guest password after users connect to it; and secured network protected by the guest Wi-Fi password. We record the type of the guest network first. If it is protected by a Wi-Fi security protocol, we investigate its Wi-Fi configurations and passphrase requirements as mentioned earlier. To test the effectiveness of isolation capability, we check whether we can access the devices in the main network or log in to the management page as an administrator from the guest network.

Remote web access. Remote management is a feature that allows users to access management pages from the WAN side via the internet. It usually requires the help of companion apps or remote web access. In the case of remote web access, the public IP address of the router's WAN interface with a designated port number allows the access from the WAN side directly, which can bring security risks [35], [36]. As with local web access, we can examine the security of remote web access the same way (except ARP spoofing and CSRF attacks), but with additional considerations since it is exposed to the internet. We also check whether the TLS certificate of the router is self-signed. If the subject and issuer of a certificate are the same, it means that this certificate is self-signed and users may still be vulnerable to MITM attacks [37]. Additionally, we record the version of TLS because older versions may have known vulnerabilities.

Telnet/SSH. Telnet and SSH are commonly used protocols

[1] `nmap -sV -T5 -p 1-100`, then `-p 101-1000`, etc.

for remote management. Routers support them for remote debugging; Therefore, they tend to hide the status of Telnet and SSH (the default usernames and passwords of these services) from the users. However, Adv_WAN and Adv_LAN may exploit Telnet or SSH to access the terminal of the router if it is not configured properly (e.g., employ weak hard-coded passwords [32]). We first check whether Telnet or SSH is enabled by default on the LAN or WAN interfaces of the router according to the management page and the results of port scanning. If the router supports Telnet or SSH, we try to connect to it and check whether a username and password are required to login and whether the default username and password are weak.

IPv6. In IPv4, Network Address Translation (NAT) is a mechanism that is often seen as necessary due to the scarcity of global IPv4 address space. Though not a primary purpose of NAT, it provides a “better-than-nothing” security boundary as well by translating and masking private IP addresses, thereby making it more challenging for outside attackers to initiate connections to internal devices [38]. IPv6 eliminates the need for NAT thanks to globally unique addresses, making internal hosts directly reachable via the internet and poses new security risks. Therefore, the security of IPv6 depends on whether the IPv6 addresses of connected devices are generated by the router, whether they are predictable, whether IPv6 NAT is leveraged by default, and whether the IPv6 firewall blocks IPv6 packets from the WAN side.

First of all, we leverage `dhcpd6` and `radvd` to build a virtual stateless DHCPv6 environment to assign IPv6 address to the router and capture traffic between the router and the connected device when connecting to the router’s Wi-Fi. Then, we can find out whether the router employs SLAAC, stateful DHCPv6 or stateless DHCPv6 to assign IPv6 addresses. Only in the stateful DHCPv6 IPv6 addresses are generated by the router, and in the other two methods IPv6 addresses are generated by the device itself. We record the WAN IPv6 addresses and MAC addresses of the router and connected devices to check whether the generated addresses are predictable. Usually, the generation algorithms of different devices are different.

To check whether IPv6 NAT is leveraged by default, we ping the internet from the LAN side and compare the source IPv6 addresses of ping request packets captured on the LAN side and WAN side. If they are different, it means that the router employs NAT, and the IPv6 address of the device is only used in the LAN, so the adversary cannot directly connect to the device with the IPv6 address. When there is no NAT, we set up a simple web server on the LAN side and try to access it based on the IPv6 address from the WAN side to check whether an IPv6 firewall that blocks incoming connections from the internet is enabled by default. If not, it means that the connected devices are being exposed to the internet, bringing potential security risks to users. To check the blocking range of the firewall, we deploy the server on the common port 80 and the private port 8000 for testing.

DDNS. Dynamic Domain Name Server (DDNS) can point internet domain names to dynamic IP addresses of routers to support external storage, remote web access and other features that need to be accessed via the internet. DDNS service is usually provided by third-party servers, so users must register an account of service providers (e.g., No-IP [39], DynDNS [40], and Oray [41]) first and bind it with a domain name. Then, routers will send the account information and current IP address to the DDNS server, so that users can access the router via the domain name. Similar to firmware update, the communication between the router and the DDNS server is also vulnerable to MITM attacks if TLS is not implemented properly. Adv_NET may eavesdrop on the account name/password or change the IP address to the IP address of a malicious server. Therefore, we run `Wireshark` and `mitmproxy` to capture and check the traffic when we log in to the DDNS account on the management page.

Cloud account. Certain router manufacturers provide cloud account services for users. These accounts can be bound with routers and their companion apps for remote access and management or used for the DDNS service. If the cloud account is compromised, the bound router may be controlled by adversaries. Therefore, account password strength requirements should be very strict. Additionally, similar to firmware update, routers also need to communicate with servers when users log in to their cloud accounts, so we capture login packets and check whether the password of the account can be intercepted by Adv_NET.

App. To facilitate accessing and managing the router, manufacturers develop companion apps for their routers. These apps can be bound to routers or manufacturer accounts and communicate with routers directly by Wi-Fi or through a server on the internet. If users want to rely on the app to remotely control the router, the manufacturer’s server needs to forward packets to the router. Similar to firmware update, Adv_NET can control the router by hijacking the packets between the router and the server [42], so we capture the traffic between the router and the server to check whether HTTPS is employed and whether the TLS certificate is validated properly when binding the app with the router and leveraging the app to remotely manage the router.

UPnP. Universal Plug and Play (UPnP) is an architecture for easy and robust connectivity of many sorts of devices in homes, offices, and elsewhere [43]. For routers, UPnP can help peer-to-peer software in the LAN access the internet more smoothly by automatically configuring port mapping. The main security risk of UPnP comes from the port forwarding function, which can be exploited to carry out MITM attacks by adding port mappings [44]. `PortMapper` [45] can be leveraged to detect whether the router supports UPnP port forwarding and whether new port mappings can be added on the LAN and WAN side.

External storage. External storage is a feature that supports users to access the content in USB devices connected to the router. Generally, there are three different access methods: SMB-based, web-based over HTTP or HTTPS, and FTP-based. SMB is limited to file sharing within the LAN, while the latter two methods support remote access. To prevent exposure of the user's private data stored in the external storage, we check whether authentication is enabled and verify the strength of default passwords. Additionally, similar to remote web access, properly implemented TLS is also important for remote web-based and FTP-based access methods to prevent attacks from Adv_NET. For remote FTP, we capture the traffic on the WAN side when leveraging `FileZilla` [46] to connect to the FTP server on the router's WAN interface. The captured traffic allows us to validate whether the traffic is encrypted by TLS and whether the router's TLS certificate is valid. We also record the version of TLS.

Reset. Router refurbishment and second-hand router trading are very common. Usually, users reset routers to dispose of their sensitive data. However, there may be residual data in the router due to improper implementations of the reset function. Given the data extraction cost, we do not consider the sensitive information left in the flash chip of the router [47], [48], but only focus on the information available on the management page, including Wi-Fi SSID & password, admin username & password, cloud account, third-party accounts (e.g., DDNS, VPN, PPPoE, email, and FTP) and logs. Additionally, the binding status between the companion app and the router is also sensitive, and it is necessary to ensure that the router is automatically unbound from the app when reset. We reset the router after modifying and recording all sensitive information. If the router offers different reset options, we record them and select the default recommended options. Thereafter, we try to skip the setup wizard and check whether such sensitive information is retained and whether we can still use the app to remotely manage the router. If the wizard cannot be skipped, we will pay additional attention to whether the sensitive information is kept as default values in the setup wizard. Note that we consider the reset feature as a special case as it is expected to restore to the default settings (which is our focus) and may pose a security risk; however, it does not map to any of our four types of adversaries.

D. Extensibility

Our analysis framework is a minimal security testing manual that can not cover all potential security issues (which is also impossible), but we believe that routers should pass these tests to ensure a minimal level of security. Moreover, this framework is extensible because based on the threat model and analysis environment, we can continuously add new sensitive features, analysis methods, and security indicators into it. To encourage researchers interested in this work to participate and continuously improve this framework, we summarize our ideas for designing this framework and propose the following recommendations for extending it:

- For existing features of home routers, newly disclosed vulnerabilities may bring overlooked security issues to our attention and encourage us to add new testing items into our framework. It should be noted that our framework focuses on vulnerabilities caused by the misconfigurations of common features because they may be widely present, easy to detect but easily overlooked, which means that traditional binary vulnerabilities (e.g., buffer overflow, and command injection) are not within our scope of consideration.
- For new features that may be commonly supported in the future, we should first understand the usage, purpose, principles, and common types of this feature. Then, we need to consider which sensitive information of users or important permissions of the router will be involved in the usage process. Finally, we consider how to steal them from the perspective of each adversary in our threat model. Following this approach, we can identify potential security issues in new features.
- Security issue in our framework is a broader concept than vulnerabilities. We are concerned about the security risks that routers with default settings may face, which means that issues that can be avoided by simple modification of settings cannot be ignored either. For example, a router enables "Preserve network configuration when restoring factory settings" by default when users reset it. Users can disable this option to ensure that all privacy is cleared, but we still think they should be taken seriously. In conclusion, we believe that all potential risks (from insecure default settings to incorrect code implementations) are valuable and should be added into our framework, even if some of them may not be confirmed by manufacturers as security vulnerabilities.

IV. Analysis Results

We analyzed 40 home routers based on a selection detailed in Section IV-A, by following our analysis framework (introduced in Section III-C). The routers were running the factory firmware unless the setup wizard explicitly recommends updating the firmware before continuing, mimicking common user behaviors. The factory version will run for a period of time until the firmware is automatically updated(if supported).

We share below the essential findings from the analysis. The key results are summarized in Table II by brand and model (we have anonymized model names for ethical reasons). We found a total of 105 potential security issues (marked with "!!" in Table II), out of which, we have confirmed 89 to be exploitable in the latest version and reported them to manufacturers or CNVD/CVE (marked with "*" in Table II). More information about detailed analysis results and vulnerability reporting will be updated at https://github.com/YjjNJUPT/extended_version_of_default_settings.

It is noteworthy that our analysis may have covered only a small portion of all available routers on the market. Our research objective is to understand trends to help manufacturers improve home router security, rather than guiding consumers in choosing a router.

A. Router Selection

Home routers are diverse in vendors, regional markets, popularity, and firmware filesystems. Due to the prohibitive cost of buying all available router models, we choose to evaluate a selection of commercially off-the-shelf routers that can reflect both popularity and diversity.

TABLE II
SUMMARY OF POTENTIAL AND CONFIRMED SECURITY ISSUES FROM OUR ANALYSIS OF 40 ROUTERS

		Has IPv6 NAT disabled	Has IPv6 firewall disabled	Uses predictable IPv6 address	Supports WPA v1	Supports TKIP encryption	Does not require setup wizard	Offers open Wi-Fi by default	Offers no or weak admin pwds	Supports configured WPS	Uses PIN hidden from users	Accepts WPS PIN attempts	Does not rate-limit PIN attempts	Uses Hard-coded PIN	Is vulnerable to ARP spoofing	Transmits admin pwd insecurely	Is vulnerable to CSRF attacks	Checks version info insecurely	Downloads FW image insecurely	Has insecure app remote mgmt	Has insecure DDNS (No-IP/DynDNS/Oray)	Does not change Wi-Fi/admin pwd	Allows open Wi-Fi to remain	Offers open guest network	Supports WPA v1	Supports TKIP encryption	Allows guests to access mgmt page	Leaves sensitive info after reset
Brand	**Model**	IPv6			WiFi		Plug&Play			WPS					Local			TLS				Setup		Guest				
360	T—	-	-	-	!	-	-	-	-	-	-	-	-	-	-	-	-	!!	-	-	-	!	-	!	-	-	-	-
ASUS	R—	!	-	-	-	-	!	-	!	!	-	!	-	-	!	!!*	-	-	!!*	-	-	-	-	-	-	-	-	-
ASUS	T—	-	-	-	-	-	-	-	-	!	-	!	-	-	!	!!*	-	-	!!*	-	-	-	-	!	-	-	-	-
ASUS	T—	-	-	-	-	-	-	-	-	!	-	!	-	-	!	!!*	-	-	!!*	-	-	-	-	!	-	-	-	-
D-Link	D—	!	-	-	!	!	!	-	!	!	!	-	-	-	!	-	-	!!*	-	-	-	-	-	-	!	!	-	-
D-Link	D—†	!	-	-	!	-	-	-	-	!	-	!	-	-	!	-	-	!!	-	-	-/!!*/-	!	-	!	-	-	-	-
D-Link	D—	-	-	-	-	-	!	-	!	!	-	-	-	-	!	-	-	-	-	-	!!*/!!*/-	-	-	!	-	-	!!*	-
D-Link	R—	!	!!*	-	-	-	!	-	-	!	!	!	-	!!*	!	-	-	-	-	-	!!*/!!*/-	-	-	-	-	-	-	-
H3C	N—	!	-	!	-	-	-	-	-	!	!	-	-	-	!	!!*	-	!!*	!!*	-	!!*/-/-	!	-	!	-	-	-	-
HUAWEI	A—	!	-	-	-	-	-	-	-	!	!	-	-	-	!	-	-	-	-	-	-/-/!!*	-	-	!	-	-	-	-
HUAWEI	W—	!	-	-	-	-	-	-	-	!	!	-	-	-	!	-	-	-	-	-	-/-/!!*	-	-	!	-	-	-	-
Linksys	E—	!	-	-	-	-	-	-	-	!	-	!	-	-	!	!!*	-	-	!!*	-	!!*/!!*/-	-	-	C	-	-	-	-
Linksys	E—	-	-	-	-	-	-	-	-	!	-	!	-	-	!	-	-	!!	!!*	-	!!*/!!*/-	-	-	C	-	-	-	-
Linksys	E—	!	-	-	-	-	-	-	-	!	-	!	-	-	!	-	-	-	!!*	-	-	-	-	C	-	-	-	-
Linksys	E—	-	-	-	-	-	-	-	-	!	-	!	-	-	!	-	-	!!	-	-	!!*/!!*/-	-	-	C	-	-	-	-
Linksys	E—	-	-	-	-	-	-	-	-	!	-	!	-	-	!	!!*	-	-	-	-	-/!!*/-	!	-	C	-	-	-	-
Linksys	M—	-	-	-	-	-	!	-	!	!	-	!	-	-	!	!!*	-	-	-	-	!!*/!!*/-	-	-	-	-	-	-	-
Linksys	W—	!	-	!	-	-	-	-	-	!	-	!	-	-	!	!!*	-	!!	-	-	-	-	-	-	-	-	!!*	-
Mercury	X—	!	-	!	!	-	-	-	-	-	-	-	-	-	!	-	-	-	-	-	-	!	!	!	-	-	-	-
Netcore	N—	-	-	-	!	-	-	-	-	!	!	!	-	-	!	-	-	!!*	!!*	-	-	!	-	!	-	-	-	-
Netcore	N—	!	-	-	!	!	!	!	-	!	!	!	-	-	!	-	-	!!*	!!*	-	-	-	-	-	!	-	-	-
Netcore	N—	!	-	-	!	-	!	-	-	!	!	!	-	-	!	!!*	-	!!	-	-	-	-	-	-	!	-	-	!!*
Netcore	P—	!	!!*	-	!	-	!	!	-	!	!	-	-	-	!	-	-	!!	-	-	-	-	-	!	-	-	-	-
NETGEAR	R—	-	-	-	-	-	-	-	-	!	-	!	-	-	!	!!*	-	-	-	-	-	!	-	!	-	-	-	-
NETGEAR	R—†	-	-	-	-	-	-	-	-	!	-	!	-	-	!	!!*	-	!!	!!	-	!!*/!!*/-	!	-	!	-	-	-	-
NETGEAR	R—	-	-	-	-	-	-	-	-	-	-	-	-	-	!	!!*	-	-	-	-	!!*/!!*/-	!	-	!	-	-	-	-
NETGEAR	R—	-	-	-	-	-	-	-	-	-	-	-	-	-	!	!!*	-	!!	-	-	!!*/!!*/-	!	-	!	-	-	-	-
NETGEAR	R—	-	-	-	-	-	-	-	-	-	-	-	-	-	!	!!*	-	!!	-	-	!!*/!!*/-	!	-	!	-	-	-	-
Ruijie	X—	!	-	!	!	-	-	-	-	-	-	-	-	-	!	-	-	!!*	!!*	!!*	-/!!*/-	-	-	!	-	-	!!*	-
Tenda	A—	-	-	-	-	-	!	!	-	!	-	!	-	-	!	-	-	!!	!!	!!*	!!*/!!*/-	-	-	!	-	-	-	-
Tenda	A—	!	!!*	-	-	-	!	!	-	!	-	!	-	-	!	-	-	!!	!!	!!*	!!*/!!*/-	-	-	!	-	-	-	-
Tenda	F—	-	-	-	!	-	!	!	!	-	-	-	-	-	!	!!*	!!*	-	-	-	-	-	-	-	-	-	-	-
TP-Link	A—†	!	-	-	-	-	-	-	-	!	-	!	-	-	!	-	-	-	-	-	!!*/!!*/-	!	-	!	-	-	-	-
TP-Link	T—	!	!!*	!	!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	!	!	!	-	-	-	-
TP-Link	T—	!	!!*	-	-	-	-	-	-	!	-	!	-	-	!	-	-	-	-	-	!!*/!!*/-	!	-	!	-	-	-	-
TP-Link	T—†	!	-	-	!	-	-	-	-	!	-	!	-	-	!	-	-	-	-	-	!!*/!!*/-	!	-	-	-	-	-	-
TP-Link	T—	!	-	!	!	-	-	-	-	-	-	-	-	-	!	-	-	-	-	-	-	!	!	!	-	-	-	-
Xiaomi	4—	!	-	!	!	-	-	-	-	!	!	-	-	-	!	-	-	!!*	!!*	-	!!*/!!*/!!*	!	-	-	-	-	-	-
Xiaomi	R—	!	-	!	-	-	-	-	-	!	!	-	-	-	!	-	-	-	-	-	!!*/!!*/!!*	!	-	-	-	-	-	-
ZTE	A—	!	-	-	!	-	!	-	!	!	!	!	-	-	-	-	-	-	-	-	!!*/-/-	-	-	-	!	-	-	!!
Total	40	24	5	8	15	2	12	5	6	31	12	23	0	1	37	15	1	18	14	3	19/20/4	18	3	23	4	1	3	2

Legend: "!" indicates the presence of an issue/feature that may be a prerequisite for security issues (e.g., no IPv6 NAT is not a vulnerability by itself unless IPv6 firewall is not working). "!!" indicates the presence of a potential vulnerability and "*" means we have confirmed it to be exploitable in the latest firmware version. "C" under "Open guest network" represents a captive portal. "†" marks the 4 routers purchased between Dec 2021 and Oct 2022 for preliminary (other routers were purchased in Oct 2022, Feb 2023 and May 2023).

Brands included in our selection are from both the global market [49] and, in consideration of the huge user base in China, the Chinese market [50]. We purchased the routers from Amazon US, Taobao, and JD (two largest online shopping platforms in China), which we believe are representative of what most end users are exposed to. We selected 40 home routers (for scale, 35 routers were purchased in a previous study [51]) according to their series names, regions, price range, and popularity, under a budget of $3,000 USD. Although some of them are not the latest models, they still received updates in recent years and appear to be popular on those platforms. Consumers might buy them especially since they may be cheaper and thus more attractive. Therefore, including them is essential to reflect the actual status of the router market.

B. Key Findings

1) Finding 1: IPv6 support without firewall: 37 routers support IPv6 (but only 25 can work) and 11 of them have it enabled by default. Nearly all of the routers with a functional IPv6 stack do not implement IPv6 NAT (24/25), as expected. However, 5 of them also do not implement an IPv6 firewall to block incoming connections from the internet by default. Adv_WAN can directly reach devices located in the local network using their (public) IPv6 address. This behavior is a violation of RFC 4864 [38], and dangerous in a home network environment as it exposes IoT and other devices that are not designed to be publicly reachable. Two of the 5 routers enable IPv6 by default and can simply work as plug-and-play, leaving customers of IPv6-enabled ISPs exposed out of the box.

Worse, a router implements stateful DHCPv6 and attributes full IPv6 addresses to connected devices by following simple and predictable suffixes such as “1000”, “1001”, “1002”. Fortunately, this router does not enable IPv6 by default.

The other 4 routers rely on stateless DHCPv6 to assign IPv6 addresses to devices, which means that the IPv6 addresses are generated by the devices themselves rather than the routers. Note that modern operating systems generate random interface IDs [52], [53], making it difficult for adversaries to predict the IPv6 address of a specific computer. However, we cannot guarantee that all devices, especially IoT devices, can generate an unpredictable IPv6 address for themselves.

2) Finding 2: Insecure Wi-Fi security protocols still supported by default: 13 routers still support WPA (version 1) with AES-CCMP encryption by default. Although AES-CCMP is relatively secure, it is susceptible to KRACK attacks [6], [54] that can replay and decrypt packets. Worse, 2 routers still support the outdated TKIP encryption by default, a protocol vulnerable to both replay and forgery of packets via KRACK attacks. Out of the 40 routers we purchased, 17 do not support WPA3, while the remaining 23 routers do not default to WPA3 although supported. This shows a slow adoption of new standards among router manufacturers.

Furthermore, we encountered a router that initially supports WPA/WPA2-PSK-(TKIP|CCMP) before the setup wizard finishes then transitions to only supporting WPA2-PSK-CCMP. However, it supports plug-and-play, and thus could run with less secure configuration without the setup wizard.

3) Finding 3: Insecure default settings for plug-and-play: Plug-and-play functionality allows users to start enjoying their router without accessing the management page or completing the setup wizard, which poses security risks. Out of the 12 plug-and-play routers studied, 5 lack a default Wi-Fi password, making the Wi-Fi network open and vulnerable. Furthermore, if the setup wizard is never completed, accessing the management page will redirect to this wizard that lacks any admin password in two routers, or expects simple passwords, e.g., “admin”, “password”, in 4 routers. Adv_LAN could easily take control of the router in such cases. Finally, 5 other routers ask for the Wi-Fi passphrase as an initial admin password, which Adv_LAN might already know.

4) Finding 4: WPS PIN still supported by default, sometimes with unknown PINs: Despite the known susceptibility of the WPS PIN authentication method to brute-force attacks, 31 routers appear to still support this feature as advertised by a “Configured” status. The concern is heightened as 12 of these routers do not tell users the expected PIN.

Out of these 31 “configured” routers, 23 routers can theoretically be cracked as `reaver` can successfully complete several WPS PIN attempts. To prevent brute force attacks, all of them get locked after several failed WPS PIN attempts, but the locking time varies among routers. Seven routers are only locked for 1 minute per failure and one router is only locked for 2 minutes. Other routers are locked for a long time and difficult to crack.

As brute-forcing is time-consuming, we only cracked D-Link R15 successfully. We find that the hidden WPS PIN code of this router is very simple. We purchased another router of the same model and confirmed the PIN works successfully, meaning that it is hard-coded and likely to be the same for all such routers, which is a serious vulnerability. We reported this issue to D-Link and received a confirmation. At present, this vulnerability has been assigned a CNVD ID (CNVD-2023-59339) and D-Link has fixed it by a hotfix.

5) Finding 5: Local web access vulnerable to ARP spoofing and CSRF attacks: ARP spoofing is a common method for Adv_LAN to eavesdrop on the traffic between user's devices and routers on the LAN side. We find that 37 routers cannot detect or drop the spoofed ARP packets sent to connected devices, although existing security mechanisms e.g. DAI (Dynamic ARP Inspection) [55] and IP-MAC binding can do so, which means that Adv_LAN can capture the traffic that users send to routers. Moreover, 15 of them transmit admin names and passwords with plaintext or base64 when users log in to the management page, which results in administrator privileges being exposed to Adv_LAN. Other routers encrypt the sensitive information with unknown algorithms or encode them with hash algorithms. Additionally, HTTPS can also protect sensitive information, but none of tested routers only supports HTTPS for local web access and users access them via HTTP by default.

CSRF attack is also a potential risk. Fortunately, thanks to commonly used CSRF token (e.g. session key/token/stok) and refusal of cross-origin AJAX requests, only one router is vulnerable to simple CSRF attacks.

6) Finding 6: TLS not used when needed or without certificate validation: Several critical features in routers do not adequately protect their traffic.

Firmware updates. Checking firmware version and performing firmware update all require routers to communicate with servers on the internet. We find that 7 routers rely on HTTP to check firmware version and 7 rely on HTTP to download firmware update files. Moreover, among routers that do leverage TLS, 9 of them do not validate TLS certificates properly for version checking, and 5 routers when downloading update files. Adv_NET can easily replace update files to downgrade the router or write malicious firmware into the router. Fortunately, strict firmware image verification (including version and integrity check) can alleviate this issue and Wu et al. [56] has proposed a novel approach to check this mechanism.

DDNS. DDNS service also relies on HTTP/HTTPS to transmit account information and IP addresses. In this paper, we focus on the three most common DDNS service providers: No-IP, DynDNS, and Oray. No-IP is supported by 26 routers (but only 23 can work). 16 of them rely on HTTP and 3 rely on HTTPS but do not validate TLS certificates properly. Worse, account name and passwords are only encoded with base64, which means that Adv_NET can decode or modify them very easily. DynDNS is supported by 24 routers (but only 22 can work). 17 rely on HTTP, 3 rely on HTTPS but no validation, and they also encode account information with base64. Oray is supported by 18 routers and different from other two providers. 11 of them rely on HTTP (sometimes identified as X11 by `Wireshark`), but passwords are encoded with MD5, which can prevent adversaries from eavesdropping on them and 6 of them even leverage HMAC-MD5 (the key is in the same packet) to prevent replay attacks. Another 4 routers also rely on HTTP, but 2 of them encode with base64 and 2 transmit with plaintext.

Remote management. Also, a router did not perform TLS certificate validation when being remotely managed by the companion app, i.e., Adv_NET can use a self-signed certificate to intercept traffic between the router and the server and carry out MITM attacks. Additionally, we find 8 routers implementing custom protocols without TLS to communicate with the companion app server, which may bring new security risks. We evaluate the security of these protocols in Section VI.

External storage and remote web access. These features also require the TLS protection. However, we can only find two routers employing non-self signed TLS certificates: one served a public certificate for a real domain, and leaked the key in firmware, however this problem has already been fixed in later versions of the firmware; the other one includes an untrusted certificate issued by “ZTE-ROOT-CA”, which is not better than self-signed certificates. In addition, we find one Linksys router employing an expired self-signed TLS certificate after we update its firmware. The incorrect use of TLS certificates may result in Adv_NET being able to conduct MITM attacks.

7) Finding 7: Setup wizards fail to guarantee strong passwords: Although 28 routers require users to complete the setup wizard, 18 of them do not force users to set Wi-Fi or admin passwords during the setup wizard. Similarly, all 11 routers with optional setup wizards fail to enforce the change of either passwords. After going through all setup wizards, 4 routers end up not having a default Wi-Fi password and do not force users to set it (i.e., Wi-Fi network remains open). The same applies to admin passwords for two routers. If the user is not paying attention, these routers may run without a password. Additionally, two routers employ ”admin” as the default admin password and do not require users to change it, which is also insecure. We also find that 10 routers employ the Wi-Fi password as the admin password by default and inconspicuously remind users during the setup wizard. If Adv_LAN can obtain the Wi-Fi password, he can also access the management page and control such routers easily.

10 routers require users to set both Wi-Fi and admin passwords, but the requirements for password strength are very low. All of the selected routers only require users to set a Wi-Fi password that contains more than 8 characters and don’t have any other requirements. In this case, users may prefer to set 8-digit numeric passwords [57], which are not so difficult for Adv_PHY to crack. In addition, there are 8 routers that do not even have any admin password strength requirements.

8) Finding 8: Poorly protected guest network: As guest network is an optional feature, only one router has it enabled by default, which might be why not much attention has been given to the security of this feature. However, we find 23 routers not enforcing a Wi-Fi security protocol by default, which means that the guest network is open if the user simply enables it by a click. Among the routers that enable Wi-Fi security protocols by default, we also find 4 routers support WPA and one of them still support TKIP by default.

In addition, 5 routers offer a captive portal with an open (unencrypted) Wi-Fi by default. Those routers’ strength requirements only impose a 4 character minimum limit and their login interfaces do not seem to limit password attempts.

To protect the main network, the guest network is generally isolated from the host network. However, 3 routers allow access to the management page from the guest network, unnecessarily exposing the router to (untrusted) guests. Among them, a “plug-and-play” router has “password” as default admin password, which makes it even more vulnerable.

Given these heterogeneous results, we note that a standard definition of the characteristics of “guest networks” is missing.

9) Finding 9: Reset options requiring user diligence: It is common for routers to support resetting through the management page by clicking on a button, which is also known as soft reset. We find 2 routers retaining sensitive information after reset. One retains logs, fortunately they do not contain sensitive information. The other one is still bound with the companion app after reset and the app can remotely manage the router. This may undermine user privacy when second-hand routers are traded.

Additionally, we find 7 routers provide different reset options for users to keep some data, but enable them by default. If the user is not aware, some sensitive information may be retained. For example, 2 routers have “Preserve network configuration when restoring factory settings” enabled by default and will retain Wi-Fi passwords after reset.

```
1   void *sub_58ED4()
2   {
3     // ...
4     if ( acosNvramConfig_match("soap_fw_chk_http", "1") ==
          1 )
5       v2 = "http";
6     else
7       v2 = "https";
8     v3 = (const char *)acosNvramConfig_get("
          ver_check_https_svr1");
9     v4 = (const char *)get_firmware_update_folder(
          firmware_update_file_not_found);
10    sprintf(
11      v6,
12      "rm -f %s %s %s ;curl -k --stderr %s -o %s %s://%s/%s
          /%s/%s &",
13      "/tmp/image.chk",
14      "/tmp/curl.log",
15      "/tmp/curl_result",
16      "/tmp/curl.log",
17      "/tmp/image.chk",
18      v2,
19      v3,
20      v5,
21      v4,
22      byte_318F64);
23    sub_54AF8(3, "[upnp_sa] wget_SendGetImageCmd:%s\n", v6)
          ;
24    result = (void *)system(v6);
25    // ...
26  }
```

Listing 1. An example of calling curl with -k (simplified).

V. TLS Certificate Validation Vulnerability Detection

According to our analysis results, incorrect implementation of TLS is the most common security risk for home routers. In this section, we figured out the causes of TLS certificate validation vulnerabilities based on reverse engineering, and proposed a heuristic method to assist manual analysis in detecting these vulnerabilities without real routers.

A. Cause Analysis

Among our routers with exploitable TLS-related vulnerabilities, 22 router firmware images are available. We analyzed part of them to figure out the causes of these vulnerabilities based on reverse engineering.

In addition to vulnerabilities caused by default use of HTTP, the lack of TLS certificate validation is the main reason for vulnerability to MITM attacks, which we name TLS certificate validation vulnerability. Surprisingly, these vulnerabilities in home routers are not caused by the misuse of underlying TLS libraries, such as OpenSSL and GnuTLS, but rather by incorrect configuration when calling third-party applications, such as `curl`, and `wget`. We speculate that this is because developers of home router firmware tend to directly call existing applications to transmit information when implementing firmware updates and DDNS services.

We summarized the causes we found in these firmware and classified them into four categories:

- Call `curl` with `-k` or `--insecure` (e.g., line 12 in Listing 1).
- Call `curl_easy_setopt` to set CURLOPT_SSL_VERIFYHOST (81) or CURLOPT_SSL_VERIFYPEER (64) to 0 before calling `curl_easy_perform` (e.g., line 25 and 26 in Listing 2).
- Call wget-like programs (including `wget`, `wgets`, `uclient-fetch`, etc) with `--no-check-certificate` (e.g., line 13 in Listing 3).

```
1   int __fastcall sub_41F1C8(int a1, int a2, ...)
2   {
3     // ...
4     time(v164);
5     v134 = (_DWORD *)localtime(v164);
6     fprintf(
7       v133,
8       "%d/%02d/%02d %02d:%02d:%02d [%s.%d]DownloadUrl:[%s]\
          n",
9       v134[5] + 1900,
10      v134[4] + 1,
11      v134[3],
12      v134[2],
13      v134[1],
14      *v134,
15      "ONLINEUPDATE_REQ_DownloadVer",
16      405,
17      v116);
18    v135 = curl_easy_init();
19    v136 = v135;
20    if ( v135 )
21    {
22      curl_easy_setopt(v135, 113, 1);
23      curl_easy_setopt(v136, 10002, v116);
24      curl_easy_setopt(v136, 99, 1);
25      curl_easy_setopt(v136, 64, 0);
26      curl_easy_setopt(v136, 81, 0);
27      curl_easy_setopt(v136, 98, 1024);
28      v160 = 0;
29      v159 = (int)"/var/ftproot/version/AC_APP";
30      curl_easy_setopt(v136, 20011, sub_402E8C);
31      curl_easy_setopt(v136, 10001, v155);
32      curl_easy_setopt(v136, 13, 180);
33      v139 = curl_easy_perform(v136);
34      curl_easy_cleanup(v136);
35      // ...
36  }
```

Listing 2. An example of calling curl_easy_setopt incorrectly (simplified).

```
1   # ...
2   if [ "$update_check" == "1" ]; then
3     if [ "$netcon" == "Connected" ]; then
4       # Get variables for uclient request
5         mac_address="$(uci -q get linksys.@hardware[0].
              hw_mac_addr)"
6         hardware_version="$(uci -q get linksys.@hardware
              [0].hw_revision)"
7         model_number="$(uci -q get linksys.@hardware[0].
              modelNumber)"
8         serial_number="$(uci -q get linksys.@hardware[0].
              serial_number)"
9         ip_address="$(ubus call network.interface.wan
              status 2> /dev/null | jsonfilter -qe "@['ipv4
              -address'][-1].address")"
10        installed_version="$(uci -q get linksys.@firmware
              [0].version)"
11
12      # Send request to server
13        uclient-fetch --no-check-certificate -T 5 -O "/etc
              /fwupdate" \
14        "https://update1.linksys.com/api/v2/fw/update?
              mac_address=${mac_address}&hardware_version=$
              {hardware_version}&model_number=${
              model_number}&installed_version=${
              installed_version}&ip_address=${ip_address}&
              serial_number=${serial_number}" &> /dev/null
15  # ...
```

Listing 3. An example of calling wget-like programs with –no-check-certificate (simplified).

- Call the `wget` linked to `busybox` (old versions of `busybox` do not support certificate validation [58]).

These misconfigurations are very common in the code for implementing firmware update and DDNS service, which results in TLS certificates not being validated when routers check firmware version, download firmware images or transmit DDNS account information via the internet.

B. Heuristic Detection Method

According to the causes, we proposed a heuristic method to detect the sensitive code that may cause TLS certificate validation vulnerabilities in router firmware. The details are as follows:

1) “curl -k/--insecure”: To find the code calling `curl` with `-k` or `--insecure`, we designed a string matching method based on regular expressions. The final version of regular expressions are “[ˆa-zA-Z]curl[\x00].*-k[\x00]” and “[ˆa-zA-Z]curl[\x00].*--insecure[\x00]”.

In these regular expressions, “[ˆa-zA-Z]” is to ignore other strings ending in “curl”. “[\x00]” means that the one following “curl” and “-k/--insecure” must be a space or 0x00 because `curl` is normally called by a complete string (as shown in Listing 1) and a space follows in this string, or by a string composed of multiple words pieced together and 0x00 follows in the binary. “.*” means that there may be other parameters between “curl” and “-k/--insecure”, but which may bring false positives sometimes. Finally, we exclude `curl` program itself because it is a false positive due to the help information containing “-k” or “--insecure”.

2) “curl_easy_setopt”: `curl_easy_setopt` is a commonly used library function to configure `curl` before calling `curl_easy_perform` to send requests. The second parameter of `curl_easy_setopt` is the curl option ID. CURLOPT_SSL_VERIFYHOST (81) decides whether verifying the certificate’s name against host, while CURLOPT_SSL_VERIFYPEER (64) decides whether verifying the peer’s SSL certificate. The third parameter is the value and 0 means not verifying.

To find this type of misconfigurations, we first search for ELF binaries calling `curl_easy_setopt`. Then, we load them with IDA Pro and write an IDA script to check the parameters of `curl_easy_setopt`. We record the addresses of code setting (81,64) to (0,0), (0,1) or (1,0). We also record cases where the parameters are variables. Additionally, the selection of parameters sometimes is based on conditional branches, i.e. setting the option to 1 or 0 according to whether the SSL certificate exists or not. Therefore, if an option is set to both 0 and 1 in one function, we record it. After all, we manually check whether these special cases can be exploited.

3) “wget --no-check-certificate”: Different from “curl -k/--insecure”, “--no-check-certificate” is a characteristic parameter and commonly used by wget-like programs. Therefore, we directly search for binaries and scripts containing “--no-check-certificate”. We also exclude programs named “wget”, “wgets” or “uclient-fetch”.

4) “wget linked to busybox”: In some firmware, `wget` is linked to `busybox` rather than implemented as a independent program because `busybox` also supports this function. However, the TLS library in `busybox` does not support certificate validation. Therefore, we first check the size of `wget` in the firmware. If the size is smaller than 1KB, we will check whether it is linked to `busybox`. If yes, we leverage a regular expression to find all binaries and scripts calling `wget`. The final version is “[ˆa-zA-Z]wget[\x00].*(-O[\x00]|--output-document=)”. “(-O[\x00]|--output-document=)” is to reduce false positives because “-O” or “--output-document” is a necessary parameter for `wget` to set the path to save the content of the response.

Our heuristic method can narrow down the search scope of TLS certificate validation vulnerabilities. However, manual reverse engineering is still necessary to check which feature this code implements and whether it is exploitable, because this work is very complex and difficult to automate.

C. Evaluation

1) Implementation: We implemented our method as automated scripts with about 600 lines of Python code, including an script for IDA Pro 7.5. To improve efficiency, we employed multi-process analysis. These scripts can automatically analyze the filesystem of the target firmware and output sensitive binaries and scripts that may exist TLS certificate validation vulnerabilities. The source code will be available at https://github.com/YjjNJUPT/extended_version_of_default_settings.

2) Dataset: Among our 40 routers, 29 firmware images are available and their filesystems can be extracted by `binwalk` [59]. Because we have confirmed all exploitable vulnerabilities in the latest version, we select the latest version of firmware images to form the dataset. Additionally, the vulnerability in a NETGEAR router has been fixed in the latest version, so we also add the old version into our dataset for comparison. Finally, our dataset consists of 30 router firmware images from 10 brands and corresponding analysis results of real routers (introduced in Section IV).

3) Results: For each sample in our dataset, we run our scripts to find sensitive binaries and scripts automatically. This experiment was conducted in a Windows laptop with a 6-core CPU and 16 GB RAM. Then, we manually check which feature calls TLS improperly and whether it is exploitable. Finally, we compare the results with our analysis results of real routers based on our framework to verify the effectiveness of our heuristic method. The details are shown in Table III.

In total, our scripts produce 209 alerts in 30 firmware images. 43 of them are related to firmware update or DDNS service and exploitable for adversaries to threaten router security. Sometimes, multiple alerts are related to the same feature. 163 of them are related to other features, such as network speed test, third-party game accelerator, and opkg download. These features also implement TLS improperly and vulnerable to MITM attacks, but are difficult to be exploited to do something. Additionally, not only `curl`, but other commands (such as `unlzma`, `df`, and `openssl`) also support “-k” parameter. These 3 false positives are caused by files that both contain “curl” and call these commands with “-k”.

17 vulnerabilities found by our method are consistent with the analysis results of real routers, which are True Positives (TP). 15 vulnerabilities found in real routers cannot be found by our method, which are False Negatives (FN). The cause of these false negatives is that our method focuses on misconfigurations of TLS certificate validation, but these vulnerable features are only based on HTTP. 11 vulnerabilities found by our method were not found in real routers, which are False Positives (FP). 10 of them are related to firmware update and these routers cannot be updated successfully because there is something wrong or the latest version is the only one available version, so we cannot test them, let alone find these vulnerabilities in real routers. Other one is related to DDNS and it is also because DDNS service of this router cannot work properly, resulting in inability to test. Additionally, our

TABLE III
EVALUATION RESULTS

Brand	Model	Number of different types of alerts (False positive/Not exploitable/Exploitable)				Comparison results with real routers (FP/FN/TP)		
		curl -k/--insecure	curl_easy_setopt	--no-check-certificate	linked to busybox	Check version	Download fw	DDNS
360	T—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	-
ASUS	R—	0 / 1 / 0	0 / 6 / 0	0 / 9 / 4	0 / 0 / 0	-	TP	Google
ASUS	T—	0 / 1 / 0	0 / 5 / 0	0 / 6 / 4	0 / 0 / 0	-	TP	-
ASUS	T—	0 / 1 / 0	0 / 5 / 0	0 / 6 / 4	0 / 0 / 0	-	TP	-
D-Link	D—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 2 / 2	FP*	FP*	FN†
H3C	N—	0 / 1 / 0	0 / 1 / 1	0 / 0 / 0	0 / 0 / 0	TP	TP	FN†
Linksys	E—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	FN†	FN†
Linksys	E—	1 / 3 / 4	0 / 0 / 0	0 / 2 / 0	0 / 3 / 0	FP*	TP†	TP
Linksys	E—	0 / 0 / 0	0 / 0 / 0	0 / 2 / 0	0 / 4 / 0	-	FN†	-
Linksys	E—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 1 / 2	FP*	FP*	FN†
Linksys	E—	0 / 0 / 1	0 / 1 / 1	0 / 0 / 0	0 / 2 / 1	-	FP*	TP†
Linksys	M—	1 / 4 / 1	0 / 1 / 1	0 / 0 / 0	0 / 1 / 1	-	FP*	TP
Linksys	W—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 2	0 / 0 / 0	FP*	-	FP*
Mercury	X—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	-
Netcore	N—	0 / 0 / 0	0 / 0 / 0	0 / 1 / 1	0 / 2 / 1	TP	TP†	-
Netcore	N—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	-
Netcore	P—	0 / 0 / 0	0 / 0 / 0	0 / 1 / 0	0 / 1 / 1	FP*	-	-
NETGEAR	R—	0 / 4 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	FN†
NETGEAR	R—	0 / 0 / 4	0 / 4 / 0	0 / 1 / 1	0 / 0 / 0	TP	TP	FN†
NETGEAR	R—	0 / 3 / 0	0 / 10 / 0	0 / 0 / 0	0 / 2 / 0	-	-	FN†
NETGEAR	R—	0 / 3 / 0	0 / 13 / 0	0 / 0 / 0	0 / 0 / 0	-	-	FN†
NETGEAR	R—	0 / 4 / 0	0 / 11 / 0	0 / 0 / 0	0 / 0 / 0	-	-	FN†
NETGEAR	R—	1 / 3 / 0	0 / 9 / 0	0 / 0 / 0	0 / 3 / 1	-	-	TP†
TP-Link	A—	0 / 1 / 0	0 / 2 / 0	0 / 0 / 0	0 / 0 / 0	-	-	FN†
TP-Link	T—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	-
TP-Link	T—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	FN†
TP-Link	T—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	FN†
TP-Link	T—	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	0 / 0 / 0	-	-	-
Xiaomi	4—	0 / 1 / 0	0 / 5 / 0	0 / 0 / 0	0 / 4 / 3	FN†	TP†	TP
Xiaomi	R—	0 / 0 / 0	0 / 2 / 0	0 / 1 / 1	0 / 4 / 1	-	FP*	TP
Total		3 / 30 / 10	0 / 75 / 3	0 / 29 / 17	0 / 29 / 13	5 / 1 / 3	5 / 2 / 8	1 / 12 / 6

Legend: "†" marks the routers that implement features based on HTTP. "*" marks the routers whose firmware update or DDNS service cannot be tested because only one latest version can be found or the feature cannot work.

method found a router implementing GoogleDNS improperly. However, Google no longer offers new domain registrations now [60], so we cannot check it.

In conclusion, our heuristic method has been proven to significantly narrow down the search scope of TLS certificate validation vulnerabilities and improve the efficiency of manual analysis without real routers.

VI. Custom Remote Management Protocols

Although TLS is the most commonly used security protocol in home routers, we found a few routers employing custom protocols without TLS to implement app-based remote management, which may bring new security risks. In this section, we propose and conduct a procedure to evaluate the security of custom remote management protocols.

A. *Overview of Remote Management*

Remote management based on companion apps is a popular new router feature, which allows users to manage the router (e.g., changing Wi-Fi passwords, enabling guest network) over the Internet by the companion app. The use of this feature can be divided into two stages: binding and command execution.

Binding. As shown in Figure 6a, to enable this feature, the user needs to connect a smartphone to the Wi-Fi of the router and bind his companion app account with the router. The binding relationship will be recorded by the manufacturer's cloud server. Among the routers we evaluated in Section IV, all of them implement TLS properly to protect the traffic between the router and the server in this stage.

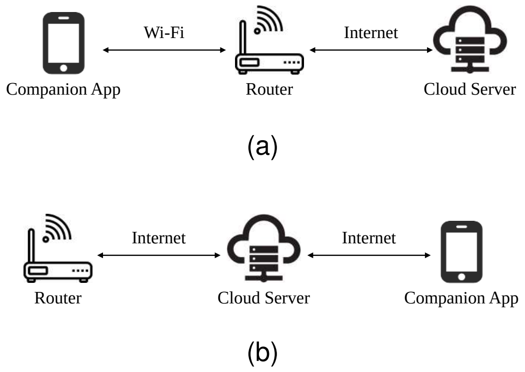


Fig. 6. Architecture of app-based remote management. (a) Binding stage. (b) Command execution stage.

Command execution. If DDNS is not configured, it is not feasible for routers to communicate with the companion apps directly over the Internet. Therefore, as shown in Figure 6b, when the user manages the router remotely, the companion app sends commands to the cloud server, and then the server forwards commands to the router. According to our analysis results, 8 of tested routers leverage custom protocols without TLS rather than HTTPS to transmit commands and sensitive information in this stage.

To sum up, besides TLS whose security has been discussed in previous sections, we found a few routers employing custom protocols without TLS to implement app-based remote management, which may bring new security risks.

B. Analysis Methodology

To evaluate the security of custom remote management protocols, we propose a procedure based on our analysis experience. This procedure is divided into two stages: protocol reverse engineering and security evaluation.

1) Protocol Reverse Engineering: To evaluate the security of custom protocols, we first need to figure out how they encrypt or encode the content of packets.

In the analysis environment introduced in Section III-B, we capture the traffic between routers and cloud servers when doing some tasks that require sending sensitive information like changing Wi-Fi password on the companion app. Sensitive strings, such as human-readable strings and domain names of cloud servers, can be collected in these traffic.

Then, we search for these sensitive strings in the firmware filesystem to find the code responsible for remote management. Based on reverse engineering, we can figure out which protocols and algorithms are used to generate remote management packets and locate the encryption library functions being called. In this step, the log strings and function names are important clues.

Finally, we trace the parameters of encryption library functions to figure out the source of secret keys. If no common encryption algorithm is called, it indicates that the manufacturer may implement a custom algorithm to protect sensitive information. Try to figure out whether a key is generated and used by this algorithm.

We also considered leveraging dynamic analysis to reduce the difficulty of protocol reverse engineering. However, debugging real routers cannot be commonly used. Firmware emulation can be another solution, but it is not yet feasible to build communication between the companion apps and the firmware in state-of-the-art emulators. Therefore, static analysis is still the most common method.

2) Security Evaluation: The security of custom remote management protocols relies on existing encryption algorithms, so we focus on checking whether these routers implement algorithms correctly. According to the rules in [61], we list the following common cryptographic misuses:

- Symmetric ciphers (e.g. AES) employ the same secret keys for both encryption and decryption. Hard-coded secret keys can be extracted from firmware images and exposed to adversaries. Using the ECB mode or the CBC mode with static initialization vectors (IVs) enables adversaries to perform chosen plaintext attacks (CPA).
- Asymmetric ciphers (e.g. RSA) employ key pairs consisting of public keys and private keys. Storing private key files in firmware images allows adversaries to decrypt received packets easily. Key sizes of less than 2048 bits in RSA are considered insecure. OAEP (Optimal Asymmetric Encryption Padding) [62] is also necessary for RSA because it can help eliminate the impact of predictable common text and enhances security.
- To generate random secret keys, pseudorandom number generator (PRNG) is needed. Using weak PRNG (e.g. *rand*) or predictable PRNG seeds (e.g. current time) causes the pseudorandom number to be predictable.

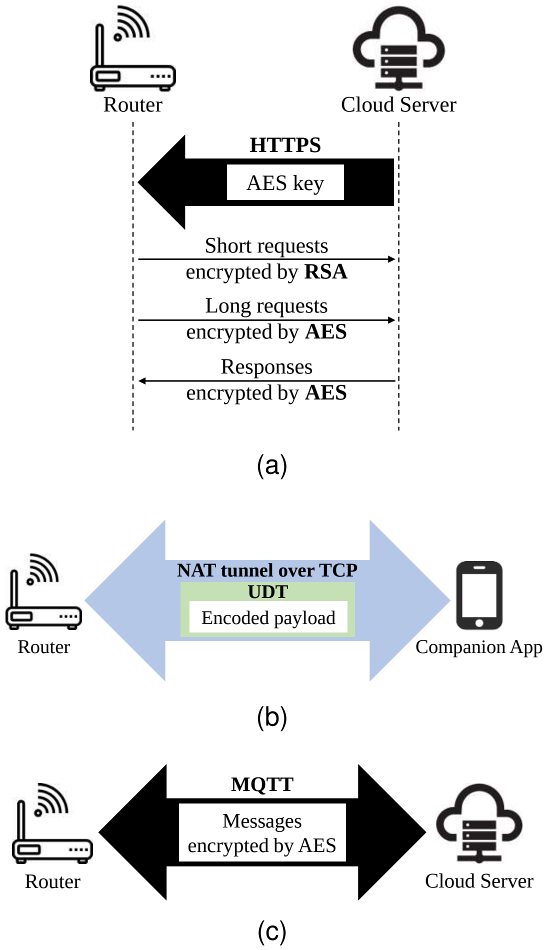


Fig. 7. Three types of custom protocols for app-based remote management.

Additionally, if we cannot find any common encryption algorithms or secret keys in a firmware image, we assume it employs a custom form of encoding or encryption, which is likely to introduce weaknesses.

C. Analysis Results

In Finding IV-B6, we found 8 routers employing custom protocols to implement app-based remote management. We evaluated the security of these protocols following our procedure. Among them, only 2 routers transmit sensitive information with plaintext. Other 6 routers protect the information with various methods, which can be divided into the following 3 types (as shown in Figure 7):

(a) 2 routers encrypt "short requests" (defined by the manufacturer) to be sent with RSA (without OAEP) and encrypt "long requests" to be sent with AES-ECB. Then, they decrypt received packets with AES-ECB. One of them leverages a 2048-bit RSA key but the other one uses a 1024-bit key, considered weak. The AES key is generated by the server and sent to the router with HTTPS when the router first establishes a connection with the server. Additionally, we found one of them also employing the custom protocol when updating firmware. In this case, it encrypts all of packets to be sent with RSA and decrypts received packets with AES. The AES key is generated by the router based on multiple factors (e.g., sysinfo, current time, and process ID) and sent to the server with RSA.

(b) 3 routers communicate with the apps remotely based on an NAT tunnel over TCP (with the PJSUA library). Their content is transmitted in the format of UDT (UDP-based Data Transfer) protocol and encoded with a custom algorithm. We attempted to check the security of this algorithm, but no common encryption algorithm is called by it and we could not easily establish whether a secret key is generated and used. More reverse-engineering effort would be needed in such a case.

(c) One router leverages MQTT (Message Queuing Telemetry Transport) protocol and encrypts the messages with AES-CBC. The AES key and IV are generated randomly based on the current time, which makes the pseudorandom number predictable. Additionally, this router employs a different strategy to transmit device data. It leverages RSA-OAEP with a 2048-bit key to encrypt the AES key and then encrypt other content with AES-CBC. We also found a function responsible for decrypting device data packets with an RSA private key, but it is not actually called.

As we know, the main advantage of AES is its fast encryption and decryption speed and UDT protocol is also an efficient data transfer protocol, so we speculate that these routers choose custom protocols instead of TLS to reduce the overhead and improve the efficiency of remote communication. However, the incorrect implementation of custom protocols brings new security risks to these routers. As shown in Table IV, we have found several cryptographic misuses in these 3 types of custom protocols, such as using the ECB mode of AES, using RSA keys with insufficient size, using RSA without OAEP, and using weak PRNG (PseudoRandom Number Generator) and predictable PRNG seeds. Once Adv_NET defeats custom protocols based by exploiting these weaknesses, they could eavesdrop on users' sensitive information (e.g., Wi-Fi and admin passwords) and even control the router remotely (e.g., changing DNS settings).

VII. Discussion and Future Work

We have found several overlooked security issues caused by incorrect default settings of home routers (as shown in Section IV), but our analysis framework is based on manual efforts and cannot be leveraged to conduct automated large-scale evaluation. If we want to know whether these issues are widely present in various routers, it is necessary to figure out the causes of them and propose firmware-based detection methods, as we did in Section V. In this section, we discuss the additional efforts we have made in this domain and possible research directions based on our findings to inspire other researchers interested in router security.

TABLE IV
Security Issues in Custom Protocols

Type	No. of samples	Custom algorithm	RSA		AES		
			Without OAEP	Key size	ECB mode	Weak PRNG	Predictable PRNG seeds
(a)	1	-	!	2048 bits	!	-	-
	2	-	!	1024 bits	!	-	-
(b)	3, 4, 5	!	-	-	-	-	-
(c)	6	-	-	-	-	!	!

Legend: "!" indicates the presence of a cryptographic misuse or potential risk.

IPv6 firewall. According to Finding IV-B1, firewall is critical for the security of IPv6. Normally, the implementation of IPv6 firewall relies on *ip6tables* commands. Developers should configure IPv6 firewall rules by calling *ip6tables* to filter and forward IPv6 traffic correctly. For example, an IPv6 firewall vulnerability has been fixed after our report. We compared the new and old versions of firmware, and found that the developer fixed this vulnerability by improving the *ip6tables* commands in function *oal_initIp6FirewallObj*. Therefore, checking whether *ip6tables* is correctly called may be a method to detect vulnerabilities in the default configuration of IPv6 firewalls in the future.

Hard-coded WPS PIN. As shown in Finding IV-B4, among the routers with hidden WPS PIN, we only cracked D-Link R15 successfully because its PIN is hard-coded and very simple, but it is difficult to know the detailed cause of this vulnerability because the firmware image is encrypted and cannot be analyzed. Other routers get locked after several failed WPS PIN attempts, so we have to check whether their WPS PIN is hard-coded by reverse engineering. However, only Netcore routers' firmware is available and we found that the WPS PIN is generated randomly. Therefore, we can only propose possible detection methods theoretically. Traditional sensitive string detection based on regular expressions is a choice because WPS PIN is an 8-digit string. Additionally, WPS-related strings, such as function names like *pm_wlan_set_wps_pin*, and firmware emulation can also assist in locating and extracting PIN codes.

Communication security. We have proposed a method to detect TLS certificate validation vulnerability in Section V, and Liu et al. [63] also explored how to detect the misuse of SSL/TLS libraries. However, the use of HTTP without TLS is also a noteworthy issue. Checking URL strings in firmware images and capturing traffic based on emulators are possible methods to detect which features transmit information without TLS. Additionally, as we discussed in Section VI, several routers leverage custom protocols without TLS to implement remote management. Therefore, besides TLS, evaluating the security of custom protocols based on cryptographic primitives identification [64] and cryptographic misuse detection [61], [65] is also an interesting research direction.

Local web security. Although firmware-based emulation may miss information stored in flash (discussed in the conference version [14]), it will not affect the basic logic of web services, which means that local web security can be tested based on emulation. For example, according to Finding IV-B5, most of routers are vulnerable to ARP spoofing, so it is necessary to check whether admin names and passwords are transmitted with plaintext or base64. Firmware emulators, such as FirmAE [9], can be leveraged to emulate the web service and capture the traffic between the host and the web server to check this issue. Therefore, emulation-based large-scale router web security evaluation can be future work.

VIII. Related Work

In this section, we revisit several previous efforts similar or comparable to our work.

IoT security analysis. Wang et al. [66] and Zhao et al. [67] conducted large-scale security analyses of IoT devices on the internet. They leverage search engines such as Shodan [17] and Zoomeye [18] or custom search tools [68] to test active IoT devices on the internet, which means only devices with exposed services (open ports) to the internet were covered in the analysis. They focus on the distribution information of device firmware versions and known vulnerabilities on the internet without covering other types of attackers discussed in Section III-A.

Taking a step further, there are also projects/studies focusing on IoT devices in home networks. For instance, Kumar et al. [32] analyzed the user's home network information collected by Avast's tool called Wi-Fi Inspector (where users could upload their scan results for insecure IoT devices), and found that many IoT devices employed weak passwords on FTP and Telnet, and default admin passwords that were left unchanged by users. Alrawi et al. [69] systematized the literature for home-based IoT security and, similar to our work, evaluated 45 IoT devices and proposed mitigation recommendations based on an abstract model that segments IoT deployments into components, including the IoT device, the companion mobile app, the cloud endpoints, and the associated communication channels.

Although home routers are also a type of IoT devices, the security implications of their default settings is tightly coupled with the users' perception and usage habits and how they use the routers — involving sociotechnical factors. Therefore, a tailored study like ours is necessary. Nonetheless, these aforementioned studies can still be good references for evaluating home router security.

Router security analysis. Visoottiviseth et al. [29] proposed an emulation-based firmware analysis tool that can perform both static and dynamic large-scale analyses for router firmware. It includes several open-source automated tools to test the security of passwords, SSL, web application and firmware update to find vulnerabilities in the router firmware. However, emulation-based analysis is limited to only router models with publicly available firmware and subject to potential low fidelity. Along this direction, similar to our work, Jeitner et al. [51] and Niemietz et al. [70] chose to evaluate real-world home routers, but they only focus on the DNS service or web interface of routers. Currently, there is no comprehensive security assessment work for the default settings of various features of home routers.

TLS security of IoT devices. Several studies [69], [71] have found that many IoT devices configure TLS improperly when communicating via the internet. Alrawi et al. [69] found that self-signed or expired certificates, domain name mismatch, and support for vulnerable versions of TLS/SSL protocol are common TLS configuration issues. Paracha et al. [71] found that 11 devices in their study are vulnerable to TLS interception attacks because they either bypass certificate validation altogether or do not validate host names. Their findings are similar to ours, but they evaluated all types of IoT devices and ignored unique features of home routers (e.g. DDNS service), which makes it difficult for them to fully demonstrate the specific considerations for TLS issues in home routers. Moreover, they only raised these issues but did not attempt to figure out the causes of them. Liu et al. [63] explored how to detect these issues, but their angle is different from ours. They found that the reason why the certificate is not validated correctly is that the developer fails to invoke necessary APIs (e.g., *SSL_get_verify_result*), passes incorrect arguments to some SSL/TLS library APIs (e.g., *SSL_VERIFY_NONE* passed to *SSL_CTX_set_verify*), or does not correctly verify the execution result of certain SSL/TLS APIs (e.g., *SSL_get_verify_result*). However, according to our analysis, TLS-related vulnerabilities in home routers are often caused by the misuse of third-party applications, such as `curl` and `wget`, rather than underlying SSL/TLS APIs, which led us to propose completely different methods to detect TLS issues at different points.

IX. Conclusion

In this paper, we conducted a comprehensive user survey to understand the configuration habits and attitudes of home router users and brought attention to the various security-related default settings of home routers. We designed a comprehensive router default settings security analysis framework and leveraged it to analyze 40 home routers from the global and Chinese markets. To our surprise, although our analysis methodology is quite straightforward (we merely verify whether the settings and basic functions are configured properly), we found numerous security issues, resulting in up to 89 exploitable security issues. Among them, incorrect implementation of TLS is the most common, and we proposed a heuristic method to narrow down the search scope of TLS certificate validation vulnerabilities and improve the efficiency of manual analysis without real routers. Not only that, we also evaluated the security of custom protocols without TLS. Finally, to inspire researchers interested in our work, we proposed several recommendations for extending the analysis framework and discussed our ideas about automatically detecting security issues based on our findings. We hope our analysis can shed some light on the user-centric security research of home routers in the community.

Acknowledgments

This work is supported by the National Key Research and Development Program of China with 2019YFB2101704.

References

[1] A. Petrosyan. (2024) Number of Internet and Social Media Users Worldwide as of April 2024. [Online]. Available: https://www.statista.com/statistics/617136/digital-population-worldwide/

[2] Manos Antonakakis, Tim April, Michael Bailey, Matthew Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou., "Understanding the Mirai Botnet," in *USENIX Security Symposium (USENIX Security)*, 2017, pp. 1093–1110.

[3] Hilt, Stephen and Merces, Fernando. (2021) VPNFilter Two Years Later: Routers Still Compromised. [Online]. Available: https://www.trendmicro.com/en_us/research/21/a/vpnfilter-two-years-later-routers-still-compromised-.html

[4] Baran, Guru. (2019) New Mozi P2P Botnet Attacks Netgear, GPON, D-Link and Huawei Routers Using Weak Passwords and Some Known Exploits. [Online]. Available: https://gbhackers.com/new-mozi-botnet/

[5] S. Viehböck. (2011) Brute Forcing Wi-Fi Protected Setup. [Online]. Available: https://www.cs.cmu.edu/~rdriley/330/papers/viehboeck_wps.pdf

[6] M. Vanhoef and F. Piessens, "Release The Kraken: New Kracks in the 802.11 Standard," in *ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*, 2018, pp. 299–314.

[7] Y. Shoshitaishvili, R. Wang, C. Hauser, C. Kruegel, and G. Vigna, "Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware," in *Network and Distributed System Security Symposium (NDSS'15)*, 2015, pp. 1–15.

[8] M. Elsabagh, R. Johnson, A. Stavrou, C. Zuo, Q. Zhao, and Z. Lin, "FIRMSCOPE: Automatic Uncovering of Privilege-Escalation Vulnerabilities in Pre-Installed Apps in Android Firmware," in *USENIX Security Symposium (USENIX Security)*, 2020, pp. 2379–2396.

[9] M. Kim, D. Kim, E. Kim, S. Kim, Y. Jang, and Y. Kim, "FirmAE: Towards Large-Scale Emulation of IoT Firmware for Dynamic Analysis," in *Annual Computer Security Applications Conference (ACSAC'20)*, 2020, pp. 733–745.

[10] Y. Zhang, W. Huo, K. Jian, J. Shi, L. Liu, Y. Zou, C. Zhang, and B. Liu, "SRFuzzer: An Automatic Fuzzing Framework for Physical SOHO Router Devices to Discover Multi-Type Vulnerabilities," in *Annual Computer Security Applications Conference (ACSAC'19)*, 2019, pp. 544–556.

[11] S. R. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *Annual International Workshop on Selected Areas in Cryptography (SAC'1)*, 2001, pp. 1–24.

[12] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker, "Security flaws in 802.11 data link protocols," *Communications of the ACM (CACM'3)*, vol. 46, no. 5, pp. 35–39, 2003.

[13] N. Nthala and I. Flechais, "Rethinking Home Network Security," in *European Workshop on Usable Security (EuroUSEC'18)*, London, England, 2018, pp. 1–11.

[14] J. Ye, X. D. C. De Carnavalet, L. Zhao, M. Zhang, L. Wu, and W. Zhang, "Exposed by Default: A Security Analysis of Home Router Default Settings," in *ACM Asia Conference on Computer and Communications Security (ASIA CCS'24)*, 2024, p. 63–79.

[15] (2024) Wenjuanxing Homepage. [Online]. Available: https://www.wjx.cn/

[16] (2024) SurveyMonkey Homepage. [Online]. Available: https://www.surveymonkey.com/

[17] (2024) Shodan Homepage. [Online]. Available: https://www.shodan.io/

[18] (2024) Zoomeye Homepage. [Online]. Available: https://zoomeye.org/

[19] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *ACM conference on Wireless network security (WiSec'09)*, 2009, pp. 79–86.

[20] D. Dolev and A. Yao, "On the Security of Public Key Protocols," *IEEE Transactions on Information Theory (TIT)*, vol. 29, no. 2, pp. 198–208, 1983.

[21] J. O'Flaherty, "Hierarchy – What Do You Want People to See? Where Do You Want Them to Go?" 2012. [Online]. Available: https://www.datadial.net/blog/hierarchy-what-do-you-want-people-to-see-where-do-you-want-them-to-go/

[22] IEEE, "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)*, pp. 1–4379, 2021.

[23] (2020) Wi-Fi Protected Setup Specification v2.0.8. [Online]. Available: https://www.wi-fi.org/downloads-registered-guest/Wi-Fi_Protected_Setup_Specification_v2.0.8.pdf

[24] t6x. (2015) reaver-wps-fork-t6x. [Online]. Available: https://github.com/t6x/reaver-wps-fork-t6x

[25] S. Whalen, S. Engle, and D. Romeo. (2001) An Introduction to ARP Spoofing. [Online]. Available: https://api.semanticscholar.org/CorpusID:59638215

[26] D. Song. (2017) arpspoof in Dsniff. [Online]. Available: https://www.kali.org/tools/dsniff/#arpspoof

[27] KirstenS. (2024) Cross Site Request Forgery (CSRF) | OWASP Foundation. [Online]. Available: https://owasp.org/www-community/attacks/csrf

[28] A. Cortesi, M. Hils, T. Kriechbaumer, and contributors. (2010) Mitmproxy: A Free and Open Source Interactive HTTPS Proxy. [Online]. Available: https://mitmproxy.org/

[29] V. Visoottiviseth, P. Jutadhammakorn, N. Pongchanchai, and P. Kosolyudhthasarn, "Firmaster: Analysis Tool for Home Router Firmware," in *International Joint Conference on Computer Science and Software Engineering (JCSSE'18)*, 2018, pp. 1–6.

[30] M. Bettayeb, Q. Nasir, and M. A. Talib, "Firmware Update Attacks and Security for IoT Devices: Survey," in *Annual International Conference on Arab Women in Computing (ArabWIC'19)*, 2019, pp. 1–6.

[31] P. K. Manadhata and J. M. Wing, "An Attack Surface Metric," *IEEE Transactions on Software Engineering (TSE'10)*, vol. 37, no. 3, pp. 371–386, 2010.

[32] D. Kumar, K. Shen, B. Case, D. Garg, G. Alperovich, D. Kuznetsov, R. Gupta, and Z. Durumeric, "All Things Considered: An Analysis of IoT Devices on Home Networks," in *USENIX Security Symposium (USENIX Security)*, 2019, pp. 1169–1185.

[33] (2024) Nmap: the Network Mapper - Free Security Scanner. [Online]. Available: https://nmap.org/

[34] M. Horowitz. (2015) Router Security. [Online]. Available: https://www.routersecurity.org/checklist.php

[35] P. Szewczyk and R. Macdonald, "Broadband Router Security: History, Challenges and Future Implications," *Journal of Digital Forensics, Security and Law (JDFSL'17)*, vol. 12, no. 4, pp. 55–74, 2017.

[36] D. Murphy. (2020) You Need to Lock Down Your Router's Remote Management Options. [Online]. Available: https://lifehacker.com/you-need-to-lock-down-your-routers-remote-management-op-1842525275

[37] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A Large-scale Analysis of the Security of Embedded Firmwares," in *USENIX Security Symposium (USENIX Security)*, 2014, pp. 95–110.

[38] E. Klein, G. V. de Velde, R. Droms, T. L. Hain, and B. E. Carpenter. (2007) Local Network Protection for IPv6. [Online]. Available: https://www.rfc-editor.org/info/rfc4864

[39] (2024) No-IP Homepage. [Online]. Available: https://www.noip.com/

[40] (2024) DynDNS Homepage. [Online]. Available: https://account.dyn.com/

[41] (2024) Oray Homepage. [Online]. Available: https://www.oray.com/

[42] D. M. Junior, L. Melo, H. Lu, M. d'Amorim, and A. Prakash, "A Study of Vulnerability Analysis of Popular Smart Devices Through Their Companion Apps," in *IEEE Symposium on Security and Privacy Workshops (SPW'19)*, 2019, pp. 181–186.

[43] B. Miller, T. Nixon, C. Tai, and M. Wood, "Home Networking with Universal Plug and Play," *IEEE Communications Magazine (IEEE COMMUN MAG)*, vol. 39, no. 12, pp. 104–109, 2001.

[44] S. Esnaashari, I. Welch, and P. Komisarczuk, "Determining Home Users' Vulnerability to Universal Plug and Play (UPnP) Attacks," in *International Conference on Advanced Information Networking and Applications Workshops (WAINA'13)*, 2013, pp. 725–729.

[45] kaklakariada. (2015) UPnP PortMapper. [Online]. Available: https://github.com/kaklakariada/portmapper

[46] (2024) FileZilla - The Free FTP Solution. [Online]. Available: https://filezilla-project.org/

[47] D. Giese and G. Noubir, "Amazon Echo Dot or the Reverberating Secrets of IoT Devices," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'21)*, 2021, pp. 13–24.

[48] P. Liu, S. Ji, L. Fu, K. Lu, X. Zhang, J. Qin, W. Wang, and W. Chen, "How IoT Re-using Threatens Your Sensitive Data: Exploring the User-Data Disposal in Used IoT Devices," in *IEEE Symposium on Security and Privacy (S&P'23)*, 2023, pp. 1845–1861.

[49] MarketWatch. (2023) Home Wireless Router Market Size 2023-2030 | Detailed Analysis of Market Size and Growth Rate. [Online]. Available: https://www.marketwatch.com/press-release/home-wireless-router-market-size-2023-2030-detailed-analysis-of-market-size-and-growth-rate-2023-05-08

[50] ZOL. (2023) 2023 Wireless Router Brand Rankings. [Online]. Available: https://top.zol.com.cn/compositor/227/manu_attention.html

[51] P. Jeitner, H. Shulman, L. Teichmann, and M. Waidner, "XDRI Attacks - and - How to Enhance Resilience of Residential Routers," in *USENIX Security Symposium (USENIX Security)*, 2022, pp. 4473–4490.

[52] J. Davies. (2007) The Cable Guy IPv6 Autoconfiguration in Windows Vista. [Online]. Available: https://learn.microsoft.com/en-us/previous-versions/technet-magazine/cc137983(v=msdn.10)?redirectedfrom=MSDN

[53] D. T. Narten, R. P. Draves, and S. Krishnan. (2007) Privacy Extensions for Stateless Address Autoconfiguration in IPv6. [Online]. Available: https://www.rfc-editor.org/info/rfc4941

[54] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce Reuse in WPA2," in *ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, 2017, pp. 1313–1328.

[55] Cisco. (2007) Understanding and Configuring Dynamic ARP Inspection. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/dynarp.html

[56] Y. Wu, J. Wang, Y. Wang, and S. Zhai, "Your Firmware Has Arrived: A Study of Firmware Update Vulnerabilities," in *USENIX Security Symposium (USENIX Security)*, 2024.

[57] E. Veroni, C. Ntantogian, and C. Xenakis, "A Large-scale Analysis of Wi-Fi Passwords," *Journal of Information Security and Applications (JISA'22)*, vol. 67, p. 103190, 2022.

[58] D. J. Ledkov. (2020) wget: implement TLS verification with ENABLE_FEATURE_WGET_OPENSSL. [Online]. Available: https://git.busybox.net/busybox/commit/?id=45fa3f18adf57ef9d743038743d9c90573aeeb91

[59] devttys0. (2014) Binwalk. [Online]. Available: https://github.com/ReFirmLabs/binwalk

[60] (2024) Google Domains. [Online]. Available: https://domains.google/

[61] J. Wang, S. Guo, W. Diao, Y. Liu, H. Duan, Y. Liu, and Z. Liang, "CrypTody: Cryptographic Misuse Analysis of IoT Firmware via Data-flow Reasoning," in *International Symposium on Research in Attacks, Intrusions and Defenses (RAID'24)*, 2024, pp. 579–593.

[62] D. Pointcheval, *OAEP: Optimal Asymmetric Encryption Padding*. Boston, MA: Springer US, 2011, pp. 882–884.

[63] K. Liu, M. Yang, Z. Ling, Y. Zhang, C. Lei, L. Luo, and X. Fu, "SAMBA: Detecting SSL/TLS API Misuses in IoT Binary Applications," in *IEEE International Conference on Computer Communications (INFOCOM'24)*, 2024, pp. 1–10.

[64] C. Meijer, V. Moonsamy, and J. Wetzels, "Where's Crypto?: Automated Identification and Classification of Proprietary Cryptographic Primitives in Binary Code," in *Annual Computer Security Applications Conference (ACSAC'14)*, 2021, pp. 555–572.

[65] L. Zhang, J. Chen, W. D. B, S. Guo, J. Weng, and K. Zhang, "CRYPTOREX: Large-scale Analysis of Cryptographic Misuse in IoT Devices," in *International Symposium on Research in Attacks, Intrusions and Defenses (RAID'19)*, 2019, pp. 151–164.

[66] D. Wang, M. Jiang, R. Chang, Y. Zhou, H. Wang, B. Hou, L. Wu, and X. Luo, "An Empirical Study on the Insecurity of End-of-Life (EoL) IoT Devices," *IEEE Transactions on Dependable and Secure Computing (TDSC'24)*, vol. 21, no. 4, pp. 3501–3514, 2024.

[67] B. Zhao, S. Ji, W.-H. Lee, C. Lin, H. Weng, J. Wu, P. Zhou, L. Fang, and R. Beyah, "A Large-Scale Empirical Study on the Vulnerability of Deployed IoT Devices," *IEEE Transactions on Dependable and Secure Computing (TDSC'22)*, vol. 19, no. 3, pp. 1826–1840, 2022.

[68] A. Cui and S. J. Stolfo, "A Quantitative Analysis of the Insecurity of Embedded Network Devices: Results of a Wide-area Scan," in *Annual Computer Security Applications Conference (ACSAC'10)*, 2010, pp. 97–106.

[69] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: Security Evaluation of Home-Based IoT Deployments," in *IEEE Symposium on Security and Privacy (S&P'19)*, 2019, pp. 1362–1380.

[70] M. Niemietz and J. Schwenk, "Owning Your Home Network: Router Security Revisited," in *Web 2.0 Security & Privacy (W2SP'15)*, 2015.

[71] M. T. Paracha, D. J. Dubois, N. Vallina-Rodriguez, and D. Choffnes, "IoTLS: Understanding TLS Usage in Consumer IoT Devices," in *ACM SIGCOMM Internet Measurement Conference (IMC'21)*, 2021, pp. 165–178.

Junjian Ye is a PhD Student in Information Security at Nanjing University of Posts and Telecommunications. He has participated in several projects related to the security of smart homes and wireless routers. His research fields concern IoT security.

Xavier de Carné de Carnavalet is a Lecturer at the Hong Kong Polytechnic University. He received his Ph.D. degree in Information and Systems Engineering in 2019 from Concordia University, Montreal, QC, Canada. He was a post-doctoral fellow from 2019 to 2020 at Carleton University, Ottawa, ON. He holds a Dipl-Ing. from École Supérieure d'Informatique, Électronique et Automatique, Paris, France, and a M.A.Sc. in Information Systems Security from Concordia University. His research aims to measure and improve user privacy and security from threats posed by software, web technologies and services.

Lianying Zhao is an Assistant Professor at the School of Computer Science, Carleton University, Ottawa, Canada, prior to which he had 6 years of experience working mainly on mainframes at IBM. He received his Ph.D. degree from Concordia University, Montreal in 2018 and was an NSERC postdoctoral research fellow at the University of Toronto in 2019. His primary research interests include hardware/architectural security support (in particular, trusted computing), systems/platform security (firmware, hypervisor and OS), authentication, privacy preservation and data protection, which has lead to publications at security venues such as NDSS, CCS, TIFS, FC and RAID.

Mengyuan Zhang is an assistant professor at the Vrije Universiteit Amsterdam, Netherlands. She received the Ph.D. degree in information and systems engineering from Concordia University, Montreal, Canada. Previously, she was an research assistant professor with the Department of Computing at the Hong Kong Polytechnic University, Hong Kong. She also worked as an experienced researcher at Ericsson Research, Montreal, Canada. Her research interests include security metrics, attack surface, cloud computing security, and applied machine learning in security. She has published several research papers and book chapters on the aforementioned topics in peer-reviewed international journals and conferences such as the IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, CCS, and ESORICS.

Lifa Wu was born in 1968 and received the Ph.D. degree from Nanjing University in 1998. He is currently a Professor with Nanjing University of Posts and Telecommunications. His research fields concern network security and software security.

Wei Zhang is a Professor at the School of Computer Science, Nanjing University of Posts and Telecommunications, China. He received the Ph.D. degree from Soochow University in 2008. His research fields concern malware detection, privacy compliance testing and AI security.