DOI: https://doi.org/10.36910/6775-2524-0560-2021-42-21
УДК: 004.021, 004.71
Костючко Сергій Миколайович, к.т.н., доцент
https://orcid.org/0000-0002-1262-6268
Кирилюк Людмила Миколаївна, асистент
https://orcid.org/0000-0002-8279-3133
Чернящук Наталія Леонідівна, д.п.н., професор
https://orcid.org/0000-0002-3178-8377
Бортник Катерина Яківна, к.т.н., доцент
https://orcid.org/0000-0001-5282-099X
Гринюк Сергій Васильович, асистент
https://orcid.org/0000-0002-0080-3167
Луцький національний технічний університет

WIRELESS ACCESS POINT WITH MULTILAYER DATA PROTECTION ALGORITHM

Костючко С., Кирилюк Л., Чернящук Н., Бортник К., Гринюк С. Бездротова точка доступу з багаторівневим алгоритмом захисту даних. В статті запропоновано багаторівневий алгоритм захисту передачі інформації засобами бездротового маршрутизатора та Raspberry Pi. Запропонований алгоритм реалізовано на базі окремих засобів шифрування, які працюють паралельно захищеними каналами зв'язку. Методом направлення трафіку на анонімні і захищені сервери (Tor) та шифруванням інформації і паралельним використанням DNS-проксі, інформація доступна тільки легітимним користувачам.
Ключові слова: Raspberry Pi, SOCKet, DNS-проксі, багаторівневий алгоритм, шифрування даних.

Костючко С., Кирилюк Л., Чернящук Н., Бортник К., Гринюк С. Беспроводная точка доступа с многоуровневым методом защиты данных. В статье предложен многоуровневый алгоритм защиты передачи информации средствами беспроводного маршрутизатора и Raspberry Pi. Предложенный алгоритм реализован на базе отдельных средств шифрования, работающих параллельно защищенными каналами связи. Методом направления трафика на анонимные и защищены серверы (Tor) и шифрованием информации и параллельным использованием DNS-прокси, информация доступна только легитимным пользователям.
Ключевые слова: Raspberry Pi, SOCKet, DNS-прокси, многоуровневый алгоритм, шифрования данных.

Kostiuchko S., Kyryliuk L., Chernyashchuk N., Bortnyk K., Hrunjuk S. Wireless access point with multilayer data protection algorithm. The article proposes a multi-level algorithm for protecting information transmission by means of a wireless router and Raspberry Pi. The proposed algorithm is implemented on the basis of separate encryption tools operating in parallel with secure communication channels. By directing traffic to anonymous and protected servers (Tor) and by encrypting information and using a DNS proxy in parallel, the information is available only to legitimate users.
Keywords: Raspberry Pi, SOCKet, DNS proxy, multilevel algorithm, data encryption.

Introduction

Nowadays, such a concept as a cyber attack is very common, and it is becoming an increasing headache for many IT companies and enterprises. In recent years, some big names like Google, The New York Times, Facebook, etc. have been the victims of hacks.

Most of the information on networks is transmitted and stored in clear text. For example, when entering a forum, a username and password are entered, a message is written - both a username and a password, and the message is transmitted in clear text, as plain text. Moreover, many nodes are involved in data transmission, and data interception is possible on almost each of them. This is possible as in a local network by a novice hacker who downloaded a program for penetration testing of wireless networks and managed to find a password for your Wi-Fi, this is possible at the level of a city provider, where an advanced and rather curious administrator sits, this is possible on subsequent nodes up to hosting forum where you chat.

To somehow protect against this, popular sites (postal services, social networks, and others) acquired certificates, their meaning is that the exchange of data between the site and you is now encrypted. Now a novice hacker, advanced administrator and others along the chain will not be able to intercept your data so easily.

To protect the user's personal data, there are many protection algorithms, each of which in turn has a number of vulnerabilities. Therefore, it is advisable to create a multi-level algorithm for protecting personal data, which combines several security technologies.

Protection algorithm development

The following technologies will be used in the developed protection algorithm:

• Tor;
• SOCKS5;
• DNS proxy.

Tor (The onion routing) - onion routing technology, which is a distributed system of servers, between which traffic passes in encrypted form. At the last node in the chain, the transmitted data goes through the decryption procedure and is transmitted to the target server in clear text. The i2p technology was considered as an analogue of the Tor technology. I2p (invisible internet project) - a project to create an anonymous computer network running over the global Internet. i2p is a more complex solution for ordinary users, while it has a number of disadvantages, such as lack of decentralized servers and insufficiently good encryption methods.

When comparing the two technologies, it was decided to use Tor because it is a more convenient and more functional method of ensuring security.

Tor technology is vulnerable to an attack in which the last node on the network is spoofed and traffic passing through it can get to the attacker who carried out the attack. In order not to transmit traffic in the clear, even at the last node, the traffic is pre-encrypted using Socks5 technology.

SOCKS5 is a network protocol that allows packets to be sent from a client to a server through a proxy server, encrypting data in transit. In this case, a protocol with the ability to authorize is used, which allows only legitimate users to gain access to this server.

An analogue for the SOCKS5 protocol can be a VPN server. VPN (Virtual Privat Network) is a server that also encrypts data in transit, but unlike the chosen Socks5 technology, it becomes necessary to rent a VPN server, which in turn leads to additional financial costs. At the same time, setting up this technology is possible independently, but it requires a lot of time and specialized skills in this area.

Tor technologies together with SOCKS5 technology ensure the security of user data transmission, excluding most possible attacks.

The developed multi-level algorithm for protecting user data makes it possible to exclude the option of spoofing the target server. DNS proxy technology is responsible for this function.

DNS (Domain Name System) is a protocol that interprets a literal domain name into an IP address. In simple terms, its main function is to turn a user-friendly domain name into an IP address.

This protocol is susceptible to several types of attacks, such as DNS spoofing and Fast Flux DNS, in order to prevent these attacks, DNS proxy technology is used, which allows you to use a private DNS server, thereby protecting the user from data leakage through the domain name system. The operation scheme of the multi-level data protection algorithm is shown in Figure 1.

To implement the developed multi-level protection algorithm and to provide secure access to the global Internet for several users at the same time, a device is required that can act as an access point with the ability to build a user's route to the Internet using an encryption and data protection algorithm.

Figure1 – scheme of the multi-level data protection algorithm

Element base selection and calculation of the portable router cost.

As a hardware platform for creating a portable wireless access point with a personal data protection system, the Raspberry Pi 4B microcomputer was chosen, which is optimal for creating such projects.

The Raspberry Pi is a Raspbian (-unix-like operating system) single board computer, the main feature of which was to be powerful hardware and a budget price. It quickly gained popularity due to its ease of use, wide range of possibilities, low price, and now it is the most popular single-board computer in the world.

Today, many research projects in the field of technical creativity, such as game consoles, storage servers, media players and many others, are designed based on the Raspberry Pi single-board microcomputer.

Figure 2 – Part of wireless access point (Raspberry Pi 4B, Bluetooth keyboard, LAN cable)

The Raspberry Pi 4B model we have chosen has a built-in WI-FI adapter, Bluetooth, Ethernet port, as well as modern technical characteristics.

Step 1. Updating the system and installing HostAPD

In the process of setting up we will need the following tools:

utilities for working with the bridge-utils network bridge

daemon hostapd AP

DHCP server

In addition, it is worth updating the system, because with each updated version of the kernel, driver versions are updated, which will allow you to expand the capabilities of the adapter.

Step 2. Interface settings

Since we are going to configure a static wireless IP address, we will later need it to ignore the wlan0 interface, the default Wi-Fi card and not allow other interfaces to use it. This should isolate our access point from anyone interfering with its work.

Step 3. Static IP address

To access the device at a permanent address, you need to configure a static IP for the Raspberry Pi after setting up network access. First, this should be done on the router, because there will be a problem with the address conflict. Raspbian uses the dhcpcd service and the dhcpcd.conf configuration file to configure these settings. By default, the addresses of all interfaces are requested from the router via DHCP. [16]

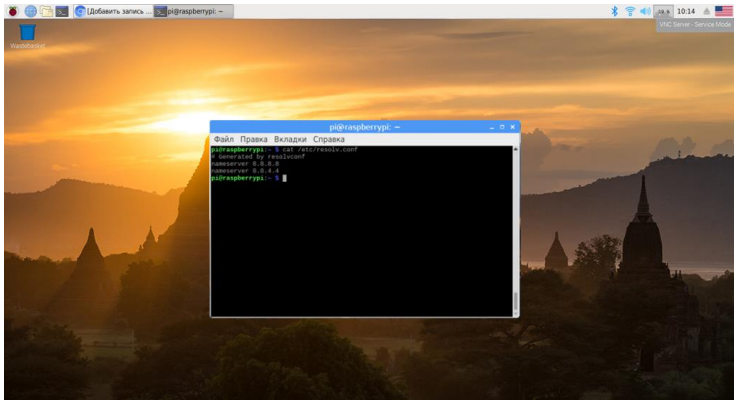

Figure 3 – Use the wpa_passphrase utility to encrypt the password

Step 4. Configure Hostapd

After making all the previous settings, proceed to the actual setting of the access point - hostapd. To do this, make changes to the file hostapd.conf at / etc / hostapd. This file contains a lot of detailed information about commands and settings. Upon request

sudo cp /etc/hostapd/hostapd.conf /etc/hostapd/hostapd.conf.origina

you can access the original version of the file in case of loss of information and unintentional damage. [17]

Step 5. Configuring Dnsmasq

The default dnsmasq config file is very complex and won't work for our purposes. It is much easier to create a new config file.

It's a good idea to move the default config file somewhere just in case. We can do this by typing

sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

into the console and then creating our own empty file by typing

sudo nano /etc/dnsmasq.conf.

We also need to do a few things: bind the interfaces again to make sure they are not sending anything, redirect to Google DNS queries rather than short names, remove the non-routable address, and assign IP addresses between 192.168.220.50-150 from 12- hour rent.

Step 6. IPv4 forwarding

We now have a few things to do on the Wi-Fi side, as we won't have an access point if it can't connect to the internet. We fix this by redirecting wlan0 traffic to the Ethernet connection. When IPv4 forwarding is working properly, we can get our NAT (network address translation) between wlan0 and eth0 interfaces. To do this we update iptables.

Step 7. Enabling services

The final step is to start hostapd by typing

sudo service hostapd start

and start dnsmasq by typing

sudo service dnsmasq start.

After entering each of these commands, wait a few seconds and use a different Wi-Fi device to see if the hotspot is visible.

Conclusion.

As a result of research work, on the basis of the Raspberry Pi 4b microcomputer, a wireless access point (router) was created with a developed multi-level security algorithm that can encrypt incoming and outgoing traffic, as well as ensure user anonymity when working on the global Internet. The wireless access point (router) developed in a research project, in addition to a high degree of data protection, has a compact size, the ability to supply power from 5 volts, unlike factory counterparts.

References.

1. S. Kostiuchko and V. Tchaban, "Variational Method of Auxiliary Equations in Nonlinear Systems Analysis and Synthesis Problems," *2019 IEEE 20th International Conference on Computational Problems of Electrical Engineering (CPEE)*, Lviv-Slavske, Ukraine, 2019, pp. 1-5, doi: 10.1109/CPEE47179.2019.8949123.
2. S. Kostiuchko, O. Kuzmych, A. Aitouche, S. Grinyuk and O. Mekush, "Application of Parametric Sensitivity Method to Analysis of Automatic Mooring Winch with Electric Drive System," *2019 4th Conference on Control and Fault Tolerant Systems (SysTol)*, Casablanca, Morocco, 2019, pp. 294-299, doi: 10.1109/SYSTOL.2019.8864751.
3. "PlanetLab – an open platform for developing, deploying, and accessing planetary-scale services." URL: https://www.planet-lab.org
4. K. Ali and M. Scarr, "Robust methodologies for modeling web click distributions," in WWW. ACM, 2007. URL: https://nymity.ch/tor-dns/ pdf/Ali2007a.pdf
5. J. Alstott, E. Bullmore, and D. Plenz, "powerlaw: A Python package for analysis of heavy-tailed distributions," PLoS ONE, vol. 9, no. 1, 2014. URL: https://nymity.ch/tor-dns/pdf/Alstott2014a.pdf

6. Amazon Web Services, "Alexa top sites." URL: https://aws.amazon. com/alexa-top-sites/
7. H. Asghari, "pyasn – Python IP address to autonomous system number lookup module." URL: https://github.com/hadiasghari/pyasn
8. S. Banerjee, T. G. Griffin, and M. Pias, "The interdomain connectivity of PlanetLab nodes," in PAM. Springer, 2004. URL: https://nymity.ch/tor-dns/pdf/Banerjee2004a.pdf
9. S. Bortzmeyer, "RFC 7816 – DNS query name minimisation to improve privacy," Mar. 2016. URL: https://tools.ietf.org/html/rfc7816
10. X. Cai, R. Nithyanand, and R. Johnson, "CS-BuFLO: A congestion sensitive website fingerprinting defense," in WPES. ACM, 2014. URL: https://nymity.ch/tor-dns/pdf/Cai2014a.pdf
11. X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: Website fingerprinting attacks and defenses," in CCS. ACM, 2012. URL: https://nymity.ch/tor-dns/pdf/Cai2012a.pdf
12. A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," SIAM Review, vol. 51, no. 4, 2009. URL: https://arxiv.org/pdf/0706.1062
13. G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in Security & Privacy. IEEE, 2003. URL: https://nymity.ch/tor-dns/pdf/Danezis2003a.pdf
14. R. Dingledine and N. Mathewson, "Tor protocol specification." URL: https://spec.torproject.org/tor-spec
15. R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The secondgeneration onion router," in USENIX Security. USENIX, 2004. URL: https://nymity.ch/tor-dns/pdf/Dingledine2004a.pdf
16. https://www.raspberrypi.org/documentation/configuration/wireless/access-point-routed.md
17. https://learn.pi-supply.com/make/how-to-setup-a-wireless-access-point-on-the-raspberry-pi/