Net-Trainer-ST: A Swiss Army Knife for Pentesting, Based on Single Board Computer, for Cybersecurity Professionals and Hobbyists

K. Hołda, D. Śliwa, K. Daniec

***Abstract*—**This article was created as part of the developed master's thesis. It attempts to present a developed device, which will support the work of specialists dealing with broadly understood cybersecurity terms. The device is contrived to automate security tests. In addition, it simulates potential cyberattacks in the most realistic way possible, without causing permanent damage to the network, in order to maximize the quality of the subsequent corrections to the tested network systems. The proposed solution is a fully operational prototype created from commonly available electronic components and a single board computer. The focus of the article is not only put on the hardware part of the device but also on the theoretical and applicatory way in which implemented cybersecurity tests operate and examples of their results.

***Keywords*—**Raspberry Pi, ethernet, automated cybersecurity tests, ARP, DNS, backdoor, TCP, password sniffing.

I. Introduction

TWO decades have passed since the most upsetting issue for a great deal of the internet users was the internet connection speed. Every device in every household, office and factory was connected through a twisted pair and coaxial cables, wireless connection was not very common. Nowadays, Wi-Fi, hotspots and other forms of wireless internet connection are present almost everywhere [1]. It is abnormally difficult to find a device that does not have a Wi-Fi module built in, with industry bent on connecting every new piece of technology to the global network, starting from smartphones through to cars ending in fridges or dishwashers. The craze to connect every single thing to the internet does not seem to go away any time soon [2]. Alas, even if those marvels of technology offer an outstanding comfort in using them, unprotected, they create massive holes in the defense mechanisms of every network they are connected to.

II. Solutions Overview

A. Literature

Browsing through available literature can yield ready and working receipts for problem solutions tied to cybersecurity; alas in most of the cases, these researches' focal point is singular type of threats or group of threats and their analysis. During the literature review, articles treating theoretical and practical descriptions of various vulnerabilities contained only the effects of the cyberattacks studied and statement that the type of attack was performed.

A group of scientists from Noroff University College in Norway [3] conducted research involving capturing malware type programs employing a specially crafted and configured hotspot. Through this solution, it was possible to capture command logs sent by the attacker to a network localization he was focused on. Information available in the article describing machine learning to detect DDoS attacks [4] was also analyzed. In the aforementioned article, the templates of normal unobstructed network traffic were shown, and compared with the ones affected by the attack. The next article dealt with drone hacking using Raspberry Pi [5], it referred inter alia to unsecured deauthorization frames. In the latter part of said article, the concept of automation of such an attack was presented. Another work was found that delved into broad terms of penetration tests using Raspberry Pi [6]. Thanks to the conclusions in that article, the relevance of earlier prepared plans of implementing chosen modules of the device could be evaluated. Articles describing basic attacks such as DNS attack and its more advanced forms [7] were crucial in the analysis of each step during the process of creating chosen attacks. Moreover, it is worth noting that an article from 2009 [8] was capable of providing knowledge regarding ARP attacks, and as a result of gaining such noesis, a module that simulates ARP attack was created.

B. Commercial Solutions

Wireless networks used by companies, even if they are secured against most popular cyberattacks, can be less secure than correctly configured "home" networks. There is the growing need for devices that run various scenarios used during cybersecurity audits. Alas, those devices are often not widely-available, especially in Central European countries. Tools like WiFi Pineapple, or any devices licensed by Hak5 [9] company are in most cases unobtainable, in limited amount, on secondary markets where scalpers raise the price of said devices to the amount that exceeds their original factory price.

C. Handcrafted Solutions

Hobby-made tools developed on GitHub [10], [11] for many years should also be mentioned. Woefully, most of those projects are abandoned by its creators, which makes compatibility with new forms of checking the network security

K. Hołda, D. Śliwa, and K. Daniec are with the Silesian University of Technology Faculty of Automatic Control, Electronics and Computer Science, 41100 Gliwice, Akademicka 16, Poland (e-mail: kacphol520@student.polsl.pl, danisli638@student.polsl.pl, Krzysztof.Daniec@polsl.pl).

lessen every year. In extreme cases, such projects cease to work entirely because of usage of deprecated communication standards, or the architecture of those projects makes it unable to add modules that would make them operate in wider spectrum.

Designated few devices for cybersecurity penetration testing already exist [12]; alas, they focus on checking a specific vulnerability [13] and not the broad range of present vulnerabilities. Additionally, virtually all the open-source tools require pre-defined configuration of the specific cybersecurity penetration test attack, making them quite unflexible to use in the field. In most cases, the vulnerability detection is left to the tester, and he has to do his job manually. This creates added unnecessary workload on the cybersecurity administrator that does the testing, draining his time which could already be spent on implementing fixes to the vulnerabilities.

Cybersecurity penetration test attacks offered by already existing devices frequently cause temporary damage to the network infrastructure. Repairing problems caused by those tests is sometimes tied to the need of interference in the real devices of the inner network. In addition, tools used to perform said tests are often designed to provide tests from a certain limited pool of cybersecurity attacks. Their design and structure, both hardware- and software-wise, makes adding tests for newly discovered vulnerabilities quite difficult. Cybersecurity penetration test attacks made through our tool can be used “at home”, but they do not compromise the network they are working on to outside threats. This makes testing cybersecurity possible in every environment. What we offer is a tool that is focused on having modular design and packing greater computing power that is needed to perform cybersecurity evaluation. Parts can be effortlessly replaced at low cost and are widely available, such as newer generations of Raspberry Pi.

D. Proposed Solution

For the problems set out in the previous paragraphs, we propose NET-TRAINER-ST, which offers basic attacks based on commonly used programming language [14]; a clean operating system dedicated to the device equipped with ARM processor, and additionally, fully compatible communication interface in the form of 8P8C socket. Thanks to the aforementioned socket, in the first mode of work, all network traffic can be directed through the NET-TRAINER-ST device. Whereas in the second mode, the device functions independently of the host, NET-TRAINER-ST does not even require connection to it because of dedicated UPS usage. This in turn allows conducting cybersecurity tests without using any wired connections to the NET-TRAINER-ST, which is connected to the chosen LAN network, or any other network offering an adequate socket.

Most of the composed cybersecurity tests available on NET-TRAINER-ST do not require any complicated preparations, assuming that the elementary set of software tools to monitor the network traffic is present. All the prepared cybersecurity tests will work regardless of the end-user operating system; for example, they would work both on Linux and Windows OS. They can even be used in a common household network without any need for additional laboratory preparations.

Possible test environment configurations are presented in Fig. 1, in which three example scenarios can be seen. In the first one, the device is connected to the chosen network, and runs cybersecurity tests automatically. In the second scenario case, the device is connected to the host, which gives NET-TRAINER-ST access to the said host and direct filtration of the data received. In the third scenario, the device is connected to the same network as the example PC, but the access to the given unit is only possible remotely. Aside from the presented scenarios, the device offers a multitude of other possible usage and access scenarios thanks to the dedicated hotspot sharing feature.

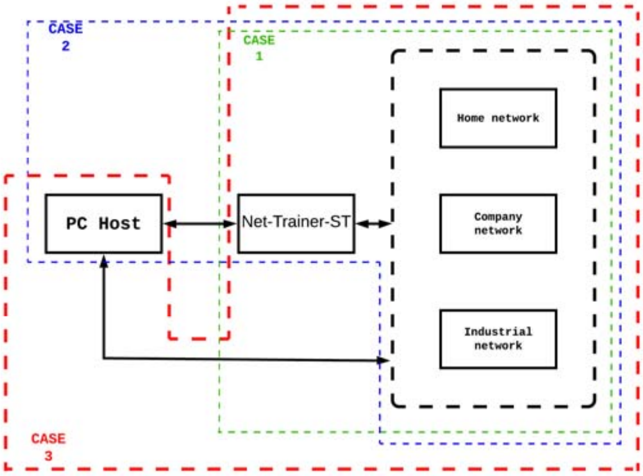


Fig. 1 Net-Trainer-ST operation in suggested examples of environments

The device filters out network traffic and will be compatible with the most possible amount of the cybersecurity penetration tests such as:

- wiretapping passwords of the TELNET, X11, ICQ, SMB, MySQL, HTTP, NNTP, SNMP, IRC, RLOGIN, POP3, RIP, IMAP 4, BGD, FTP, SOCKS 5, VNC, SSH1, LDAP, NFS and NAPSTER standards from host computer which device will be connected to.
- ARP poisoning [15]/DNS spoofing [16] with the possibility of spoofing the entire network and monitoring all the present traffic, searching and capturing for all the transferred passwords within a given network, or targeting a single connected device - in this variant, user has access to a pre-prepared DNSmask.
- reverse SSH attacks, which uses a predefined SSH client on the targeted device, or using Netcat, simple TCP/UDP communication software - which allows for creation of a reverse TCP [17] backdoor.
- reverse TCP meterpreter [18], which may require port routing on the network router in order to gain access to the backdoor which in turn uses the Metasploit [19] multi handler.

It should be mentioned that the clean Raspbian system [20] offers basically unlimited modular addition of the software and easy process to adapt them. Sharing previously mentioned attack scenarios between the devices that serve the same purpose as NET-TRAINER-ST is not a problem either.

E. Design Assumptions

To help with the network administrators and network security testers workload, we present NET-TRAINER-ST, a tool that makes independent development possible, and does not have the downside of dedicated hardware limitation which is present in other such devices. In this subsection, a general overview on the possibilities of our device will be presented, and in the next subsections a delve into the detailed structure, principles of operation and future development plans will be made.

The device is capable of redirecting whole network traffic from the host through the connected USB-A socket and RS485 cable, not only from the LAN network but the whole internet traffic. The device is also capable of working independently; thanks to this, after it is connected to the network through the 8P8C socket, fully automatic preprogrammed payloads can be effortlessly called up.

Through the support of simultaneous servicing of Ethernet port, Wi-Fi module, which makes creating a hotspot possible, and self-contained USB-A socket, the development possibilities are unlimited. The only thing that could become an obstacle is the computing force of the hardware part, if a user wanted to use dictionary-type attack or any brute force-type attack.

NET-TRAINER-ST will support two modes, "Turtle" mode which will entail basic functionalities of network adapter, saving encrypted and non-encrypted data packets, activating chosen cybersecurity attack tests, which were described in previous subsections whereas "Shark" mode, which is already being implemented at this point, will make automatic movement through the wireless network, LAN and outside network achievable. That in turn is made possible thanks to the active access of the user interface visible when logging into a given port, after the user connects to the shared hotspot of the device. Above everything previously mentioned, most of the options present in the "Turtle" mode, besides those directly tied to the host connected to the device, will be available in the second "Shark" mode, which makes all the cybersecurity tests autonomous.

High-level flow of data diagram between each component of the device is presented in Fig. 2. Inside the square boxes are shown all the main hardware parts which are required to run all the tests and modes correctly. Inside the rhombuses, all the connections between parts, interfaces, which in turn are exhibited in ovals, are demonstrated. Fig. 2 lays out a universal diagram that can be used to create a clone of the NET-TRAINER-ST device. Upon creating this device, we have followed this diagram as closely as possible. The only thing that is not represented is the change of the powerbank to the dedicated Raspberry Pi Zero UPS overlay [21].

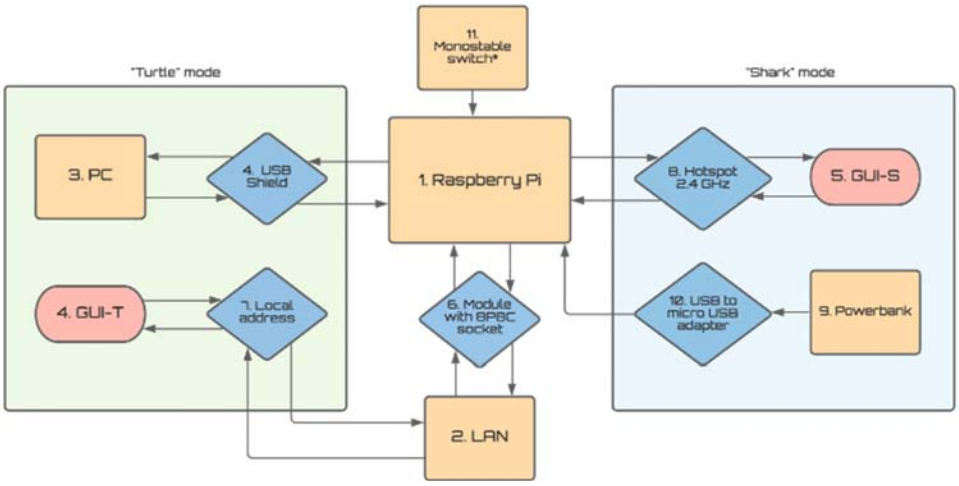


Fig. 2 Universal schematic of data flow between every device component

III. Hardware Components

The heart and soul of this device is Raspberry Pi Zero WH. It is a computer consisting of the only one electronic board which earned it its name — single board computer. It is manufactured by non-profit organization Raspberry Pi Foundation — its design is presented in Fig. 3 as a part of functioning prototype. Created to further the goal of improving the prospects of programming learning, it is widely used to produce embedded device prototypes of all sorts, but it also found its corner as a common office computer running Raspbian, Debian, Fedora, Arch Linux, or FreeBSD operating systems.

The board dimensions are 65 x 30 x 5 mm [22]. The circuit draws 150 mA. The computer is built on the Broadcom BCM2835 circuit base. This specific circuit entails functionalities like wireless LAN in the 802.11 b/g/n standard and Bluetooth 4.1 protocol. Thanks to these, the future plans hold developing a dedicated interface for the smartphones. Aside from the communication modules, the board is equipped with CPU with 1 GHz clock rate, which remains still stable at 1.5 GHz if boosted, 512 MB of RAM, mini HDMI and CSI socket, and 40 pins GPIO - used to handle UPS and ethernet module connections.

Dedicated for this model of Raspberry Pi, USB shield is used for directing data from and to the computer, which in turn is the target of the cybersecurity tests, or the computer which the testing personnel is using. Moreover, in this prototype, on which the test was run, dedicated battery shield with UPS function was used. The battery has nominal capacity of 1100 mAh, which provides the means for the device to work continuously up to 3 hours.

Dedicated GPIO pins permitted NET-TRAINER-ST to have additional 8P8C port functionality, without the need of taking the space of the micro-USB, which uses an OTG function socket. Chosen ENC28J60 module [23] is in truth dedicated for Arduino boards, but in this project, full functionality of the aforesaid module was achieved even on the Raspberry board, as the result of employing Serial Peripheral Interface (SPI).

Demonstrated in Fig. 3 is the final appearance of the prototype without being encased in the 3D printed cover. Such action was taken to show board components and LAN module wiring, as much as possible without the case obstructing the view.

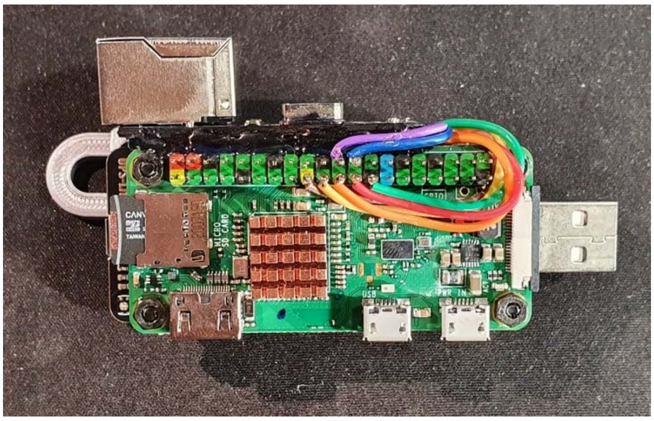


Fig. 3 Prototype used in the development of the project

IV. Security Test Modules

A. Theoretical Description

In this section, all the details regarding the cybersecurity tests that are available on the device will be laid out. The following description will be theoretical in nature and will focus on explaining the idea behind every cybersecurity test solution provided by NET-TRAINER-ST. Foremost, all the types of attacks, which target one device in the network at a time, like all kinds of backdoors or password wiretapping will be laid out. Subsequently, in the second part of this section, focus will shift to cybersecurity tests that target multiple devices in the network at the time, making attacks like spoofing possible.

- *Backdoor*: a gap in the structure of the operating system, which sabotages all the security mechanisms of a given system. It is an access point of the software or OS in question. The NET-TRAINER-ST device is equipped in a few, most universal variants of this attack as well as creating them depending on the demand.
- *Simple SSH:* uses "autossh" program to configure backdoor working on the principle of reverse activation of SSH communication protocol standard on the victim device. It is an attack of so-called reverse backdoor type, the only difference being that instead of configuring listener, NET-TRAINER-ST enables SSH protocol, which in turn gives the device access to the attacked unit through creating an additional user. This module would not have any difficulties in implementing a more complex scenario utilizing a virtual private server (VPS) [24], its task is granting an additional layer of anonymity to the attacker side. VPS is one of the methods practiced by cybercriminals for implementing broad spectrum of attacks. As a result of this solution, the attacker gains access to all the needed tools, which are installed on the server, whereas the local afflicted device displays one legal client with the admittance to the server.
- *Netcat BD* is another backdoor module which utilizes quite popular "netcat" software [25], which is a tool for monitoring and scanning networks. This program was chosen, since it can be executed without the GUI, moreover it can be scripted to complete more complex tasks. Netcat can read and write network connections using protocols like TCP and UDP. Similar to every other server, it is provided with functionality to send files and port listening, which is how it can be used as backdoor.
- *TCP Metasploit:* is third module, which allows backdoor usage. It is important to mention that this module gives access to the Meterpreter shell, which in turn has admission to the Metasploit modules and other commands unavailable to the normal level shell. Metasploit is an apparatus, which can serve to automate the process of cybersecurity testing that focuses on exploitation of the network. It can create a reverse TCP connection from the victim computer to the NET-TRAINER-ST device. This attack relies on opening a connection on the assailed device, whereas on the used by tester device side, listener configuration is in order. Such listener will wait for a connection to the targeted device. After those actions are completed, the aforementioned Metepreter shell is accessible.
- *Sniffing*: that is, intercepting data in the network traffic, is one of the most popular techniques used to gain control over an account of a user who has fallen victim to it in the chosen application, system, or even bank credential. Other than that, it could be utilized for message or email wiretapping, and other forms of digital identity theft. Sniffing is also one of the most developed methods in this project due to large quantity of already existing solutions in cybersecurity field. Ranging from simple hub with keylogger, ARP spoofing to the more complex techniques such as EvilPortal used also in WiFi Pineapple [26], DNS EvilTwin [27] or even dedicated ransomware. At this stage of NET-TRAINER-ST development, the focus was on creating the interception and detection of the passwords from the network traffic functionality.
- *Passive Sniffing Mod:* is a module presenting passive data

interception in the network traffic. Properly configured and installed on the targeted computer, in the form of NET-TRAINER-ST device, it has the potential to go unnoticed for a prolonged period of time and conduct espionage on the connected network. In the case of a complex device, it should be noted that passive data wiretapping can also be conducted through direct wired connection to the local network through hubs, routers, switches or the free 8P8C access points using the NET-TRAINER-ST adapter. This module allows detecting passwords going through the network from the wide range of protocols such as HTTP, TELNET, POP, FTP, SNMP, NNTP, MySQL, ICQ, SSH1, NFS, IRC, BGP, IMAP, SOCKS, RIP, SMB, X11, NAPSTER, ICQ, RLOGIN, LDAP and other common standards.

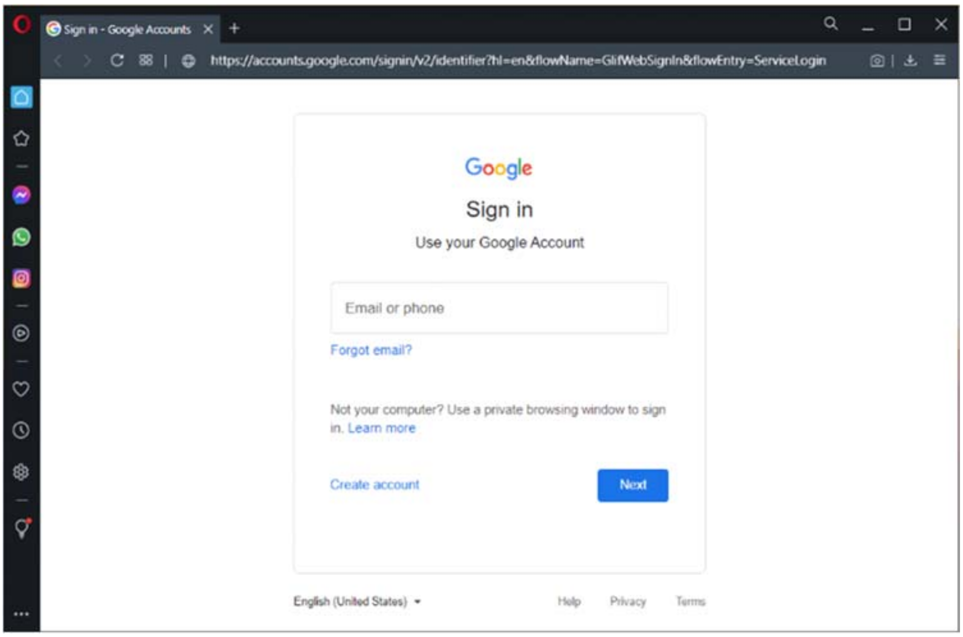
- *ARP/DNS Mod:* are active modules, created in response to the fact that network hubs are being growingly pushed out by switches or enterprise class routers. Switches can also be simulated in home environments, using opensource software OpenWRT [28], but the presented modules will work even without it.
- *ARP Mod*: can be treated as an expanded function of the Passive Sniffing module, as a result of using the same Ettercap. However, in this case, the whole network traffic, which NET-TRAINER-ST is being part of, is being monitored. This attack relies on NET-TRAINER-ST sending the falsified APR messages to the whole local network. As a result of this action, the MAC address of the used device is paired to the IP address of the server in the network or other verified unit. Upon completing those activities, NET-TRAINER-ST begins receiving and filtering all data intended for aforementioned devices, which the NET-TRAINER-ST impersonates. It should also be noted that these kinds of attacks, not only serve to detect and steal classified information but are commonly used as a basis to implement more complex attacks. More complicated attacks can exist on the grounds of ARP Spoofing, including denial-of-service attack - through redirecting network traffic designated for many IP addresses to the given targeted unit MAC address, which in turn chokes and overloads the said unit making it unresponsive and unable to work properly in the network; man-in-the-middle attack - modifying captured network traffic between victim units; session-hijacking - takeover of session identification, so as to give the attacker the access to the systems and applications as authorized individual.
- *DNS Mod:* relies on implementation of falsified network traffic from the assailed computer. It is then an attack, that targets simultaneously one computer defining anew the inner host file located in the NET-TRAINER-ST, which all network traffic goes through. Host file is a configuration file, serving the purpose of mapping domain names with accordingly assigned IP addresses. The attack leans on changing the cached DNS records, so as to the request was redirected to the predefined malicious website, in which the attacker with ease could capture previously unavailable data. In this scenario of attack, the aforementioned malicious website would be the closest possible copy of the actual website, for example: the login page of the bank account or the social media such as Twitter, Facebook or Instagram. In the more advanced version of the attack, the created malicious website is often configured to pass the entered by user data to the actual, real website of said service. This means the user will not even notice theft of credentials that just took place.
- *Masquerade Mod*: creates peculiar variation; in some measure it paraphrases the original "DNS Mod". The difference between the two lies on Masquerade using the host Dnsmasq server file instead of "hosts" file. In addition, this module sets the aforementioned server, so as to it is forced to use preprepared hosts file. As a result of possible Dnsmasq server configuration on the NET-TRAINER-ST, so the preprepared malicious website on the server would respond on the called IP address of the original website, it is even simpler to call up and process the prepared malicious website. Thanks to this module, simple server can be run on NET-TRAINER-ST device, which makes it possible to transmit requests and cached memory. Thanks to the prospect of controlling the network traffic of the connected user device, it is possible to wiretap encrypted connections, utilizing for example the mitproxy or sslplit tool. Monitoring the whole network traffic in the dedicated interface or even modifying the requests and responses for said requests upon their sending is also achievable.

B. Explanation behind Security Tests' Machinations

This chapter was dedicated to the practical implementation of the appliance of NET-TRAINER-ST modules. The focus will be on applicatory side of the project, letting the reader know-how on the functionality of every test.

- *Simple SSH*: In order to implement this module, a new user has to be created on the targeted device. Said user can have restricted permissions, it will not be obstructing the procedure of this test. It is essential to point out that the access password of the aforementioned user must be hard-coded into the configurational part of the module code. However, the code is simple to modify, so the password could be entered on every start of the device. In the next step, tester has to log in on the preprepared user and set the public key authentication. NET-TRAINER-ST authentication key is available under the terminal command "`cat ~/.ssh/id_rsa.pub`". Said key has to be copied to the authorized keys file located under path "`ssh/authorized_keys`" - which was created on previously mentioned user located on targeted device. The only thing left to do is running the module on NET-TRAINER-ST and feeding it the correct IP address (public address, local one or even the VPS one). It is important to bear in mind that SSH server in most cases works on the 22nd port, if no settings were modified on own VPS. However, in this module case, NET-TRAINER-ST connects to the SSH from 4444 port from the level of the created user, then the SSH session is started. The reason for using a port other than the standard one is that the 4444 port is a default listener port for tools like Metasploit [29].

- *Netcat BD*: In order to properly configure this module, the tester has to prepare the environment, so the used router has the "port routing" option enabled. After turning on the module, the router IP address has to be input. Next up, the previously set up port number has to be entered, which is routed as a result of the previous steps. Now, the command `"nc -l -v -p <chosen_port>"` has to be executed to configure the Netcat listener. This module behaves in the same way as the previously explained backdoor (as in the option to use VPS): this module has permanent configuration regarding opening new sessions with NET-TRAINER-ST.
- *TCP Metasploit*: During the employment of this backdoor module, the multi handler available in the Metasploit tool is used. If the testing surrounding is placed in the home environment, using the standard Wi-Fi router, the port routing setting needs to be configured the same way as in Netcat BD module. After invoking the module, the IP address of the targeted device, which has Metasploit installed, needs to be inputted. The same has to be done with port number, on which the listening will take place. After those actions are completed, the NET-TRAINER-ST is ready to work. Directly after, the tester should start the listener service on his device. To accomplish that, The Multi Handler (command: `"use exploit/multi/handler"`) should be executed in the Metasploit terminal, and then the tester should invoke "reverse tcp" attack through the executing `"set payload python/meterpreter/reverse_tcp"` command. At that moment, the only thing left to do is, through the help of correct commands, set the local IP host: `"set lhost <local_IP>"`, and then the port, on which the listening will proceed: `"set lport <chosen_port>"`. At the end, the tester executes a simple command `"exploit"`, which commences listening a previously predefined connection with NET-TRAINER-ST; and, as a result of that, and after a few seconds pass, the freshly created backdoor can be used to control the target device.
- *Passive Sniffing Mod*: This module makes use of added ethernet interface and tool named Ettercaps [30]. After invoking the module, it starts to intercept network traffic and saves accumulated data into a log file. This log file can be browsed through once the listening procedure is done. Now, the only thing left to do is to use Etterlog [31] to find the passwords and every defined network movement that was recorded in the log file.
- *ARP Mod*: As in the case of passive listening, this module makes use of the ethernet port and Ettercap tool. After selecting this module on NET-TRAINER-ST, the device will attempt to falsify network traffic so that every ARP request sent from computers in the same network that will receive the MAC address of the NET-TRAINER-ST device. As a result of this, every query to the dynamic ARP table with IP addresses from the computers in the network will receive the NET-TRAINER-ST MAC address in response. Now, the module will automatically turn on the IP forwarding service, so as to receive answers for queries through all rerouted computers, which makes it look like it is functioning properly. After the module work is complete, the log file will be available on the device memory, in which all the details will be present. This file can also be browsed through with the Etterlog tool.



https://accounts.google.com/signin/v2/identifier?hl=en&flowName=GlifWebSignIn&flowEntry=ServiceLogin

Not your computer? Use a private browsing window to sign in. Learn more

Fig. 4 Malicious login website appearance, prepared for "DNS Mod" test procedure

- *DNS Mod*: In order for this module to work properly, the /home/pi/mods/hosts file needs to be modified. The file has to be altered in such a way that domain name will match to the correlated IP address, which redirects to a falsified, malicious website. An example of such falsified site can be seen in Fig. 4. Following this, the module needs to be invoked from the main module menu, and the remainder of the processes will complete automatically. From this moment, the NET-TRAINER-ST will begin falsifying incoming network traffic to the connected computer through the hosts file.
- *DNSmasq Mod*: This module is quite similar to the "DNS Mod" module, as a result of the preparatory activities prior to the launch of the module being the same. Meaning, the tester has to modify the /home/pi/mods/dns/hosts file on the NET-TRAINER-ST device accordingly. After that is done, the module can be run from the main module menu. Furthermore, additional options given by the DNSmasq server [32] can be tested, which can be defined in the configuration file of the server [33]. This file is located on the NET-TRAINER-ST device in `"/etc/dnsmasq.conf"`.

V. Conclusion

In this article most important aspects of the created prototype were laid out. It has been shown that, the NET-TRAINER-ST device compared to the other commercially available alternatives is drastically more elastic in terms of hardware, thanks to its modular build, resulting in a wide range of possibilities when it comes to developing software. Care has been taken in the developing stage for all modules in order to be as user-friendly as possible, even for beginner pentesters. Already in the conceptual phase of the NET-TRAINER-ST project, it was pointed out that the use of new, fresh communication technologies in LANs and wireless networks that do not go hand in hand with adequate security of these networks [34]. All the available modules are used for simulating attacks, which are still very common in terms of their occurrences in corporate networks and growingly in the industrial networks, as a result of companies adopting IoT technologies, the 4th industrial revolution and cloud computing [35].

The ability to easily access all the hardware components of the prototype, the open-source nature of the implemented code, as well as the systematic addition of newer modules and extensions to existing modules, places the NET-TRAINER-ST in a strong position amongst the pentesting tools available in the market. In addition, plans to add two more modules to the device, thus opening more doors to new cybersecurity branches, places the device in a strong position to be called the ultimate pocket Swiss army knife for cybersecurity testers.

Active testing of introduced security features, often by simulating selected attacks in a 1:1 way, is the best form of their verification. In addition, the automation of attacks results in faster implementation of the required fixes so that a real attack cannot take place. Thanks to the work started on the NET-TRAINER-ST, it was possible to determine how much is still to be done in terms of providing the best possible security for corporate and industrial networks.

References

[1] Gohdes, A. R. (2020). Repression technology: Internet accessibility and state violence. American Journal of Political Science, 64(3), 488-503.

[2] Abhishek Mukherjee, Bill Rojas, Hugh Ujhazy, "Business Models for the Long-Term Storage of Internet of Things Use Case Data", Market Perspective - Doc # AP45984120, 07.2020

[3] E. D. Martin, J. Kargaard and I. Sutherland, "Raspberry Pi Malware: An Analysis of Cyberattacks Towards IoT Devices," 2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2019, pp. 161-166, doi: 10.1109/DESSERT.2019.8770027.

[4] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 29-35, doi: 10.1109/SPW.2018.00013.

[5] O. Westerlund and R. Asif, "Drone Hacking with Raspberry-Pi 3 and WiFi Pineapple: Security and Privacy Threats for the Internet-of-Things," 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), 2019, pp. 1-10, doi: 10.1109/UVS.2019.8658279.

[6] M. Yevdokymenko, E. Mohamed and P. Onwuakpa, "Ethical hacking and penetration testing using raspberry PI," 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), 2017, pp. 179-181, doi: 10.1109/INFOCOMMST.2017.8246375.

[7] HUDAIB, Adam Ali Zare; HUDAIB, E. A. Z. DNS advanced attacks and analysis. International Journal of Computer Science and Security (IJCSS), 2014, 8.2: 63.

[8] Hsiao, H. W., Lin, C. S., & Chang, S. Y. (2009, August). Constructing an ARP attack detection system with SNMP traffic data mining. In Proceedings of the 11th international conference on electronic commerce (pp. 341-345).

[9] "hotplug attack tools" (Online) Available on: https://shop.hak5.org/collections/hotplug-attack-tools

[10] "MouseJack device discovery and research tools" (Online) Available on: https://github.com/BastilleResearch/mousejack

[11] "P4wnP1 ALOA payloads" (Online) Available on: https://github.com/akhil1136/P4wnP1-ALOA-payloads

[12] Fábio Mestre, "P4wnP1 A.L.O.A.— An advanced HID attack device" (Online) Available on: https://medium.com/azkrath/p4wnp1-a-l-o-a-an-advanced-hid-attack-device-d906ae5bf48c

[13] "LAN TURTLE BASICS" (Online) Available on: https://docs.hak5.org/hc/en-us/articles/360010554853-LAN-Turtle-Basics

[14] Redondo, J. M., & Ortin, F. (2014). A comprehensive evaluation of common python implementations. IEEE Software, 32(4), 76-84.

[15] Nachreiner, C. (2003). Anatomy of an ARP poisoning attack. Retrieved July, 4, 2005.

[16] Brahara, B., Syamsuar, D., & Kunang, Y. N. (2020). Analysis of malware DNS attack on the network using domain name system indicators. Journal of Information Systems and Informatics, 2(1), 131-153.

[17] Atwell, C., Blasi, T., & Hayajneh, T. (2016, April). Reverse TCP and social engineering attacks in the era of big data. In 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS) (pp. 90-95). IEEE.

[18] "Reverse TCP shell with Metasploit" (Online) Available on: https://hacksland.net/reverse-tcp-shell-with-metasploit/

[19] "Meterpreter" (Online) Available on: https://doubleoctopus.com/security-wiki/threats-and-tools/meterpreter/

[20] Harrington, W. (2015). Learning raspbian. Packt Publishing Ltd.

[21] „MAX17040-MAX17041 1-Cell-2-Cell Fuel Gauge with ModelGauge" (Online) Available on: https://datasheets.maximintegrated.com/en/ds/MAX17040-MAX17041.pdf

[22] „Mechanical Drawings" (Online) Available on: https://datasheets.raspberrypi.com/rpizero/raspberry-pi-zero-w-mechanical-drawing.pdf

[23] „Stand-Alone Ethernet Controller with SPI Interface" (Online) Available on: http://ww1.microchip.com/downloads/en/devicedoc/39662c.pdf

[24] "Virtual Private Server (VPS) or Virtual Dedicated Server (VDS)" (Online) Available on: searchservervirtualization.techtarget.com

[25] Kurth, M., Gras, B., Andriesse, D., Giuffrida, C., Bos, H., & Razavi, K. (2020, May). NetCAT: Practical cache attacks from the network. In 2020 IEEE Symposium on Security and Privacy (SP) (pp. 20-38). IEEE.

[26] Mojica Serrano, A. (2019). Hacking Mobile Devices Using WiFi Pineapple Nano. Computer Science Program.

[27] Lanze, F., Panchenko, A., Ponce-Alcaide, I., & Engel, T. (2014, September). Undesired relatives: protection mechanisms against the evil twin attack in IEEE 802.11. In Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks (pp. 87-94).

[28] Fainelli, F. (2008, January). The OpenWrt embedded development framework. In Proceedings of the Free and Open Source Software Developers European Meeting (p. 106). sn.

[29] Kennedy, D., O'gorman, J., Kearns, D., & Aharoni, M. (2011). Metasploit: the penetration tester's guide. No Starch Press.

[30] Roy, D., Moazzami, K., & Singh, R. (2007). ARP Spoofing and Man in the Middle attack using Ettercap. School of Computer Science, University of Windsor, Canada.

[31] ” etterlog - Log analyzer for ettercap log files” (Online) Available on: http://manpages.ubuntu.com/manpages/impish/en/man8/etterlog.8.html

[32] Kelley, S. (2014). Dnsmasq-network services for small networks. Accessed 2016-04-27. URL: http://www. thekelleys. org. uk/dnsmasq/doc. html.

[33] „Setup the Ethernet gadget of PI Zero with dnsmasq” (Online) Available on: https://maxammann.org/posts/2019/03/setup-g_ether-dhcp/

[34] „134 Cybersecurity Statistics and Trends for 2021” (Online) Available on: https://www.varonis.com/blog/cybersecurity-statistics/

[35] „Cloud computing - statistics on the use by enterprises” (Online) Available on: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Cloud_computing_-_statistics_on_the_use_by_enterprises