ORIGINALAN NAUČNI RAD – ORIGINAL SCIENTIFIC PAPER 10.7251/ZRPIM2201482P

NETWORK AND PORT SCANNER - A TOOL FOR GOOD OR BAD

Predrag Pecev[1,2], Anita Milosavljević[3]

[1]University of Novi Sad, iDEAlab, Dr Ilije Đuričića 3, 21000 Novi Sad, Serbia, predrag.pecev@gmail.com, predrag.pecev@uns.ac.rs
[2]Preschool Teacher Training and Business Informatics College of Applied Studies "Sirmium", Zmaj Jovina 29, 22000 Sremska Mitrovica, Serbia
[3]University of Novi Sad, Technical faculty "Mihajlo Pupin", Đure Đakovića bb, 23000 Zrenjanin, Serbia, anita.milosavljevic@hotmail.com

ABSTRACT

In this paper we examine and compare several custom-made implementations of a network and port scanner that are written in Python programming language as a part of a Information systems security course teaching material. Network scanning tools play an important role in network information gathering, which is the first step in penetration testing. First version of network and port scanner comes in two sub versions: for Windows and Linux operating systems, and a slightly modified version to address specifics of Android devices. Both varieties of first version use ping command to see if a device with a certain IP address will respond to them and then uses TCP sockets to scan range of ports on a device that responded. Second version of network and port scanner uses ARP Protocol to scan a network for connected devices and then uses TCP sockets to scan range of ports on a device that responded to broadcasted package. In the late section of this paper, we emphasize advantages, disadvantages, performances and ease of usage of previously mentioned implementations. We also discuss ethical dilemmas of using such tools from the aspect of security professionals, programmers, common computer users and IT students.

Keywords: network scanner, ports, TCP sockets, ARP, information gathering.

INTRODUCTION

Network and Port Scanner tool was developed out of several reasons. Mainly, we realized that we need such tool based on a couple of software development projects that required network information gathering. There are similar pieces of software such as Advanced IP Scanner, Angry IP Scanner, nmap etc., however we needed a tool with specific set of features that was platform independent and easily programmable/modable. Another reason for developing aforementioned piece of software is that the first author of this paper teaches a course on Information systems security and one of the topics that is covered by that course is network information gathering which is a first step in penetration testing. At first, developed software was in a form of basic, easy to understand Python scripts and in time it was upgraded to a tool as it stands today. Underlying reason for developing this tool is an everlasting desire for personal growth and a search for knowledge that drives any researcher.

We have chosen Python 3 as a programming language out of several reasons. First of all, Python is a script language and it runs quite fast. Also, it is cross platform therefore it runs on a large variety of operating systems such as Linux and Windows which are our prime target operating systems. It has great support in terms of available libraries and a community that is constantly growing. Python is a programming language that is easy to understand and therefore it is great for IT students which aligned with our second goal that is aimed towards making custom course teaching material. One of the "advantages" of Python as a programming language is that it is "free" of semicolons and curly braces and it that way diverges from a large variety of C-based syntax programming languages that can push away students in the early stages of grasping basic programming concepts. Developed software exists in two official versions, besides one unofficial initial version that consisted of several batch (Windows) and shell (Linux) scripts.

RELATED WORK

In this section of our paper, we will mention several papers that provided us some useful guidelines, principles and ideas that guided us during development of Network and Port scanner tool. Solutions described in (Ahmed, et al., 2008; Ivanov, Ryzhkov, Safronov, Voloshin, & Usachev, 2021; Niedermaier, Fischer, Merli, & Sigl, 2019; Anand, Waghela, & Varghese, 2011). gave us an idea to seek for a solution that would run on multiple platforms and to possibly think about running it on custom hardware as well. (Arzhakov, & Silnov, 2017) provided guidelines for architecture of a multithreaded network scanner which we took into consideration when developing our tool. Paper by authors (Chikohora, & Mogomeli, 2021; Kiggundu, 2019; Skaggs, Blackburn, Manes, & Shenoi, 2002; Kocher, & Gilliam, 2005; Bhuyan, Bhattacharyya, & Kalita, 2011; Pattanavichai, 2017) talk about design and applications of various Network and Port scanners that share several features with our solution. Our tools are similar to tools and solutions that are presented in previously mentioned papers. However, they differ on technologies that where used during development, level of accessibility and number and quality of implemented features (some are better, some are not). Also they differ on a goals they are aiming to fulfill. As previously mentioned our tools where made with a sole goal of making a tool that suits our need regarding projects that we are working on and a tool that can be used as a teaching material on Information systems security course.

NETWORK AND PORT SCANNER – FIRST VERSION

First version consists out of two Python scripts. One script scans the network using ping command and lists IP addresses of all devices (computers) within a scanned network. The other script is used to scan open ports using sockets on a selected computer when IP address is provided. Developed scripts accept command line arguments that specify network and port range respectively. With first version of our tool we strived to use basic commands and tools provided by an operating system and libraries that are distributed along basic Python installation.

With this approach we have achieved a high level of interoperability and made scripts operating system independent. When we presented created scripts as a part of a course material, we concluded that the scripts can be easily red and understood by most of students. However, since we used ping command to scan the network, the first script that scans the network takes considerable amount of time to execute and during the scan address of a device being currently pinged (scanned) is displayed. If device responds to ping command, its address is added to the list of found devices. Also, ping can be disabled via firewall thus disabling network scanning using ping command and rendering our network scanning script useless.

Figure 1 shows results of an execution of a network scanning script. Script was run in Visual Studio Code IDE within a PowerShell terminal on a computer that runs Microsoft Windows 10.

Figure 1. Network scanner – First Version.

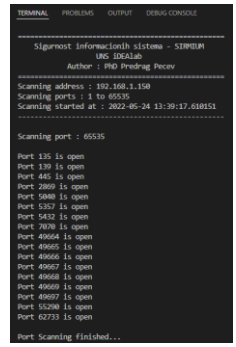

Figure 2. Port scanner – First Version.

Introductory data that can be seen from figure 1 are as follows:

- local IP address of a computer running the scan is 192.168.1.150.
- addresses from 192.168.1.1 to 192.168.1.255 are being scanned (pinged).

- scanning started at 5PM on 24th of May 2022.

When we inspect figure 1 in detail we can see the results of the scan. Scan has found seven devices within a scanned network. Given that we know what devices are connected to our network we can confirm that this method detected all of them. Within tested network topology router's address is 192.168.1.1., addresses of two PC's running Microsoft Windows are 192.168.1.113 and 192.168.1.150 while all other listed IP addresses are assigned to Android phones that are connected via wireless interface. Also since broadcast address 192.168.1.255 responded to ping command it is in a list of found devices but it is not an actual standalone computer on a network.
On figure 2 an output of a port scanner script is shown. Introductory data that can be seen from figure 2 are as follows:

- IP address of a computer being scanned for port is 192.168.1.150.
- Range of ports that is being scanned is 1 to 65535. By default, we use a value range of an unsigned short integer but it can be altered and specified via command line parameters.
- scanning started at 13:39 on 24th of May 2022.

When we inspect figure 2 we can see various open ports on a scanned device:

- Port 135 is commonly used for RPC (Remote Procedure Call), DCE endpoint resolution, Microsoft EPMAP (End Point Mapper).
- Port 139 is commonly used by NetBIOS Session Service.
- Port 445 is commonly used by a SMB which is a file sharing protocol. Initially it ran on NetBIOS and after Windows 2000 it started using port 445.
- Port 5357 is used by Microsoft Network Discovery.
- Port 5432 is used by PostgreSQL server.
- Port 7070 is used by AnyDesk.
- Ports 49664 to 49667 are local ports used by Task Scheduler, Spooler, EventLog, Skype etc.

Based on the analysis of available scanned ports it is clear that scanned device is a PC running a Windows operating system that has file sharing enabled and runs AnyDesk for remote desktop connection. Once we found out about Pydroid3, which is an IDE for Python3 that runs on Android devices, we have slightly modified previously mentioned scripts in order to facilitate proper dynamic hostname lookup. Rest of the scripts remained the same. Output of Network and Port scanner scripts that where run on an Android phone within a Pydroid3 IDE are shown on figure 3 and 4. There are slight differences in output having in mind that scripts are running on a different platform and that some devices where not online when network was scanned.

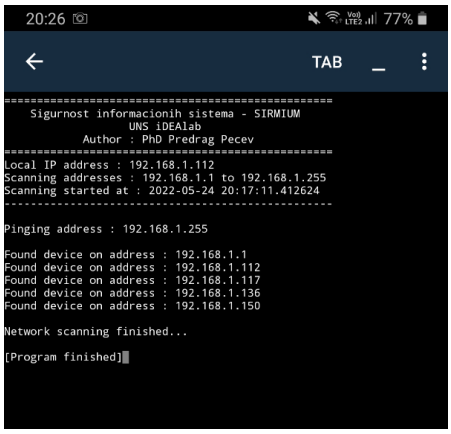

Figure 3. Network scanner – Android Version.

Figure 4. Network scanner – Android Version.

NETWORK AND PORT SCANNER – SECOND VERSION

In second version of our tool we decided to combine two scripts into one and provide more user friendly interface. This meant that we needed to use some Python graphical libraries that provide text coloring and text formatting for a CLI (Command Line Interface) application with emphasis on table formatting so we can have neatly stacked results. We used colorama, PrettyTable, termcolor and pyfiglet libraries to enhance standard command line experience. Therefore, interface of a second version is more user friendly compared to the first version.

Second version of a Network and Port Scanner provide more data about devices that where detected on the network. Compared to the first version additionally it provides:

- Information on MAC address and its vendors.
- Information on types of detected devices by noting that detected device is a router, phone, computer etc.
- Information on device name that was detected, mainly hostname / identifier.

Second version of developed tool has two stages. On figure 5 first stage of our tool is shown. At first stage network is scanned using ARP protocol and 256 requests are being broadcasted. In this particular case, five devices responded to broadcasted packages while there are 251 unanswered requests. Next, developed tool will ask should it proceed with port scanning for all listed devices. Port scanning on any device can be skipped by pressing Space key. Also, by default, only first 1000 are scanned, but that can be altered by specifying proper parameters values via command line.

Figure 5. Network and Port scanner – Second version – First stage.

Figure 6 shows second stage of our tool and end results of network scanning. From figure 6 it can be seen that a user has not interrupted port scanning on any listed detected device and there are 3 mobile phones (Android, iPhone etc..) one computer, and one router connected to the scanned network.

Out of the data from NIC Vendor column it can be concluded that manufacturer of a router is TP-LINK and that manufacturers of detected phones are Samsung, Huawei and HiSense. Also manufacturer of NIC (Network Interface Card) for detected computer is ASUSTek indicating that this computer has an ASUS motherboard with integrated NIC by the same manufacturer.

Out of the data from Open Ports column it can be concluded that within port range of 1 to 1000 there are no open ports on phones that are connected to the network. Router has several open ports: 22 (SSH, SCP, SFTP), 53 (DNS – Domain Name Service), 80 (HTTP) and 443 (HTTPS) while scanned computer has open ports 135, 139 and 445 whose function was previously mentioned and discussed in this paper.

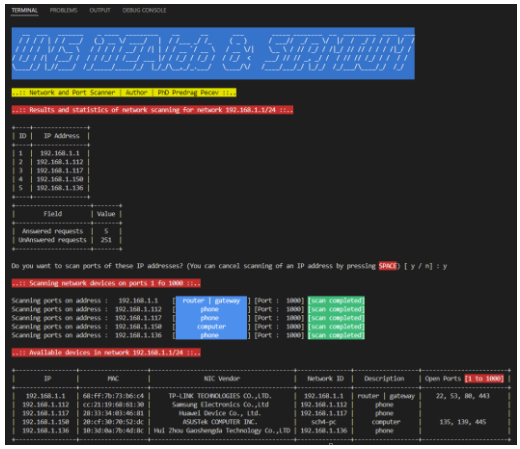

Figure 6. Network and Port scanner – Second version – Second stage.

Regarding network scanning we decided to use ARP (Address Resolution Protocol) to scan for devices on a network. ScaPy Python library provides various methods for packet manipulation and we found it to be very useful regarding our needs. Since we switched from using ping command to using ARP protocol, we have greatly decreased time that takes to scan the network. Using ARP protocol network is scanned within a couple of seconds, while using ping, it takes up to 2 minutes to scan all of the IP addresses in a network.

ADVANTAGES AND DISADVANTAGES OF DEVELOPED TOOLS

In current era, scripts written in Python present a clear advantage over many other solutions due to its accessibility across multiple platforms (as shown in this paper) and syntax readability which can lead towards various modifications and upgrades if the user wills it so. First version of our tool is written using only basic Python libraries thus making it quite easy and straightforward to run. Second version of our tool uses a lot of libraries to enhance user experience and speed up network scanning process thus it requires a bit of an expanded knowledge from a user in order to set it up properly, especially on Windows based operating systems. Having in mind what was previously said, using Python for presented task can be both an advantage and a disadvantage depending on the users standpoint level of knowledge.

As previously said first version relies on ping command that takes time to ping all computers on a network in order to discover them. Also ping command can be blocked via firewall thus rendering our tool unusable. Stated is a clear disadvantage but here we have a tradeoff between ease of access and performance. In a manner of speaking first version says : “I am slower than a second version, and I present less data, but I am easier to setup and run”.

Second version of our tool relies on ARP protocol and works well and fast with IPv4 addresses. However, there is a minor chance that this version of our tool will be rendered unusable if networks start using IPv6 addresses exclusively. NDP (Neighbor Discovery Protocol) is a protocol that will in time replace ARP thus rendering many ARP based attacks on the network obsolete. If networks keeps both ARP and NDP protocol then second version of our tool will still be viable, if not, we will rewrite it to incorporate NDP.

DISCUSSION

When we developed previously described tools we conducted a survey in a form of an interview and a survey. We surveyed Cyber security professionals, Programmers, Common computer users and IT students thus creating four focus groups. There where no mixing of participants between surveyed groups and groups where asked to give their opinions about presented tools and they where presented with an “ethical” dilemma of how they would use presented tools. Since network information gathering is a first step in penetration testing, respondents where asked if they would use obtained network information for good by patching

potential vulnerabilities, or they would use those potential vulnerabilities to do harm. With their responses they denoted themselves as a white, black or a gray hat.

Opinions of cyber security professionals and programmers where quite straightforward where each of them took a stand based on their area of expertise.

Comments from cyber security professionals where that created tools are quite good. Their opinion is that developed tools lack several minor features that they would like to see incorporated within a next version of our solution, but given the intended purpose of created tools they consider them to be highly functional. Programmers analyzed source code of developed tools and stated that system calls, error handling, data parsing and representation are done properly by following proper software patterns.

Opinions of Common computer users where various and mostly depending on their skill level and their general level of awareness of what is happening on the network. Here we present their answers categorized in five groups based on the contents of their responses:

- I do not know how to run this.
- I do not understand what this is used for.
- I do not need this.
- Those scripts are neat and useful, but not for me.
- Nice, so now, in theory, I can hack someone?

Getting an opinion of IT students was the most interesting part of the conducted survey. Their responses where various and mostly depending on their skill level and their general level of awareness of what is happening on the network. Here we present answers they gave categorized in six groups based on the contents of their responses:

- I do not understand this.
- I am not interested in this.
- Will this be on an exam?
- So now, when we know how to get IP addresses and detect open ports what can we do next?
- I am going to expand this script (second version)
- The script can be used in any computer network, right?

When we further analyzed opinions of aforementioned four groups, we noticed that there is an obvious difference in opinions between |Cyber security professionals and Programmers compared to opinions of Common computer users and IT students. Cyber security professionals and Programmers based their opinions from the standpoint of their respective professions and did an in-depth analysis of a presented tool which yielded some interesting advices that can be adopted as a valuable contribution to the developed of our tools. Common computer users and IT Students took a bit of a "relaxed" standpoint that was influenced by their current interests and needs.

When presented with an ethical dilemma of how will they use provided tools and data they obtain using them each group had some interesting opinions and answers.

Cyber security professionals had no ethical dilemmas whatsoever or there might be something they are not telling. They know exactly how the created tools work and what service they provide. They say that, should they use them, they would use them for good e.g. penetration testing to determine network vulnerabilities in order to sanitize them not to exploit them.

Programmers took a neutral stand regarding usage of created tools with slight tendencies towards usage for good since every programmer that produced any piece of software that dwells online wants its software as secure as possible.

Common computer users and IT students had various responses aligning with positive, negative and neutral outcomes. Some Common computer users had malevolent intention but most of them stated that they would use created tools for good (about 72%). Slight percentage of them where unsure (17%) while the rest where the "bad boys and girls". Regarding answers that IT students provided 15% of them where unsure, 14% where the ones aligning with negative

outcomes while 71% where the "good boys and girls". It is very interesting that ratios among given outcomes are very similar when comparing aforementioned groups.

CONCLUSIONS

Information is power – a statement well known and easily applicable regarding this topic. Given that created tools are intended to be used as an information gathering tools it is up to the user to decide how gathered data will be used. Created tools can provide valuable data for penetration testing specialists so they can patch up any potential vulnerabilities in network security, but it can also provide valuable information to a "black hat" that is up to no good. Emphasis of a decision is on the person using those tools and fortunately, so far, based on the conducted survey, the good guys are wining.

Regarding further development of a presented tool, we will most likely accept advices given by Cyber security professionals and expand our tools by adding suggested features. Also, since presented tool is a part of Information systems security course teaching material, a part of second version of our tool was expanded on and integrated as a part of a Python script that runs ARP Spoofing / ARP Poisoning attack, also shown in aforementioned course. Everything shown as a part of course on Information systems security was done for educational purposes only and it was done on a local network that I own without any malicious intent.

LITERATURE

Ahmed, N., Khalib, Z. I., Ghossoon, W. M., Ahmed, R. B., Sudin, S., & Asi, S. M. (2008). Embedded Port Scanner (EPSS) System using linux and Single Board Computer. *In Proceedings International Conference on Electronic Design* (pp 1 – 5).

Anand, T., Waghela, Y., & Varghese, K. (2011). A scalable network port scan detection system on FPGA. *In Proceedings International Conference on Field-Programmable Technology* (pp 1 - 8).

Arzhakov, A. V., & Silnov, D. S. (2017). Architecture of multithreaded network scanner. *In Proceedings 18th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)* (pp 43 - 45).

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2011). Surveying Port Scans and Their Detection Methodologies. *The Computer Journal , 54*(10), 1565 - 1581.

Chikohora, E., & Mogomeli, L. (2021). A Study on the Impact of Network Vulnerability Scanners on Network Security. *In Proceedings 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)* (pp. 1 - 4).

Ivanov, A. A., Ryzhkov, A. K., Safronov , B. A., Voloshin, A. A., & Usachev, S. S. (2021). Development Of A Mobile Network Scanner Of Information Flows With Support For Protocols Of The IEC 61850 Standard. *In Proceedings 4th International Youth Scientific and Technical Conference on Relay Protection and Automation (RPA)* (pp 1 - 7).

Kiggundu, J. (2019). Advanced considerations for defensive cyber products with regards to network security and enterprise integration capabilities. *In Proceedings IEEE Integrated STEM Education Conference (ISEC).*

Kocher, J. E., & Gilliam, D. P. (2005). Self port scanning tool: providing a more secure computing environment through the use of proactive port scanning. *In Proceedings 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)* (pp 139 - 143).

Niedermaier, M., Fischer, F., Merli, D., & Sigl, G. (2019). Network Scanning and Mapping for IIoT Edge Node Device Security. *In Proceedings International Conference on Applied Electronics (AE)* (pp 111 - 116).

Pattanavichai, S. (2017). Comparison for network security scanner tools between GFI LanGuard and Microsoft Baseline Security Analyzer (MBSA). *In Proceedings 15th International Conference on ICT and Knowledge Engineering (ICT&KE)* (pp 85 - 91).

Skaggs, B., Blackburn, B., Manes, G., & Shenoi, S. (2002). Network vulnerability analysis. *In Proceedings 45th Midwest Symposium on Circuits and Systems, 2002.* MWSCAS-2002.