Implementation and Vulnerability Test of Stealth Port Scanning Attacks using ZMap of Censys Engine

Seungwoon Lee[1], Sun-young Im[2], Seung-Hun Shin[3], Byeong-hee Roh[2] , and Cheolho Lee[4]
[1]Dept. of Software, Ajou University, Suwon, Korea
[2]Dept. of Computer Engineering, Ajou University, Suwon, Korea
[3]University College, Ajou University, Suwon, Korea
[4]The Attached Institute of ETRI, Daejeon, Korea
{swleeyg, hangeul, sihnsh, bhroh}@ajou.ac.kr, cheolholee75@gmail.com

Abstract— **Port Scanning is one of the widely used attacks to collect information of devices connected to the Internet. The information may be very useful for system administrator to assure security, but can be a bridge for malicious users not only to find vulnerabilities of the devices, but also to consider subsequent attacks. There are several search engines, such as Shodan and Censys, to build and provide searchable databases of identified devices and networks. In this paper, we implement possible stealth port scan attack tools using ZMap used in Censys search engine, and test the vulnerability of devices in an area of our university using the implemented tools. It is shown that though stealth port scanning attacks except SYN scan have been generally known as they are ineffective, many vulnerable systems are still on services and able to be scanned by the attacks.**

***Keywords*—Port scans, ZMap, Censys, Shodan**

I. INTRODUCTION

As Internet of Things (IoT) services are popularly emerged in real life, the number of devices connected to the Internet is increasing too much and fast. To manage the devices, it needs to consider a trade-off between the usability and the security on those devices. To assure the usability, information of devices need to be shared with others. The information share and disclosure may cause severe vulnerabilities to various cyber attacks.

In a port scanning, which is one of the widely utilized attacks to collect system's information, an attacker sends a message to each port of a system, and analyzes the response from it. Then, the attacker can identify states of ports, services, and operating system (OS) of the system, which can be critical information for further attacks to it. Due to such vulnerabilities, firewalls and Intrusion Detection Systems (IDSs) have functions to detect and block the scanning attacks [1].

There are several search engines, such as Shodan[2] and Censys[3], to build and provide databases on the information of identified devices by using port scanning mechanisms. Unlike Shodan's closed policy, Censys opens its framework and related sources. The ZMap[4], which is one the components used in Censys as similar as Nmap[6], provides open-sourced port scanning tools. Stealth port scanning attacks have been proposed to get hosts' information without leaving the logs on the security systems [5]. Nmap and ZMap provide sources for the development of various stealth port scanning mechanisms.

In this paper, we implement several stealth port scan tools utilizing ZMap, and build a Censys-like environment to get the information of hosts and network devices in an area of our university. With the implemented tools, we also test the states and vulnerabilities of the hosts and network devices.

The remainder of the paper is organized as follows. Section 2 describes briefly backgrounds on network scans, and Section 3 explains the implementations using ZMap. Section 4 gives experimental results, and Section 5 concludes the paper.

This work was supported partially by the National Security Research Institute.

II. BACKGROUNDS

A. Network Scanning Tools

Network scanning tools are used to detect devices connected to the Internet, and to identify their useful information such as ports' states and related services, OSs, and so on.

1) Nmap

Nmap [6] is a free and open utility for network discovery and security auditing. It uses a raw IP packets to identify what hosts are available on the network, what operating system and services they are running, and so on. It supports many port scanning mechanisms such as TCP SYN, TCP connect, UDP, and others. It can be used on both a graphical and console versions.

2) ZMap

ZMap [4] has been designed to perform comprehensive scans of the IPv4 address space or large portions of it. It is possible to perform a complete scan of an IPv4 address space in under 5 minutes. It supports various port scanning methods such as TCP SYN, UDP, and ICMP echo as similar as in Nmap. ZMap can find more listening hosts than Nmap despite running hundreds of times faster. However, it cannot obtain information about the operating system unlike Nmap.

B. IP Device Search Engine

The IP device search engines, also known as IoT search

engines, collect the information of devices connected to the Internet. The search engines provide users the collected information of the devices in a variety of ways. There are two representative search engines, Shodan and Censys.

1) Shodan

Shodan [2] collects information about 500 million devices per month. Users can search the detailed information by using filters defined by Shodan, such as a country, operating system, port, and so on. Users can only see 10 of search results without account and 50 of search results with account. To view all results, users must submit information for the purpose of obtaining the results and pay. Shodan also provides services such as Map, Exploits, Scanhub as well as search results.

2) Censys

Censys [3] is a search engine that allows computer scientists to ask questions about the devices and networks that compose the Internet. Censys collects data on hosts and websites through daily port scan of the IPv4 address space with ZMap. It offers the collected information using the Google cloud Platform. Censys is limited to the scanning target and services, but it is attempting to expand through open source.

C. Stealth Port Scanning Attacks

Stealth port scanning refers to any scan that bypassing filter, firewall, router and behaving as casual network traffic. There are several stealth port scanning mechanisms as following.

1) SYN Scan

SYN Scan is a well-known scan and used as a default for major network scanning tools. It is also known as 'half open scan' because it does not complete the three-way handshake. A port can be determined as open when ACK is received, whereas close when RST is received.

2) FIN Scan

A FIN scanning module creates and send a segment with the FIN flag set. The expected behavior on it is defined in RFC 793, in which any TCP segment with an impossible flag to an open port is discarded, whereas a RST response is made for closed ports. That is, the sender will receive RST segment for the FIN segment to a closed port, while no response for open ports. However, it depends on OSs. For example, Windows OS returns ACK segments regardless of that ports are open or not. Further, most of stable firewalls ignore FIN packet.

3) Xmas Scan

In Xmas scan, a sender sends a TCP segment with all flags set. Similar to FIN scan, the receiver may return a RST segment for a closed port, whereas it will ignore the segment for an open port. Because of its violation of TCP rule, Xmas scans are easily detected by almost all IDS, and do not work.

4) NULL Scan:

On the contrary with Xmas scan, a TCP segment is sent with no flag set in NUL scan. It may result in the target host responding with a RST segment for closed ports, but nothing for opened ones. NULL Scan, similarly with Xmas scan, is not effective anymore.

5) ACK Scan

Unlike other scans, when a receiver receives segments with ACK flag set, it checks filter configurations, which define whether it responds or not for the segment, rather than port's state. Accordingly, it is not able to detect whether the port is open or not by observing the response. Both open and closed ports returns RST packet when the unfiltered system is scanned. Filtered system drops the ACK packet and responses nothing.

III. STEALTH PORT SCAN IMPLEMENTATION WITH ZMAP

As mentioned before, ZMap provides sources for the development of various stealth port scanning mechanisms. Here, we implement some well-known stealth port scanning mechanisms with ZMap. SYN scan is implemented as the default scan in ZMap, while other scans are not.

In order to add a new function easily, ZMap structure is modularized with several modules, as shown in Figure 1, which are responsible for filling in the body of probe packets.

Figure 1 shows the module view of ZMap. There are three thread modules running on ZMap: `send`, `recv`, and `monitor`, for sending, receiving, and monitoring segments, respectively, as shown in Figure 1. The `probe` module is called by `send`, and it has specific probe modules including detailed rules based on protocols and scanning methods. We add the stealth port scan mechanisms on specific probe modules by modifying it. With the modularity and extensibility of ZMap structure, the stealth port scanning modules can be added by changing small parts of TCP headers.

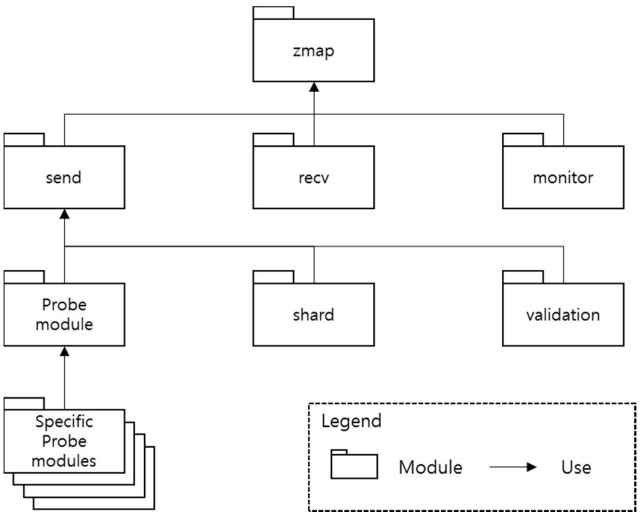

Figure 1. ZMap Module View

Four probe modules are added according to mostly used stealth port scanning attacks which is mentioned in Section 2. Each probe module is implemented by developing and registering the necessary callbacks in the structure. In the each probe module, there are six callback functions which are *probe_inintialize()*, *thread_initialize()*, *make_packet()*, *validate_packet()*, *print_packet()*, and *process_packet()*. We modified *make_packet()* in order to change the TCP flag set.

To verify the implementation, we used Wireshark installed on Both sender and receiver to capture the packet and confirmed the transmission of packets we sent. We checked if the flag of

captured packet is right as we intended according to each scanning method and the implementation was confirmed by its result.

IV. EXPERIMENTAL RESULTS

To implement the stealth port scanning tools, we used the ZMap 2.1.1 source, which is written in C, on Ubuntu 16.04.1. To test the implementation, we applied it to the hosts in some area of Ajou University. Even though it is stealthy, since the scanning is still a kind of cyber-attack, the permission of the scanning was granted by university computer center of Ajou University. The number of hosts scanned is more than 12,000 with two ranges of IP Addresses. We scanned well-known ports such as 23 (telnet), 25 (smtp), 80 (http), 443 (https), 525 (printer). Especially, we scanned 25565 port which is used for game 'Minecraft' because this port might not be used much in the university network.

Table 1 shows the result of the experiment. The number of responses by FIN scans ranged from 235 to 584, about 3% of total scanned hosts. XMAS and NULL scans did not have any response from the hosts. There were 19 responses for ACK scans regardless of port numbers. The detailed analysis on the results is as follows.

A. FIN scan

From 235 to 584 hosts made responses to FIN segments with RST ones. As we can see from TABLE 1, the results on different ports show different result. Generally, firewalls running on Windows OS blocks FIN scan segments. However, it is observed that when the firewall is disabled, RSTs for Fin segments are being sent. That is, whether fire wall is abled or not is checked by using FIN scanning attack, because most of Windows firewall is consistently updated against abnormal Fin packet. On the other hand, for Linux OSs including VMWare and Apache, FIN scans still work. Hosts with closed ports' scans responded RST segments. With that, the FIN scan results can be considered as a sum of the number of hosts with Windows not running firewalls and Linux OSs with closed port.

B. NULL and Xmas scan

As expected, no responses returned for Null and Xmas scans. Even firewall was deactivated, nothing was still answered from the scans. From the results, it can be seen that the scans do not work in normal environments.

C. ACK Scan

As mentioned in Section 2, only the hosts without filter configurations send RST segments back. In normal operation environments, hosts configure the filters.

However, it has observed that some hosts returned the RSTs in our experiments, and they were devices such as printers with old embedded web server. Figure 2 shows the result that we accessed to a printer, which responded RSTs for ACK scans, with the old and vulnerable web server through the browser. The devices were made in from the middle of 2000s by well-known enterprises. In addition, we succeeded test prints on the printers, which may cause the resource depletion attack against the printers and users who utilize them. In addition, most of printers with the vulnerable web browsers were still using and they set the administrator's password as default, which can be easily find out on the Internet.

TABLE I. STEALTH PORT SCANNING REUSLT

Port Number	Number of response from the hosts			
	FIN Scan	*XMAS Scan*	*NULL Scan*	*ACK Scan*
23	513	0	0	19
25	359	0	0	19
80	235	0	0	19
443	375	0	0	19
515	370	0	0	19
25565	584	0	0	19

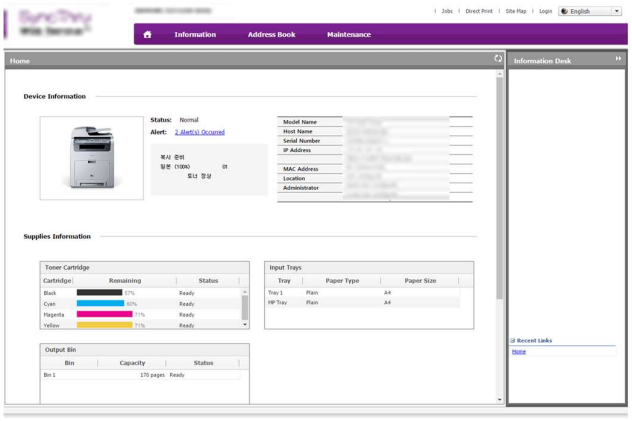

Figure 2. Access to the Web Server of the Vulnerable Printer

V. CONCLUSION

In this paper, we implemented stealth port scans, such as FIN, NULL, Xmas, and ACK scans, using ZMap open source, which is a part of Censys search engine, and tested them to actual hosts in Ajou University. With the scans, it was observed that they can be used to identify hosts' basic running information such as port states, services, configurations, and OSs. Especially, with ACK scans, the vulnerable printers with old web server, for example printers were identified. Stealth port scanning attacks except SYN scan were generally known that they are ineffective. However, many vulnerable systems are still on services and able to be scanned by various scan mechanisms. As a future work, it needs to develop how to detect and protect systems from the search engines.

REFERENCES

[1] E. Bou-Harb, et al., "Cyber Scanning: A Comprehensive Survey," IEEE Comm. Surveys & Tutorials, Vol. 16, No. 3, 3rd Quarter 2014.

[2] Shodan, Available: https://www.shodan.io/

[3] Z. Durumeric, et al., "A Search Engine Backed by Internet-Wide Scanning," ACM CCS' 2015, Oct. 2015

[4] Z. Durumeric, et al., "ZMap: Fast Internet-wide scanning and its security applications, " In 22nd USENIX Security Symposium, Aug. 2013.

[5] J. Gadge and A. Patil, "Port Scan Detection," IEEE International Conference on Networks, Dec. 2008

[6] G. Lyon, "Nmap Network Scanning: The official Nmap project guide to network discovery and security scanning," Insecure, 20