IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 4, August 2018
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

A Network-Layer Intrusion Prevention System for a Metasploit Application Attack

Abdulaziz Almehmadi

Department of Information Technology, University of Tabuk
Tabuk, Saudi Arabia

Abstract - Network and software related attacks have become a trend that is occurring more frequently in today's corporate and private networks. Tools and methodologies developed for ethical hackers and penetration testers that are used to discover vulnerabilities are being used maliciously by individuals and organizations for such purposes. Because Metasploit is freely available to the public, the exploits it provides are known to be used maliciously. Therefore, in this paper we contribute by implementing Intrusion Detection System (IDS) that detect a specific exploit that is intended to vulnerability in software called Icecast. Then we implement an Intrusion Prevention Systems (IPS) to prevent the attacker from gaining access to the system that has the Icecast service running. The detection and prevention from the attack are done from the network layer by analyzing the behavior of the packets, create a signature to detect the attack and then prevent it.

Keywords - *Intrusion Detection System, DDoS, SYN Flood.*

1. Introduction

The free open-source exploitation framework called The Metasploit Framework (MSF) is an environment where exploits can be created and used to compromise vulnerable machines. It speeds up and makes the process of creating new exploits less work intensive by offering many reusable libraries of code and applications to simplify the development process [1]. Furthermore, because Metasploit includes numerous pre-created exploits, it can standardize the usage patterns of exploits. In 2013 -2014, exploits were created differently with different probability of success. However with Metasploit, standards for how exploits are created and used have been developed.

Metasploit was developed in 2002 as a network security game by four core developers. It has since turned into the largest Ruby project in the world. It is now an open-source exploitation framework. Its main uses are for penetration testers, exploit developers and researchers. The Metasploit Framework is an environment which provides its users with the ability to launch exploits against vulnerable systems, and then provide them with a means to do post-exploitation tasks. These tasks can involve uploading or downloading files, gaining shell access, monitoring the attacked system, or other tasks.

There are several other tools which also offer to do similar tasks as Metasploit. Core Impact and Canvas, are examples of two such tools. There are several differences between them. First, Metasploit is free, while the other tools are commercial and paid-for applications. The other main difference is that the Metasploit framework is aimed at users who are developing exploits for operating systems, and applications. The other tools are mostly applications, which exploit vulnerabilities that are already known by the companies behind the applications. The framework also gives the user the ability to do the post-exploitation tasks, which the other tools do not. The other tools have a set exploit and payload which limits the amount of operations the user can do with the tools.

The Metasploit framework has been developed to operate successfully on Windows, Linux, UNIX and Mac OS X. It has also been ported to numerous other platforms as well; an example of this would be its operation on the Apple iPhone/iTouch devices. Due to several differences in the numerous operating systems that Metasploit works on, some of the GUI (graphical user interface) functions do not operate fully on all operating systems. Therefore, in the interest of best operation the user should use the command line interface, a liveCD Linux distribution with Metasploit built in, or a virtualization environment such as VMware with a supported Linux distribution can be used.

In this paper, Metasploit's arsenal is explained in Section II. In Section III, the main framework directory interfaces and modules are introduced followed by the Meterpreter in Section IV. In Section V, the intrusion detection literature review is provided. Furthermore, in Section VI vulnerable software ―Icecast is exploited using Metasploit and an

algorithm and implementation of detection and prevention of the attack from the network layer is explained in Section VII. Finally the conclusion is provided in section VIII.

2. The Metasploit Arsenal

There are several different sections in the Metasploit arsenal for the user to use: exploits and payloads. Exploits are pieces of codes that take advantage of vulnerabilities in a target system in order to run a payload. Payloads are pieces of codes that do a specific task on a target such as getting command shell or taking a full control of the target system.

Separating exploits in Metasploit from payloads gives users the ability to mix and match. Thus for an individual vulnerability in a system, the Metasploit users have the ability to choose the kind exploit and payload combination they desire. All of these choices can be made through any of the interfaces that the users can work with, such as the GUI oriented or by using the Meterpreter (discussed in section IV).

The Metasploit user first chooses the desired interface; selects an exploit and a payload; configures different options for the exploit and payload, and finally uses Metasploit to lunch the exploit at a target system.

New vulnerabilities are found and released regularly as modules by security researchers and users are able to use them by updating their version for Metasploit. In addition, new payloads are released to create new capabilities usable by the exploits already included in Metasploit.

To view the exploits and payloads that are included in the current version that a user has, s/he can go to the directory framework –VersionNumber
Then we use: ./msfconsole
To view exploits type: show exploits
To view payloads type: show payloads

There are currently two versions of the Metasploit framework which are currently being supported, version 2 and version 3. If the user is just installing Metasploit for the first time, it is recommended that they use version 3. Version 3 is a complete re-write of the code of version 2. It is written in the Ruby coding language, which makes it very easy to modify, change, and develop future exploits/payloads. Version 2 is the older version, but it is kept updated as it has various plugins, which may not work with version 3.

3. Main Framework Directory, Interfaces and Modules

There are five main sections which comprise the framework. The sections are: documentation, user interfaces, modules, exploits creation tools, and other tools. In this section the breakdown of each section and a basic understanding of the purpose of each section of the framework are given.

3.1 Documentation

This section of the framework gives detailed information about each application, vulnerability, exploit, and tool in the framework. Each aspect of the framework provides a detailed explanation as to the definition and the function of the specific part. The documentation parts of the framework can be accessed through all of the interfaces, but the easiest way is through the command line interface. The user can simply add a –h to any command and Metasploit will give instructions about its use. Another command that can provide information about the use of a tool is the "show options" command. This command will display any possible options that could be set in the current configuration and information about how the function can be utilized.

3.2 User Interfaces

As mentioned earlier in this paper, there are several different interfaces that can be used to interact with the Metasploit framework. The following is a breakdown of the built in interfaces:
1- Msfconsole – This is the most commonly used interface, it is solely command line driven [3].
2- Msfd – This daemon listens on TCP port 55554 that offers msfconsole to whoever connects to it. This is used when Metasploit is used in a team environment. It would be set up on a centralized system which all of the users would connect to it, instead of having it installed on their own workstations [3].
3- Msfcli – This is a built in command line interface for users to creating scripts that can be run in the Metasploit framework [3].
4- Msfgui – This is a ruby-based GUI application which is no longer supported. Rapid7 has created a new commercial enterprise level GUI application to replace this interface [3].
5- Msfweb – This interface creates a web server on TCP port 55555 that the user's web browsers can connect to it in order to operate Metasploit [3].

3.3 Modules

This is a directory in the Metasploit framework that contains the most important tools, exploits and payloads that are used in most of the attacks which Metasploit is used for.

1- Auxiliary - This section holds that miscellaneous items, vulnerability checks, denial of service tools, Exploits that don not have payloads, and tools which do not necessary exploit a system (such as information gathering tools) [3].
2- Encoders – These functions assist users to convert the exploit and payload code into a different form in order to bypass firewalls and IDSs [3].
3- Exploits – There are hundreds of built in exploits in Metasploit, and they are updated daily with new ones. The exploit modules are classified by platforms (OSes) and then by types (protocols) [3].
4- Nops – This is a tool which is used in the creation and development of exploits for use with Metasploit. [3].
5- Payloads – This section contains the numerous payloads that are possible to use with various exploits. These payloads include: Meterpreter, Shells, Backdoors, Telnet, etc. [3].

3.4 Payloads

Payloads are generally small pieces of code that are executed on the targeted system as part of an exploit. A payload is usually a sequence of assembly instructions that helps achieve the attacking user's specific post-exploitation objective. Some of the common types of objectives are: adding a new administration level user to a remote system, or launching a command prompt and binding it to a local port so that it can be connected to at a later time. The following sections break down some of the main payloads that are built into the Metasploit Framework.

3.4.1. Singles

Singles are stand-alone payloads that have their functionality and communication bundled together.
1- Adduser – Used to add a user [3].
2- Exec - Used to execute commands [3].
3- Download_exec – Used to download a file using http and executes it on the target computer [3].
4- Shell_bind_tcp - TCP shell listener [3].
5- Shell_bind_tcp_xpfw – Command to shut off windows firewall and start a TCP shell listener [3].
6- Shell_reverse_tcp – Starts a reverse shell back to attacker, used if there is a stateful firewall on the network [3].

3.4.2. Stagers

Stagers are parts of a total payload. They are used when the exploit that is being utilized cannot load the entire exploit and full payload at once. They allow for the payload to load the first part and then allow a later stage to communicate with the attacker in numerous flexible fashions.

1- Bind_tcp - Listens on a TCP port for new connections.
2- Findtag_ord - Use the existing TCP connection that exploit was delivered over.
3- Reverse_tcp - Make a reverse connection from target back to attacker.
4- Reverse_ord_tcp – Make a reverse connection using ws2_32.dll already loaded into memory of an exploited process.
5- Passivex - Run an ActiveX control in IE for reverse HTTP communications.

3.4.3 Stages

Stages are payload piece-parts that implement a function, but communicate using an already-loaded stager.
1- Dllinject - Inject arbitrary DLL into target memory.
2- Upexec - Upload and run an executable.
3- Shell - Windows cmd.exe shell.
4- Vncinject – A virtual network creating remote GUI control. 5- Meterpreter – A flexible specialized shell environment.

3.5 Exploit Creation Tools

There are numerous different types of tools included in the Metasploit Framework for creating new exploits as well as for discovering new vulnerabilities in operating systems and applications. These tools help in the development of new exploits for applications. Some of the information which the tools can provide to the exploiter is to find the necessary length of possible buffer overflows. It can do this by automatically creating a string of characters X length long, and the string has a changing pattern [2]. When the user crashes the program that they are testing with a buffer overflow, they can copy the text from the stack, and use Metasploit to find that text in the premade string [2].

This simplifies the task of finding the length needed to overflow the buffer, so that the exploit command is written into the correct memory address. There are also several other tools which are present in the Metasploit Framework, each with a specific type of exploit that it helps develop.

3.6 Other Items: Libraries and Plug-ins

There are also numerous other items which are included in the Metasploit Framework. There are a number of API libraries which can be used in the development of Exploits, Payloads, or separate application development (such as a click and go exploit application). However, due to the fact that the Framework is open-source, anyone is able to develop new plug-ins for the application. Plug-in gives the Framework additional features, which can enhance the use of the Framework.

4. Meterpreter

Meterpreter is one of the payloads which can be used in a post-exploitation attack. Its name is short for the Meta-Interpreter. When a user is attempting to exploit a remote system, they have a specific objective in mind. Typically the objective is to obtain command shell access of the remote system, and thereby run arbitrary commands on that system. The user would also like to do this in a stealthy manner, as well as evade any Intrusion Detection Systems (IDS). The user usually only has one shot at launching a command shell or running an arbitrary command before an IDS puts out an alert and/or a system administrator notices that an attack is occurring against one of their systems.

The Meterpreter is not simply a payload, but rather an exploit platform that is executed on the remote system. The Meterpreter has its own command shell, which provides the attacker with a wide variety of activities that can be executed on the exploited system. The Meterpreter shell is injected into the memory of the running process [2]. Thus, it avoids detection by Host IDS (HIDS) as well as bypasses the limitations of the operating system's native command shell. When the Meterpreter is running on the exploited machine, there is no new process shown running in the Task Manager. This is because Meterpreter runs by injecting itself into the vulnerable running process on the remote system once exploitation occurred [2]. The user can also add Meterpreter to another process once the exploitation has occurred, so that it is even less likely to be found. All commands run through Meterpreter also execute within the context of the running process. In this manner, it is able to avoid detection by anti-virus systems or basic forensics examinations. A forensics expert would need to carry out a live response by dumping and analyzing the memory of running processes, in order to be able to determine the injected process [2]. This is different if shell access is gained directly, as it will be shown in the Task Manager, even if the window is hidden from the user [1].

It also provides complex and advanced features that would otherwise be tedious, if possible at all, to implement purely in assembly. It can dump the SAM hashes without loading any type of program to do so, which keeps it hidden from antivirus, and blocked by permissions. Another function that it provides is the ability to lock the keyboard and mouse of the exploited machine, so that the user cannot physically do anything while the Meterpreter is operating.

5. Literature Review

5.1 Intrusion Detection

Intrusion Detection Systems (IDS) whether host-based or network-based are systems that are responsible for monitoring and reporting security events. Network-based techniques are preferred as not all hosts can afford the resources for installing host-based IDS. Network-based IDS analyses traffic/packets and reports security events once detected. The network-based IDS are deployed inline on the network and passively collect packets for analysis. IDS that can be used to detect attacks in networks can be signature-based systems or anomaly-based systems. Since networks follow a predictable communication flow model as stated by [4] and [5], we believe anomaly-based IDS systems will have a higher probability of detecting the attacks if trained correctly. Further work done by [6,7,8 and 9] discuss various approaches and difficulties in signature based intrusion detection system as well as providing information regarding the usage of SNORT and NESSUS

5.2 Anomaly-based Intrusion Detection Systems

Anomaly-based Intrusion Detection Systems detect abnormal protocol behavior and network traffic by classifying the packets as malicious. Heuristics algorithms are usually used for the classification [10] and [11]. Heuristics algorithms compare traffic to a predefined baseline. Once an abnormal behavior is detected, the IDS reports the incident for further processing. Defining a correct baseline is a challenging task; however, specifically to the static nature of networks, the task becomes simple. The baseline defining heavily depends on capturing traffic for a period of time defined by months. This will allow a classifier learn the normal behavior of the network. Anomaly-based IDS and the static nature of networks make the development of an effective IDS possible.

Other intrusion detection systems exist including Flow-based intrusion detection where network flow can be predicted and anomaly in the flow can trigger an alert of an intrusion [12]. Finally state-based intrusion detection where a normal state of a network is baselined and a

change in the state of a network protocol triggers an alert [13,14,15].

6. Icecast Exploitation

Icecast is a free application that is used as a streaming media application that was released for the first time in 1998, version 2 of this software was released in 2011. The Icecast server is capable of streaming content as Vorbis over standard HTTP [16].

Vulnerability in the Icecast2 application was discovered and an exploit was written to take advantage of it. The exploit is available in the Metasploit Framework, and any payload, a way of interacting with the exploit, can work with it.

This section will guide the reader in the necessary steps to install, configure, and execute the exploit in the Icecast2 application with the use of the Metasploit Framework.

6.1 Preparing the Environment

Download and install the following with giving the IP addresses to both Victim and Attacker as specified:
Victim 192.168.1.10/24:
Icecast2 from http://www.icecast.org/download.php
Attacker 192.168.1.30/24:
Metasploit Framework 3.2 from http://www.metasploit.com/
Firewall 192.168.1.20/24

6.2 Preparing the Attack

On the target machine, the user should first run Icecast2 and click on start server.
For attacker:
1- First lunch Metasploit the console interface by moving to Metasploit directory and then typing:
./msfconsole

2- Choose the exploit; in this case it is the Icecast_header exploit by typing:
USE exploit/windows/http/icecast_header

3- Then set the payload as desired. With this example the bind_TCP payload will be used.
SET PAYLOAD windows/meterpreter/bind_TCP

4- Set the remote host that is running icecast to exploit it by typing:
Set RHOST 192.168.1.10

5- Type “exploit” to exploit Icecast and make it listen to a TCP connection and connect to it.

Now the attacker is connected to the target machine through Meterpreter, a user can type sysinfo to get information about the system. All of the commands that are available to be run from Meterpreter can be used now. To gain shell access to the target machine, the following command needs to be entered and run:
execute –f cmd.exe –c –H –i

This command tells the targeted computer to start the command prompt, in a channel, hidden from the current user, and make it interactive with the attacker.
Now the attacker has full shell access and can migrate to any running process as required by typing ps to get the running processes with each one’s Id, Then type migrate [the process id].

7. Detection and Prevention Implementation

Once the attack is launched successfully, it is in the best interest of security experts to secure it against future exploitation by malicious attackers. One solution is to ask for a patch from the application developer or removing the software and getting another one that has the same functionality and no vulnerability. It is possible to create a specialized method to detect and prevent the attack since all of the tools and steps necessary to exploit the vulnerability are known.

Therefore, the first step to detect the attack is by launching it and capturing all the packets exchanged in-between both systems. The second step is to analyze the behavior of the packets and the sequence of them in order to detect a specific signature. Third step is by lunching the attack several times to detect if the sequence of packets change or stay the same. The forth step involves writing a signature to detect the attack. Finally after the attack is detected, we implement a way to prevent it.

After going through the steps 1-3, it has been concluded that the packets do not change and a specific signature can be found to detect it.
1- Capture a packet from the network interface.
2- Get the source and destination IP addresses, ports and the packets flag.
3- If seven conditions are true, trigger an alert that an Icecast Metasploit attack is detected and then prevent it.

The seven conditions that have to be found in the first set of packets are as follow:
1- SYN–Dst port 8000-Dst IP Victim IP

2- SYN-ACK–SRC port 8000-SRC IP Victim IP
3- ACK-Dst port 8000-Dst IP Victim IP
4- RST-ACK–SRC port random number-SRC IP Victim IP
5- SYN–Dst port Same random number-Dst IP Victim IP
6- FIN-ACK–Dst port 8000-Dst IP Victim IP
7- PSH-ACK–Dst port 8000-Dst IP Victim IP
The pattern of packets is shown in figure 1.

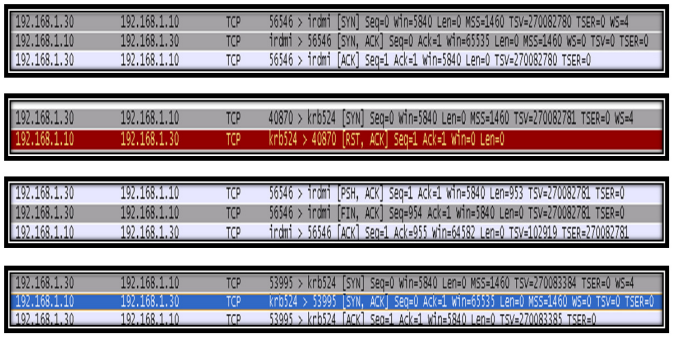

Fig. 1 Pattern of packets when sending the exploit and payload.

The Metasploit Icecast exploit detection and prevention algorithm is shown in Figure 2.

Once all of the seven conditions are true an alarm is raised that the attack is detected as shown in figure 3. The prevention mechanism is by killing the Icecast process by injecting the command taskkill /IM Icecast2.exe. The command will search for the name Icecast2.exe in the running processes and kills it once an attack is detected as shown in Figure 4.

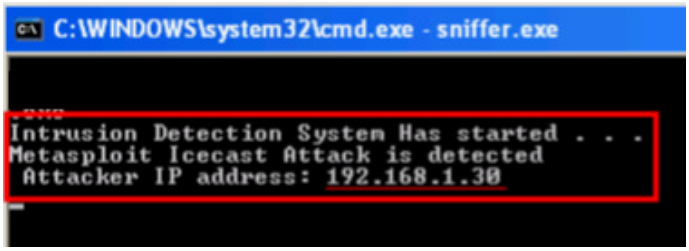

Fig. 3 Metasploit Icecast attack is detected

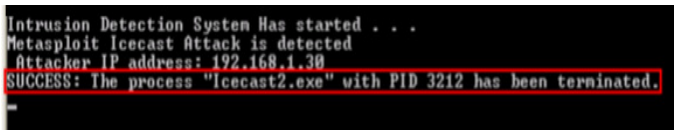

Fig. 4 Metasploit Icecast Attack is prevented

Although the prevention technique, killing Icecast process, prevents the attack from completing successfully, it also causes a denial of service (DoS). This is especially important, as Icecast is a casting server; its main purpose is for it to be available to users to connect to in order to listen to an ongoing webcast. In order to prevent the attacker without (DoS), Attacker's IP address will be recorded and the following will be done:
1- Once the attack is detected, Attacker's IP address will be recorded.
2- Putty application will be executed to get SSH to the Firewall that sets in-between the Icecast server and the attacker's machine.
3- The Attacker's IP address will be prevented from accessing the private network.

The injected code into the firewall is the fowling and the result is shown in Figure 5.

iptables -A FORWARD -d Icecast server's IP –s attacker's IP -j REJECT

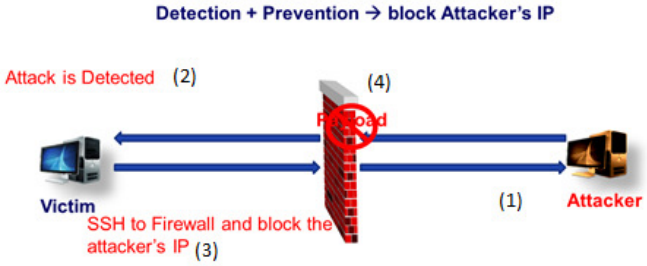

Fig. 5 the attacker victim detection and prevention

7. Conclusion

There are numerous good uses for the Metasploit Framework, especially in the penetration testing and exploit research fields. This application can drastically reduce the amount of work that the users in this field will need to do in order to successfully complete their jobs. This application helps to reduce the amount of manual programming work that individuals might have had to do previously to using the framework. It also provides the users with multiple ways in which to undertake an exploitation task, and gives the ability to set the type of payloads, which they would like to use. These features are not widely found in other applications, free or otherwise; however, the framework can be used maliciously and therefore IDS and IPS signatures for any exploit that the framework provides need to be defined and used in any organization's network otherwise it will be susceptible to be attacked. In this paper, we show an exploit that can be easily detected and prevented by analyzing the network traffic for a defined signature of a malicious known exploit.

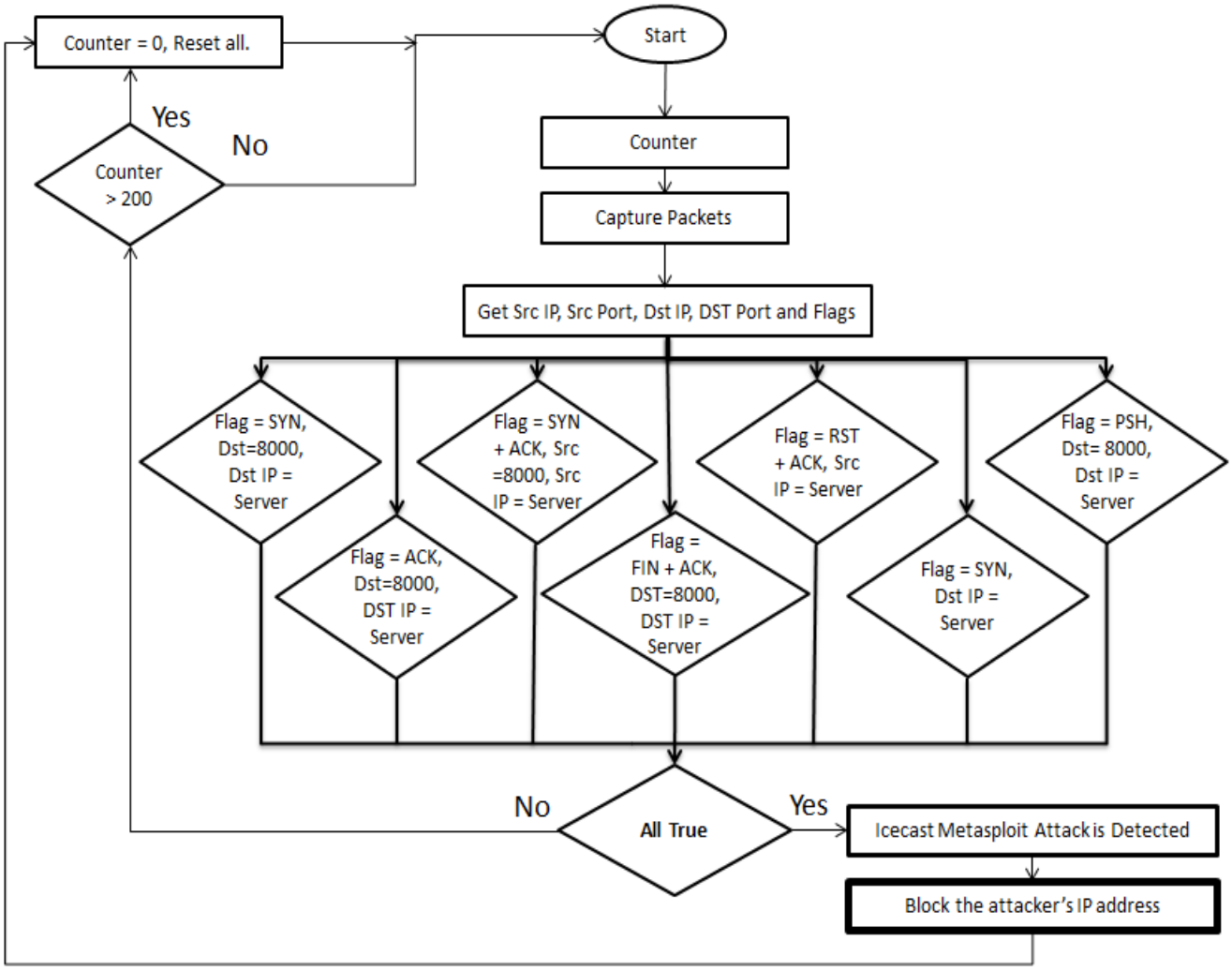

Fig 2 Metasploit Icecast Exploit Detection and Prevention Algorithm

References

[1] "The Metasploit Project," [Online]. Available: http://www.metasploit.com/

[2] D. Maynor, K. Mookhey, J. Cervini, F. Roslan and K. Beaver, ―Metasploit Toolkit. Burlington, MA: Syngress Publishing, 2007. [E-Book] Available: Syngress.com.

[3] "SANS: Network Penetration Testing and Ethical Hacking (GPEN)," [Online]. Available: http://www.sans.org/security-training/network-penetration-testing-ethical-hacking-937-mid.

[4] Patel, Sandip C., and Yingbing Yu. "Analysis of SCADA Security models. "International Management Review 3.2 (2007).

[5] Stouffer, K., J. Falco, and K. Kent. "Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security." NIST Special Publication (2006): 800-82.

[6] Axelsson, Stefan. "The base-rate fallacy and the difficulty of intrusion detection." ACM Transactions on Information and System Security (TISSEC) 3.3 (2000): 186-205.

[7] Alder, R., et al. "Snort: IDS and IPS Toolkit." (2007).

[8] Signatures, I. D. S., Field Device Protection Profile, and Nessus SCADA Plugins. "Digital Bond." (2007).

[9] Zhu, Bonnie, and Shankar Sastry. "SCADA-specific intrusion detection/prevention systems: a survey and taxonomy." Proceedings of the 1st Workshop on Secure Control Systems (SCS). 2010.

[10] Bigham, John, David Gamez, and Ning Lu. "Safeguarding SCADA systems with anomaly detection." Computer Network Security. Springer Berlin Heidelberg, 2003. 171-182.

[11] Yang, Dayu, Alexander Usynin, and J. Wesley Hines. "Anomaly-based intrusion detection for SCADA systems." 5th Intl. Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies (NPIC&HMIT 05). 2006.

[12] Barbosa, Rafael Ramos Regis, and Aiko Pras. "Intrusion detection in SCADA networks." Mechanisms for Autonomous Management of Networks and Services. Springer Berlin Heidelberg, 2010. 163-166.

[13] Carcano, Andrea, et al. "State-based network intrusion detection systems for SCADA protocols: a proof of concept." Critical Information Infrastructures Security. Springer Berlin Heidelberg, 2010. 138-150.

[14] Fovino, Igor Nai, et al. "Distributed intrusion detection system for SCADA protocols." Critical Infrastructure Protection IV. Springer Berlin Heidelberg, 2010. 95-110.

[15] Fovino, Igor Nai, et al. "Modbus/dnp3 state-based intrusion detection system." Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. IEEE, 2010.

[16] "Icecast is free server software for streaming multimedia," [Online]. Available: http://icecast.org/

Author –

Abdulaziz Almehmadi, PhD received the Bachelor's degree in computer science and the Master's degree in information technology security, with a specialty in biometrics, from King Abdulaziz University, Jeddah, Saudi Arabia, and the University of Ontario Institute of Technology (UOIT), Oshawa, ON, Canada in 2007 and 2010 respectively. Dr. Almehmadi received his PhD in computer science from UOIT in 2015 with a specialty in biometrics and access control. His thesis work was submitted to the United States Patent and Trademark Office (USPTO) and was granted the patent US9703952 in July 11, 2017 titled: Device and Method for Providing Intent-based Access Control. Dr. Almehmadi is currently working on designing non-identity-based access control systems to detect and prevent insider threats. He is an assistant professor at the Information Technology department at the Faculty of Computing and Information Technology (FCIT) at the University of Tabuk, Saudi Arabia. He is also the Vice-Dean for Graduate Studies and Scientific Research at FCIT. Furthermore, Dr. Almehmadi has recently founded and is the Director of the Industrial Innovation and Robotics Center (IIRC) at the University of Tabuk with projects to support the NEOM SmartCity.