Paper Review 2016.09.23

Reviewer: Zaina Hamid and Chun-Yu Chen

I. Paper Information

A. Title: Resilience of Deployed TCP to Blind Attack

B. Authors: Luckie, M. et al.

C. Published in: Proc. of ACM IMC '15, pp. 13-26, 2015

II. Summary of Paper

This paper gives an explanation about blind attacks within a web network and preventive measure, and specifically focuses on quantifying what percentage of web servers or internet infrastructure are vulnerable to TCP blind attacks. The authors conducted several simulated attacks, including blind RESET, SYN and data injection attacks. They set up an experimental environment for the attacks rather than conducting the attacks in the real environment to have better recording and control of the experiments.

The results have shown that 22%, 30%, and 38.4% of the system tested are vulnerable to in-window SYN/RESET attack, in-window data attack, and at least one kind of blind attacks, respectively. Furthermore, on the basis of the analysis of these attacks and respective vulnerabilities, a pattern has been drawn to compare validation methods, while suggesting alternatives.

III. What do we like about this paper (novel and “aha”)?

A. The authors stated specifically that their work can not represent any particular population. They just wanted to show the current behavior of

TCP stacks. This is very true since only limited machines and OSes have been tested.

B. The authors not only considered the TCP stacks in servers but also the devices of infrastructure. Because of the application nature, the implementation of servers and infrastructure differs. This is a point that we have not thought before.

C. In Section 3.1, the authors discovered that some TCP stacks will establish a parallel TCP connection using the same 4-tuple. This is an interesting fact that can be a hidden bug in the implementation.

D. Some of the functions in TCP stacks that are intended to enhance the robustness may lead to vulnerable point for blind attacks. For example, some systems will still send out ACK if an out-of-window message to indicate the expecting message.

E. It is interesting to know that even if there are suggestions in RFC, but there are still some systems that do not implement those measurements.

IV. How to extend or adopt the work?

A. Shortcomings and overlooked points

One of the most important shortcomings of this paper is that the attacks are all conducted in a simulation manner. The authors specifically created a experimental environment for testing and, therefore, the result cannot reveal the true condition when there is a real blind attack. For example, because the real blind attack will generate a lot of redundant packets, the server or infrastructure devices should be able to detect this condition and take further action. To have a more precise behavior of how systems will react to blind attacks, experiments conducted in real environment and having the condition of knowing no settings of the victims are crucial.

Furthermore, since the attacks were conducted in such a controlled environment, the authors cannot tell the degree of threat of blind attacks. In reality, attackers have no knowledge regarding to server ports or IPs connected to server. Therefore, blind attacks may be more difficult to succeed. Even though the paper is able to give a rough idea of how much attempts that a single attack needs, the real degree of difficulty to launch a successful attack is still unknown.

Also, during the test, the authors only conducted limited forms of attacks in each testing scenario. For example, in blind data injection attack, the attack is limited to transmit one of the three data sets twice. Maybe there are more TCP stacks that are vulnerable to blind data injection attack, but some of them still remain hidden because of the limited test cases. More test cases should be tested. The results in this paper are just lower bound of the percentage of vulnerable systems.

When the authors tried to justify whether an OS selects their port for TCP in a predictable manner, they only assume that the port selection can only be incremental. This is probably not the only case an OS is implemented. For example, the port selection can also be in a descending way. To have a more general view, they have to look into the patterns of selected ports.

In general, this paper needs more test cases and more testing devices to reveal the real threats from TCP blind attacks. Even though, the results shown in the paper are just a small part of the systems we use today, the results are some kind of lower bound of unsafe systems as mentioned earlier (there are not only blind attacks that are able to reset the TCP connections). One way to improve this work is that the authors can survey the percentage distribution of the servers' OSes, so that they can have weighted sum to have a more precise view that exactly how many percentage of systems is vulnerable to certain attacks.

B. Advantages not recognized

The blind attacks that the authors conducted can be a debugging tool to determine whether there are strange behaviors when unexpected packets are received. Furthermore, it can be a checking tool to test whether certain version of TCP stacks indeed implement the specified or suggested feature.

This paper also gives us a rough knowledge of whether certain operating systems follow the suggestions in RFC. Therefore, users are aware of the potential security threat they are facing. Users are now able to select the operating system with higher resilience to TCP blind attacks to deploy servers.

C. How does it compare to other works in area

The TCP-related researches focus on how to improve the efficiency of protocol and performance. While other papers trying to identify the potential threats to TCP stacks. This paper focuses on exploiting the vulnerability of different TCP stacks to known blind attacks. This paper provides a glance of how many machines or OSes will be affected if blind attacks are actually launched.

D. Application to your own work

The idea of using a simulated environment for testing is very useful for conducting preliminary experiments. Even though the results may not reflect 100% reality, it may give us the idea of how a system will work in reality. The reason to use simulated environment has two benefits: (1)

having a better controlled environment and (2) we do not have to worry about the authorities that are in charge of certain systems we are testing to consider us as malicious users.