Network filtering in a distributed app

Brian Cheung and Sam Wang

ABSTRACT

We will examine the vulnerabilities of a distributed application without using any layers of security. Then we will implement Pod Security Policies and Data Plane Filters to imporve the security of the application itself and filter out suspicious network traffic. The filtering of certain packets based on information such as the source/destination IP addresses and protocols can further reduce the attack surface and strengthen current security practices for applications. Through our research, we hope that the implementation of filtering out network traffic will lead to an increase in security performance.

1. INTRODUCTION

There exists many vulnerabilities within networks and distributed applications, and the implementation of network filtering as well as strong pod security policies can help mitigate these threats.

Second para: describe the key problem that if solved would make an impact. Why the current approaches leave a gap? Figuring out how to effectively utilize network filtering and pod security policies to work together will hopefully result in a strong security infrastructure that can successfully defend against all attacks and will be adopted by all applications.

Third: describe your approach. Key insight that enables your approach, and what is novel/interesting about the insight. We will be researching how network filtering and pod security policies function and how they can be applied to improving the security of an application. By understanding how each work, we will be able to understand how to effectively filter out network traffic that seems suspicious, thus reducing the amount of potential attacks from the start, and we will understand how to write a concrete and complete pod security policy to defend against the attacks that slip through the filter.

Fourth, fifth: Delve deeper into the approach and experimental setup. In the final report, describe key findings. First, we will setup a simple application and document the vulnerabilities. Next, we will implement pod security policies and document the results. Then we will implement network filtering on the base application without pod security policies and document the results. Finally, we will combine both layers of security and document the results. Hopefully, the final application will be free of vulnerabilities.

End with outline or what comes next and why.

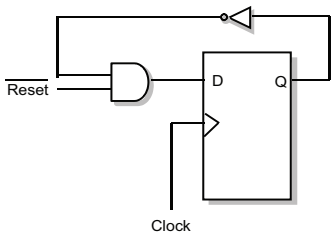

Figure 1: A 1-bit counter with reset. With the conventional technique of OR-ing all input shadow values, the feedback loop ensures that a counter shall never be trusted once it gets marked as untrusted. Our shadow logic is more precise and recognizes that a trusted reset guarantees a trusted 0 in the counter value.

2. MOTIVATION

Motivation describes the most important of the related works. The ones that you either build on, prove/disprove, or in any way "extend".

Other related work, that is orthogonal to your approach but is in the same general problem-area, can be included in a separate related work section. One good place for that is at the end, so it doesn't disrupt the story here.

It is important to continue researching how network filtering can lead to better application security, especially with a lot of technologies and applications all moving into the cloud. A strong initial filter that effectively identifies a majority of attacks dramatically decreases the chance of a successful attack. If most of the attempted attacks are initially thwarted, there are few attacks that can get through the security in place. A good network filter means less attacks an application has to be wary of, which should lead to a higher defense rate.

3. OUR ARCHITECTURE

4. EXPERIMENTAL RESULTS

First, we will improve the base security of the application through Pod Security Policies. Then we will add another layer of security through Data Plane filters in order to reduce the attack surface and prevent suspicious activity. These extra layers of security should hopefully eliminate the vulnerabilities of the application.

5. RELATED WORK

Point out other important approaches in the problem area. For example, if you are proposing an architecture, maybe OS or PL approaches to this problem.

The following paragraph included just for a figure. The caption of a figure is very important – I try to tell the entire story in the figures and captions alone, just in case that is all the reader sees.

The general problem of determining whether information flows in a program from variable x to variable y is undecidable, as "any procedure purported to decide it could be applied to the statement **if** $f(x)$ halts **then** $y := 0$ and thus provide a solution to the halting problem for arbitrary recursive function" [1].

`https://www.researchgate.net/publication/324562008_Threats_and_Vulnerabilities_of_Cloud_Computing_A_Review`
The article talks about how more and more enterprises are moving their workloads onto the cloud and while security has evolved over time, is still a major concern. The paper goes into details about the various forms of threats and vulnerabilities of the cloud.

`https://www.researchgate.net/publication/267691532_MODERN_NETWORK_SECURITY_ISSUES_AND_CHALLENGES`
This paper discusses the threats that networks face and the current network security practices to counteract these attacks.

`https://www.researchgate.net/publication/289756317_Security_Threats_on_Cloud_Computing_Vulnerabilities`
This paper further discusses the vulnerabilities of cloud computing services.

`https://www.researchgate.net/publication/334548954_Cloud_Security`
This paper discusses the advantages of using cloud services and also reveals the dangers and risks of those services.

6. CONCLUSIONS

7. REFERENCES

[1] D. E. Denning and P. J. Denning. Certification of programs for secure information flow. *Commun. ACM*, 20(7):504–513, 1977.