Strengthening Capsicum Capabilities with Libpreopen

Stanley Uche Godfrey

September 26, 2017

1 Introduction

On daily basis, more networks and gadgets are connected to the internet, the web has become indispensable as it hosts productivity software suites for creating documents, spreadsheets, and email. Applications suites for making scientific calculations, live television streaming, weather condition and forecast which are daily needs are also hosted on the web.

web applications provide online banking services, services for storing pictures and documents in the cloud, services that connect home devices such IP cameras to mobile phones for remote monitoring and e-commerce services. Sensitive data like passwords, credit card details are usually required to access these web services. These web applications that provide web services could have some vulnerabilities which attackers can exploit, despite network defenses like firewall and intrusion prevention systems.

One of such vulnerability in an application could be as a result of Buffer Overflow. Buffer Overflow is a programming error in which applications' buffers/ data structures have more data than they can accommodate, making the excess data to overflow to other storage or overwrite the data already stored in the buffer. A cybersecurity cracker who is able to add more data in a buffer than the buffer can accommodate could change execution path of applications intentionally and may acquire the root user right of the system which will allow the cybersecurity cracker to take total control of the system. Buffer Overflow vulnerability is common in application developed with C or C++ programming languages where pointer variables memory is allocated on the stack.

If a web application is vulnerable, Cyber attackers can use a technique known as Heap Spray to send malicious code to the heap memory of the web application in a computer. The Heap Spray technique is used to duplicate the malicious code in different locations of the running application heap memory to increase the chances of execution of the malicious code.

These sorts of unauthorized access to computer resources by Cyber attackers are what Capsicum mitigates and Libpreopen fortifies Capsicum in combating the damage intrusive malicious code could cause.

2 Conclusion

If a vulnerability in an application is exploited and malicious data injected into the system, Capsicum mitigates the spread of the malicious data by confining them to the affected process, since an application running in Capsicum sandboxed mode is compartmentalized into processes and each process sandboxed.

However, an application running in Capsicum sandboxed capability mode is forbidden to access global OS namespaces such as File system, Process IDs, IPC namespaces. The application also has a restricted access to system calls while access to system calls that involves global namespace access is forbidden. In other words, for Capsicum to contain the damage exploit of a vulnerability in an application can cause, the application has to give up its right to perform certain operations.

. Libpreopen brings solutions to this drawback of Capsicum, henceforth, applications running in Capsicum sandboxed capability mode can request directory file descriptor from the shell program Capsh. Capsh can send the directory file descriptor if Capsh has it pre-open, if not, Capsh will open the directory, add the directory file descriptor to the list of pre-open directory file descriptors and send the directory file descriptor with restricted capability rights on the directory file descriptor delegated to the untrusted application.

3 Bibliography and References

1. Paul Ionescu. The 10 Most Common Application Attacks in Action [online]. April 8, 2015. URL: https://securityintelligence.com/the-10-most-common-application-attacks-in-action/. Accessed August 16,2017
2. Mykola Protsenko. Practical Capabilities of UNIX. Conference Seminar on IT Security, 2011, 26-31.
3. Robert N. M. Watson, Jonathan Anderson, Ben Laurie, Kris Kennaway.Capsicum: practical capabilities for UNIX. Proceedings of the 19th USENIX Security Symposium, 2010.