The Paramount Importance of Secure Coding for Application Security

Project Group 3

Introduction

Secure coding: the practice of developing software in a manner which protects it against the introduction of security vulnerabilities.

As technology and applications continue to be more integrated in our daily lives, secure coding is vital for privacy/personal protection.

Huawei Cyber Security Evaluation Center (HCSEC) evaluates Huawei's technology for vulnerabilities.

The goal of secure coding practices is to prevent breaches of security and preserve the integrity of client data.

Origins Timeline

1988: the Morris worm was first spread on November 2, 1988 leading to the creation of DARPA and CERT/CC

2003/4: data stolen from Visa, resulting in the Payment Card Industry (PCI) Standards Council created secure coding guidelines known as PCI Data Security Standards (PCI-DDS)

2014: data stolen from Sony Pictures Entertainment, resulting in the push for two-factor authentication (2FA)

Origins Cont.

Secure coding is still infantile, with its origins traced back to just 32 years ago

As more security is in place, hackers continue to find ways through

Never ending cycle between secure coding and hacking

State of the Art

Software Guard Extensions (SGEs)

- Introduced by Intel in 2015
- Function by automating encryption of portions of memory by CPU
- Run in background, require no user input
- Downside: potential to be exploited in malware

Blockchain technology

- Creates secure, decentralized storage solutions
- Distributed ledger shared across numerous devices worldwide
- Data cannot be copied without owner's permission
- Utilized by NASA and under consideration by EU

State of the Art Cont.

Machine learning and AI advancements are mainstream keys for companies maintaining data security.

Supervised machine learning: utilizes labeled datasets and allows learning of what malicious behaviour looks like.

Unsupervised machine learning: utilizes unlabeled datasets and identifies behaviour that is different from the expected.

Analysis

Huawei build had code security problems in UK network

- No secure strategy to ensure that source code used is exactly what is compiled

Attempted to correct with binary equivalence

- Failed to implement

Huawei had developed secure coding guidelines

- Failed to enforce, resulting in risks for function calls such as *memcpy()*

Even one unsafe/insecure usage can open doors for hackers and vulnerabilities

Conclusions

HCSEC found numerous flaws and security concerns, from basic unsafe function calls to potentially catastrophic buffer overflows.

Huawei has the opportunity to improve with the continuing advancement of secure coding practices.

Huawei has the potential to successfully implement the safe coding guidelines created by HCSEC.

Fin

Thanks!