CyBok Project Showcase

SECURE CODING GAME-BASED LAB

2nd March, 2022

Manual Maarek

Sheung Chi Chan Heriot-Watt University

Leon McGregor

Secure Coding Game-based Lab

- Educational lab for secure coding based on a serious game of tower defence
- Complete the game-based educational platform by linking its security documentations and learning materials to CyBOK 1.1
- Help to promote serious game based learning of CyBOK

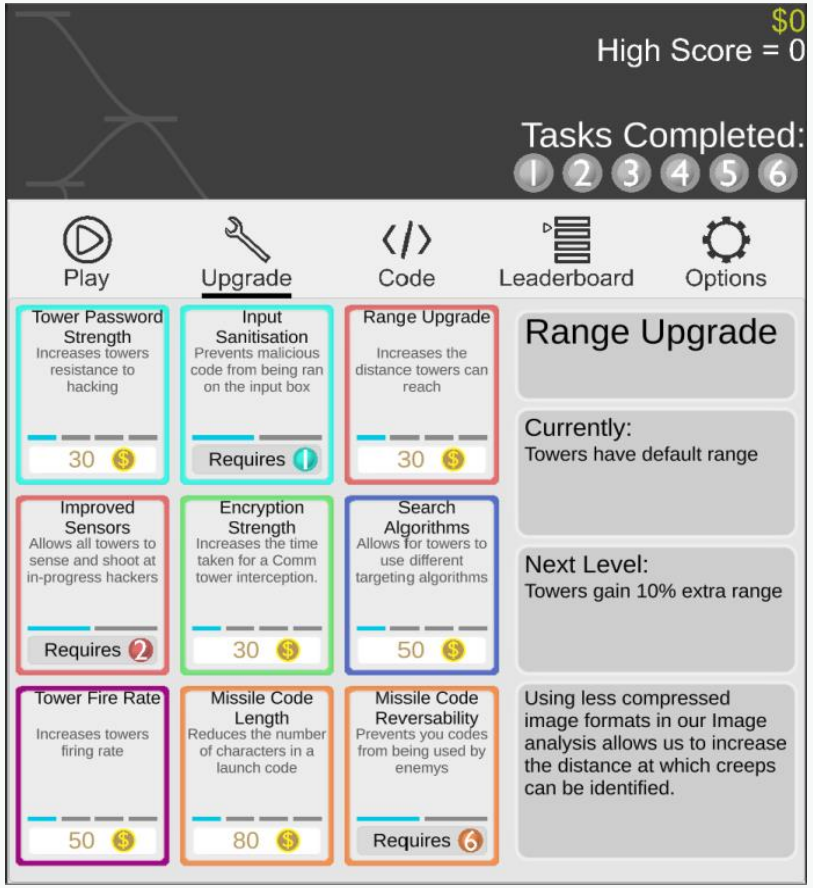
Educational Lab

Task: SSL/TLS

Transport layer security (TLS) is the successor of the now-deprecated Secure Socket Layer (SSL). They are a cryptographic based protocol to help establish and provide a secure communication channel between two stranger hosts through computer network. It can be adopted by different application in the application layer, like emails, Voice over IP (VOIP) or act as the secure channels for HTTP communication with the HTTP over SSL (HTTPS) services. TLS does not require the communication parties knowing each other in advance, the only requirement is they have the similar set of certificate validation through the x.509 certificate and public key infrastructure. With that, TLS can provide certain security features, including privacy, confidentiality, integrity and authenticity (optional) between two or more strangers who wish to communicate securely. The certificate validation and the cryptographic details and secret used in the communication are negotiate through a handshake protocol at the start of the communication. Optional two way authentication could be implemented. After a success negotiation, the TLS record protocol on each end of the communicating parties will make used of those agreed cryptographic choice and secret to wrap all remaining traffic to provide those security features. TLS is originally proposed by Internet Engineering Task Force (IETF) as SSL by Netscape in 1999.

Practical Tasks

Many of the existing SSL/TLS software and libraries has provide certain default settings to allow quick and secure negotiation of the communication channels. But sometimes when one of the parties does not have the latest standard of protocol supported, the default setting may forced to lower the security level and choose some older protocols. Also, with all those default settings, you will not have the chance to understand how the real negotiation and parameters has been con-




```
 keyring = KeyringConfigs.forGpgExportedKeys(KeyringConf
{
 keyring = KeyringConfigs.forGpgExportedKeys(KeyringConf

.addSecretKey(senderPrivateKey.getBytes(StandardCharsets
.addPublicKey(senderPublicKey.getBytes(StandardCharsets.
.addPublicKey(receiverPublicKey.getBytes(StandardCharset
ption | PGPException e) {
ntStackTrace();
null;


ign the message with the stored keyring
Stream baos = new ByteArrayOutputStream();

 BufferedOutputStream bos = new BufferedOutputStream(bao
 OutputStream os = BouncyGPG.encryptToStream()
                .withConfig(keyring)
                .withStrongAlgorithms()
                .toRecipient(receiverEmail)
                .andSignWith(senderEmail)
                .armorAsciiOutput()
                .andWriteTo(bos);
```

Linkage to CyBOK

CyBOK Knowledgebase Coverage

CyBOK 1.1 **knowledge areas** and *topics*	In-game exercises and corresponding game metaphors
Applied Cryptography	
Authenticated Encryption(AE)schemes	String encryption
Binding Public Keys and Identities via Certificates	Certificates
Cryptographic Libraries	All exercises
Diffie-Hellman Key Exchange	SSL
Digital Signatures	PGP
Hash functions	Credential storage
Managing Public Keys and Public Key Infrastructure	PGP, certificates
Law & Regulations	
electronic signatures and indentity trust services	PGP
prescriptive jurisdiction and data protection	Credential storage
Network Security	
Public Key Infrastructure	PGP, certificates
TLS (Transport Layer Security)	SSL
Privacy & Online Rights	
cryptography-based access control	String encryption
obfuscation-based inference control	URL shortener
privacy engineering	String encryption, URL shortener
Risk Management & Governance	
risk assessment	All exercises
Secure Software Lifecycle	
motivations for secure software lifecycle	All exercises
Software Security	
SQL injection	Credential storage
coding practices	All exercises
query generation	Credential storage
Web & Mobile Security	
SQL-injection	Credential storage
input sanitisation	Credential storage
password leaks	Credential storage
web PKI and HTTPS	PGP, certificates

Project Resources

- **Project public webpage**

https://citadel-programming-lab.gitlab.io/

- **Project source and deployment guideline**

https://gitlab.com/citadel-programming-lab/citadel-programming-lab