|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\Nabila Elshanum\Downloads\kad.jpg | **Tugas 6**  **Pemrograman Jaringan (CSH4V3)**  **------------------------------**  **Semester Genap 2018 - 2019**  **Dosen: Aulia Arif Wardana, S.Kom., M.T. (UIW)** | | |
| ***Berdo’alah sebelum mengerjakan. Dilarang berbuat curang.***  ***Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.***  ***Selamat belajar, semoga sukses !*** | | | |
| **Nama Mahasiswa:**  **Aup Hakim Nurzaman**  **……………………………..................................................** | | **NIM:**  **1301164395**  **…………………….............................** | **Nilai:**  **……………………** |
| **Nama Mahasiswa:**  **……………………………..................................................** | | **NIM:**  **…………………….............................** | **Nilai:**  **……………………** |
| **Nama Mahasiswa:**  **……………………………..................................................** | | **NIM:**  **…………………….............................** | **Nilai:**  **……………………** |

|  |
| --- |
| **Siapkan tools berikut sebelum mengerjakan:** |
| 1. Go Programming Language (<https://golang.org/dl/>). 2. Visual Studio Code (<https://code.visualstudio.com/>) atau LiteIDE (<https://github.com/visualfc/liteide>). 3. Disarankan untuk menggunakan linux dengan distro fedora (<https://getfedora.org/id/workstation/>). 4. Buatlah git repository pada <https://github.com/> kemudian push semua kode dan hasil laporan anda ke dalam repository github yang sudah anda buat. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini. 5. Buatlah Virtual Machine (VM) dengan virtualbox (<https://www.virtualbox.org/wiki/Downloads>). Gunakan operating system Centos (<https://www.centos.org/>) pada VM tersebut. VM yang sudah anda buat akan digunakan sebagai server. 6. Gunakan openssl (<https://www.openssl.org/>) untuk mengerjakan salah satu tugas pada modul ini. |

|  |
| --- |
| **Soal No 1 (Secure Web Server Design)** |
| Buatlah perancangan aplikasi Web Server yang dapat melakukan serve koneksi HTTPS menggunakan diagram FSM serta jelaskan cara kerjanya! |
| Jawaban:    Cara Kerja :  Pada browser meminta data web page kepada server, maka instruksi permintaan data oleh browser tersebut akan dikemas di dalam TCP yang merupakan protokol transport dan dikirim ke alamat yang dalam hal ini merupakan protokol berikutnya yaitu HTTPS.  Data yang diminta dari browser ke web server disebut dengan HTTPS request kemudian akan dicarikan oleh web server di dalam data server. Jika ditemukan, data tersebut akan dikemas oleh web server dalam TCP dan dikirim kembali ke browser untuk ditampilkan.  Data yang dikirim dari server ke browser disebut dengan HTTPS response. |

|  |
| --- |
| **Soal No 2 (Web Server Implementation)** |
| Implementasikan aplikasi secure web server dari design yang sudah anda buat, aplikasi harus mempunyai config file untuk melakukan konfigurasi aplikasi!  Anda bisa mengembangkan dari code berikut <https://github.com/aulwardana/simpe-web-server> |
| Jawaban:   |  | | --- | | package main | |  |  | |  | import ( | |  | "fmt" | |  | "io/ioutil" | |  | "strconv" | |  | "log" | |  | "net/http" | |  | "regexp" | |  | "encoding/json" | |  | ) | |  |  | |  | func SetMyCookie(response http.ResponseWriter){ | |  |  | |  | cookie := http.Cookie{Name: "testcookiename", Value:"testcookievalue"} | |  | http.SetCookie(response, &cookie) | |  | } | |  |  | |  | func GenericHandler(response http.ResponseWriter, request \*http.Request){ | |  |  | |  |  | |  | SetMyCookie(response) | |  | response.Header().Set("Content-type", "text/plain") | |  |  | |  | // Parse URL and POST data into the request.Form | |  | err := request.ParseForm() | |  | if err != nil { | |  | http.Error(response, fmt.Sprintf("error parsing url %v", err), 500) | |  | } | |  |  | |  | // Send the text diagnostics to the client. | |  | fmt.Fprint(response, "FooWebHandler says ... \n") | |  | fmt.Fprintf(response, " request.Method '%v'\n", request.Method) | |  | fmt.Fprintf(response, " request.RequestURI '%v'\n", request.RequestURI) | |  | fmt.Fprintf(response, " request.URL.Path '%v'\n", request.URL.Path) | |  | fmt.Fprintf(response, " request.Form '%v'\n", request.Form) | |  | fmt.Fprintf(response, " request.Cookies() '%v'\n", request.Cookies()) | |  | } | |  |  | |  | // Respond to the URL /home with an html home page | |  | func HomeHandler(response http.ResponseWriter, request \*http.Request){ | |  | response.Header().Set("Content-type", "text/html") | |  | webpage, err := ioutil.ReadFile("home.html") | |  | if err != nil { | |  | http.Error(response, fmt.Sprintf("home.html file error %v", err), 500) | |  | } | |  | fmt.Fprint(response, string(webpage)); | |  | } | |  |  | |  | // Respond to URLs of the form /item/... | |  | func ItemHandler(response http.ResponseWriter, request \*http.Request){ | |  |  | |  | // Set cookie and MIME type in the HTTP headers. | |  | SetMyCookie(response) | |  | response.Header().Set("Content-type", "application/json") | |  |  | |  | // Some sample data to be sent back to the client. | |  | data := map[string]string { "what" : "item", "name" : "" } | |  |  | |  | // Was the URL of the form /item/name ? | |  | var itemURL = regexp.MustCompile(`^/item/(\w+)$`) | |  | var itemMatches = itemURL.FindStringSubmatch(request.URL.Path) | |  |  | |  | if len(itemMatches) > 0 { | |  |  | |  | data["name"] = itemMatches[1] | |  | json\_bytes, \_ := json.Marshal(data) | |  | fmt.Fprintf(response, "%s\n", json\_bytes) | |  |  | |  | } else { | |  | // No, so send "page not found." | |  | http.Error(response, "404 page not found", 404) | |  | } | |  | } | |  |  | |  | func main(){ | |  | port := 8097 | |  | portstring := strconv.Itoa(port) | |  | |  | |  | |  | |  | |  | |  | mux := http.NewServeMux() | |  | mux.Handle("/home", http.HandlerFunc( HomeHandler )) | |  | mux.Handle("/item/", http.HandlerFunc( ItemHandler )) | |  | mux.Handle("/generic/", http.HandlerFunc( GenericHandler )) | |  |  | |  | |  | |  | log.Print("Listening on port " + portstring + " ... ") | |  | err := http.ListenAndServe(":" + portstring, mux) | |  | if err != nil { | |  | log.Fatal("ListenAndServe error: ", err) | |  | } | |  | } | |  |  | |