

Đào tạo An toàn thông tin VTICT

Quangbx1@viettel.com.vn



TRUNG TÂM GIẢI PHÁP CÔNG NGHỆ THÔNG TIN & VIỄN THÔNG VIETTEL

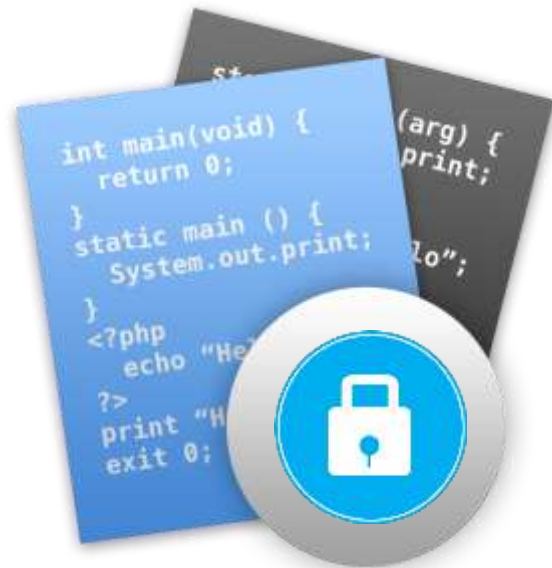
4/25/2015

Hãy nói theo cách của bạn

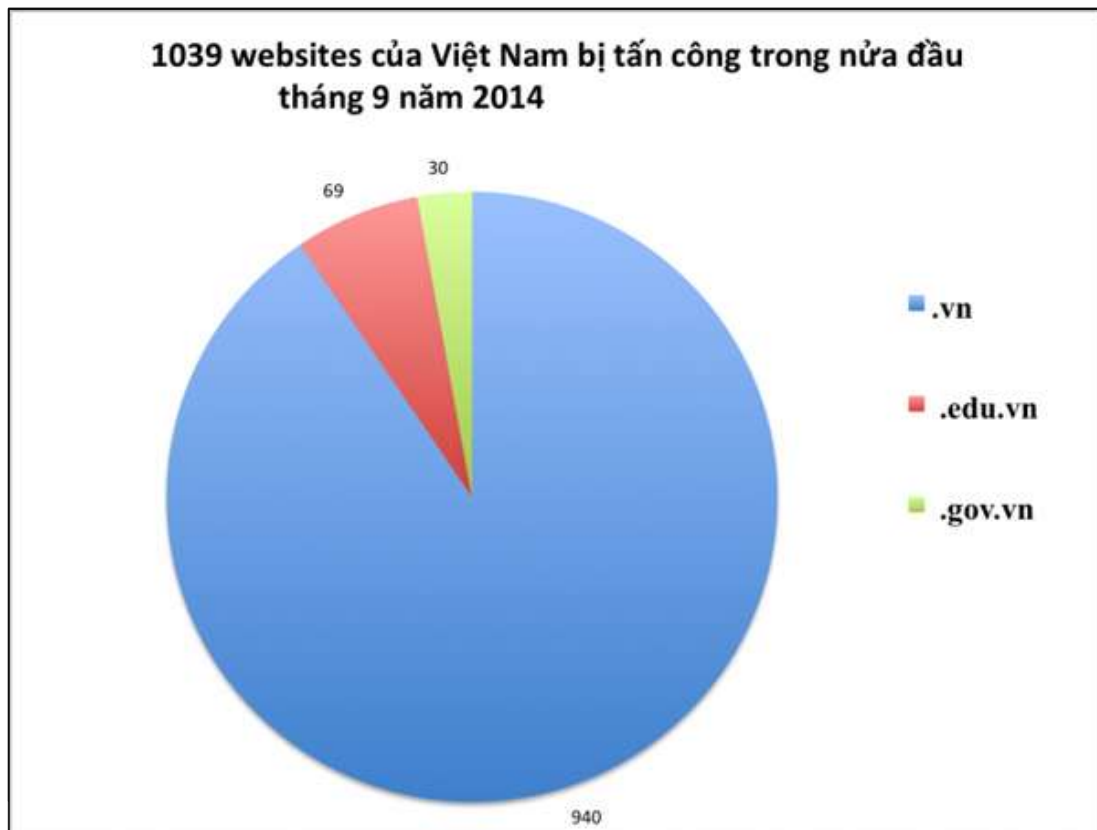
01 Tổng quan

02 An toàn thông tin người dùng

03 An toàn thông tin trong phát triển phần mềm



01 Tổng quan



0



Date	Notifier	H M R L	★ Domain	OS	View
2014/11/26	the_warri0r	R	🇻🇳 vts.net.vn/al.txt	Linux	mirror

www.bvts.com

Hello World

Second Message:

There Is No Full Security
We Can Catch You !



2014/11/20	Cyb3r_Sw0rd	H	🇻🇳 thietkenoithatkientruc.vn	Linux	mirror
------------	-------------	---	------------------------------	-------	--------

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

01 Tổng quan

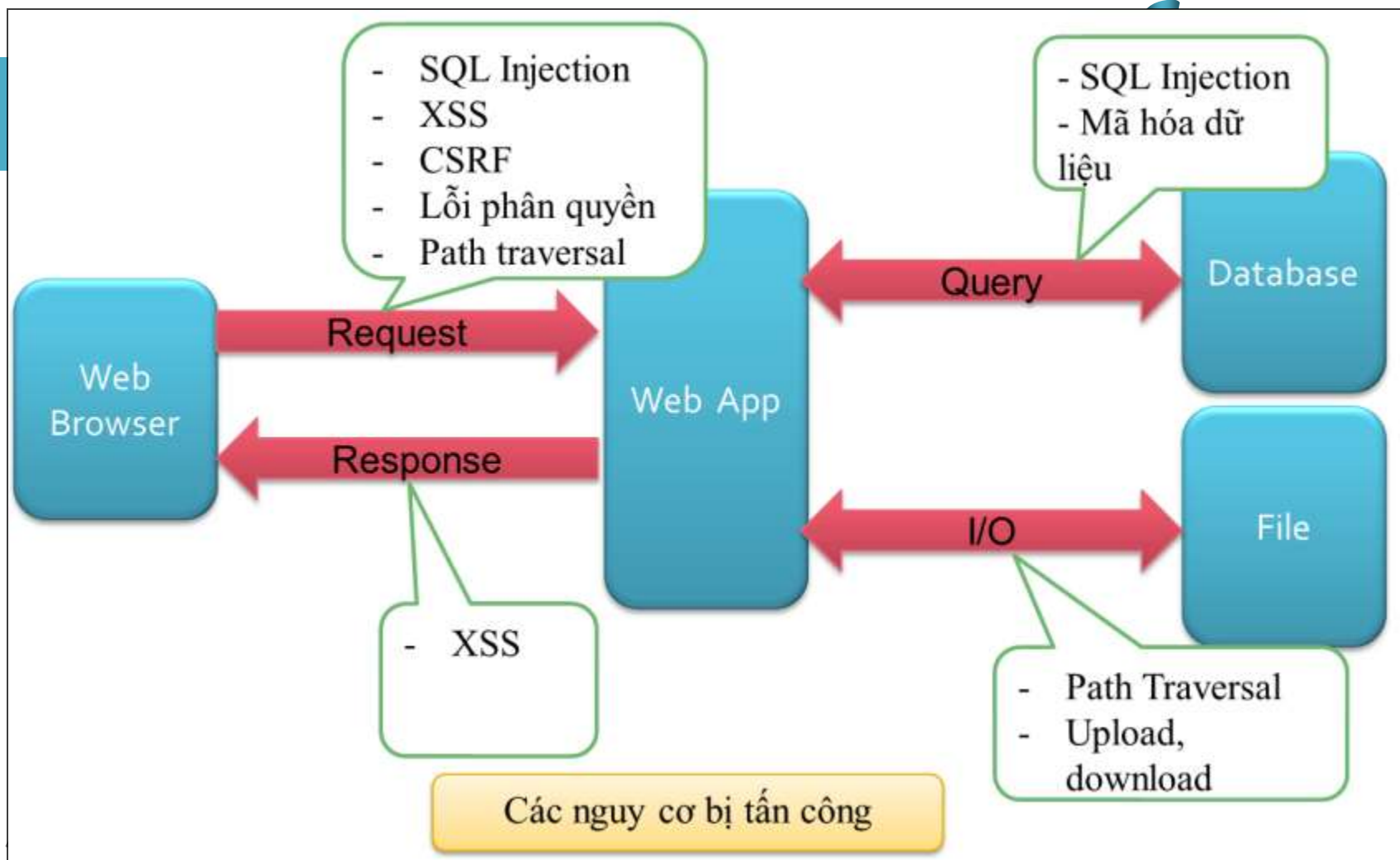
❖ Các sự cố xảy ra tại VTICT 2014

- ✓ Voffice và tra cứu phụ gia (Cục Vệ sinh an toàn thực phẩm) bị hacker tấn công chiếm quyền điều khiển, cài backdoor (tháng 12/2014).
- ✓ Tấn công DDoS hệ thống Portal Bộ y tế (tháng 11/2014).
- ✓ Tấn công Social Engineering (tháng 2/2015).
- ✓ Bùng nổ mã độc diện rộng tại TT Chính phủ.
- ✓ Sự cố hệ thống Shop.viettel.vn.

01 Tổng quan

❖ Các nguyên cơ đối với ứng dụng web

- ✓ Tấn công vào công tác vận hành, quản trị: Khai thác điểm yếu trong cơ chế quản trị, người dùng.
- ✓ Tấn công vào nền tảng phục vụ ứng dụng web: Khai thác vào lỗ hổng webserver, máy chủ OS, DB, dịch vụ đang chạy trên máy chủ.



01 Tổng quan

❖ Các nguy cơ đối với ứng dụng web

- Nguy cơ lập trình không an toàn.
- Nguy cơ sử dụng thư viện, third party không an toàn.
- Nguy cơ do thiết kế, phân tích yêu cầu.

02

An toàn thông tin người dùng

- ❖ Quy chế ATTT – 2714
 - Bảo vệ thông tin/Dữ liệu
 - Quy định truy cập internet
 - Quy định sử dụng phần mềm và cấu hình trên máy tính người dùng.
 - Quy định phòng chống virus máy tính

03

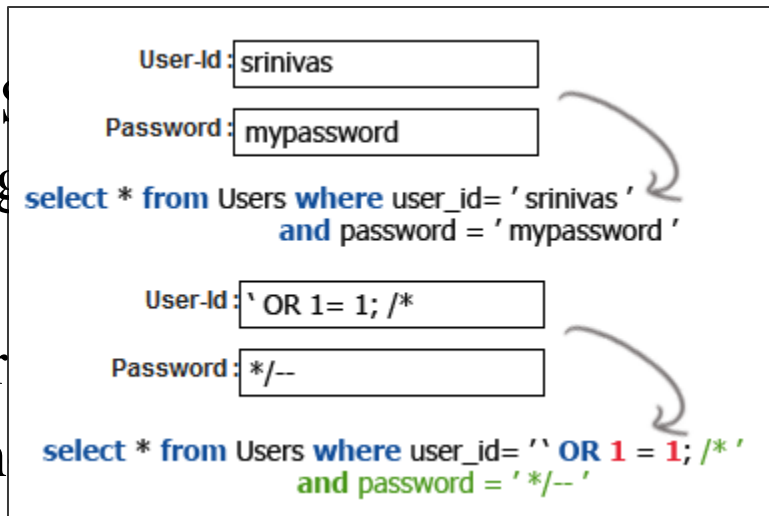
Các lỗi hổng ATTT trên ứng dụng web

❖ SQL Injection

❖ Nguyên nhân: SQL
các biến được gán

❖ Biện pháp:

- Tham số hóa truy vấn
- Trường hợp không thể whitelist để validate



uy vấn DB với

whitelist để

03

Các lỗ hổng ATTT trên ứng dụng web

❖ SQL Injection

Xử lý với Java (Hibernate)

```
@Override
public Users loginHandle(String username, String password) throws Exception {
    List<Object> params = new ArrayList<Object>();
    StringBuilder queryString = new StringBuilder();
    queryString.append("SELECT * FROM USERS WHERE USERNAME = ? AND PASSWORD = ?");
    params.add(username);
    params.add(password);
    return repo.getFirst(Users.class, queryString.toString(), params.toArray());
}
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ SQL Injection

Xử lý với ASP

Fix lỗi

```
string _sql = @"SELECT [Id],[Username],[Password],[Email] FROM [dbo].[System_Users] " +  
    @"WHERE [Username] = @un";  
var cmd = new SqlCommand(_sql, cn);  
cmd.Parameters  
    .Add(new SqlParameter("@un", SqlDbType.NVarChar))  
    .Value = username;  
cn.Open();  
var reader = cmd.ExecuteReader();  
if (reader.HasRows && reader.Read())  
{  
    UserRegister u = new UserRegister();  
    u.Id = (int)reader["Id"];  
    u.UserName = reader["Username"].ToString();  
    u.Password = reader["Password"].ToString();  
    u.Email = reader["Email"].ToString();  
    reader.Dispose();  
    cmd.Dispose();  
    return u;  
}
```

03

Các lỗ hổng ATTT trên ứng dụng web

❖ SQL Injection





03

Các lỗi hổng ATTT trên ứng dụng web

❖ XSS (Cross-Site Scripting)

Store XSS

```
<td style="color: blue; font-size: 40px;">
  Welcome <b>${sessionScope.USER_SESSION_NAME}</b>
</td>
```

Fix XSS

```
<td style="color: blue; font-size: 40px;">
  Welcome <b>${fn:escapeXml(sessionScope.USER_SESSION_NAME)}</b>
</td>
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ XSS (Cross-Site Scripting)

Fix XSS

```
<strong style="color: blue;">  
    Xin chào @Html.Encode(User.Identity.Name)  
</strong>
```

❖ Sử dụng các thư viện hỗ trợ

```
string uname = Server.HtmlEncode(user.UserName);
```

```
<httpRuntime encoderType="System.Web.Security.AntiXss.AntiXssEncoder" />
```

```
string uname = AntiXssEncoder.HtmlEncode(user.UserName);
```


03

Các lỗ hổng ATTT trên ứng dụng web

❖ XSS (Cross-Site Scripting)



03

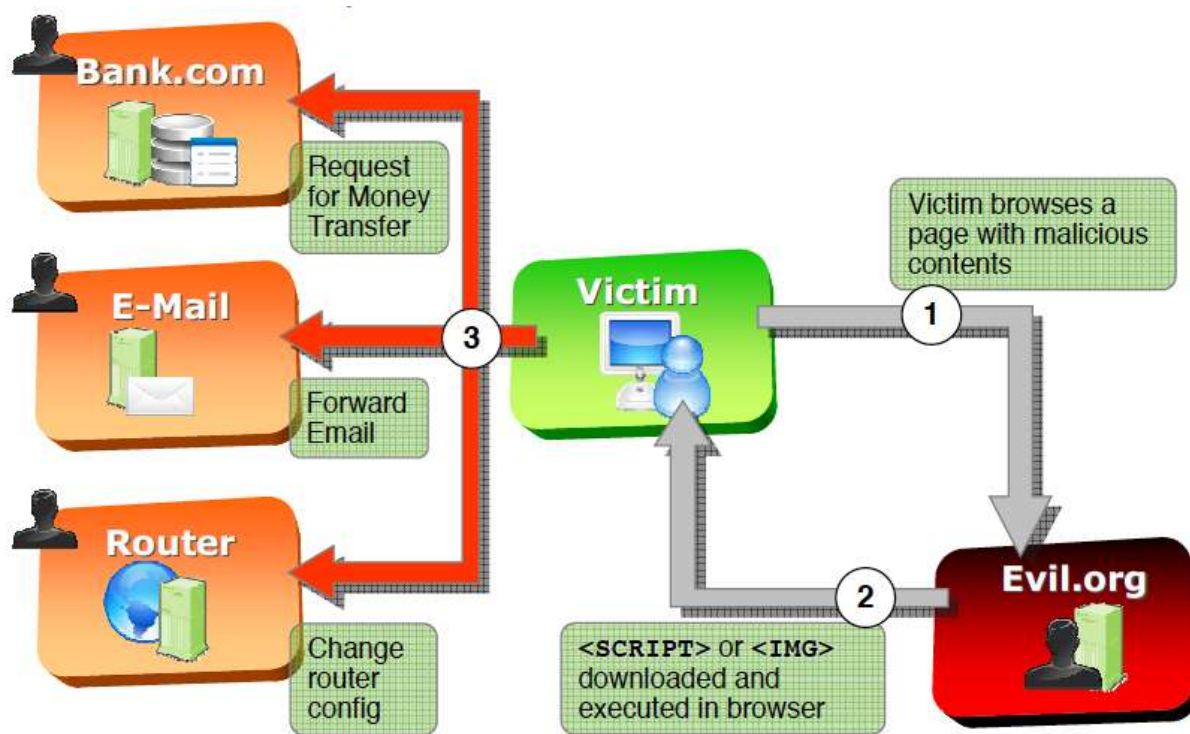
Các lỗ hổng ATTT trên ứng dụng web

- ❖ **CSRF (Cross-Site Request Forgery)**
- ❖ **Nguy cơ:** là kiểu tấn công lừa người dùng thực hiện một hành động mà họ không mong muốn lên ứng dụng web, bằng chính quyền người dùng đó.
- ❖ **Biện pháp:** Sử dụng token (sinh ngẫu nhiên) trong các xử lý quan trọng, server kiểm tra tính hợp lệ của token này.

03

Các lỗ hổng ATTT trên ứng dụng web

❖ CSRF (Cross-Site Request Forgery)



03

Các lỗi hổng ATTT trên ứng dụng web

- ❖ CSRF (Cross-Site Request Forgery)
- ❖ Khuôn dạng request:

```
http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243
```

- ❖ Khai thác lỗi:

```

```

03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ CSRF (Cross-Site Request Forgery)
- ❖ Fix lỗi
- ❖ Phía server: Tạo giá trị token và lưu vào session.
- ❖ Trong các chức năng quan trọng: Viết code xử lý khi gửi request sẽ gửi kèm giá trị token.
- ❖ Phía Server: kiểm tra tính hợp lệ của request bằng cách so sánh giá trị token gửi lên từ client và token lưu trong session.
- ❖ Hợp lệ: Thực thi code, reset lại token và lưu lại trong session

03

Các lỗi hổng ATTT trên ứng dụng web

- ❖ CSRF (Cross-Site Request Forgery)
- ❖ Java - Struts2

Method tạo token (gọi sau khi login thành công)

```
public void generateToken() {  
    if (request != null && request.getSession() != null) {  
        token = UUID.randomUUID().toString();  
        request.getSession().setAttribute("SESSION_TOKEN",  
                                           token);  
    }  
}
```

03

Các lỗi hổng ATTT trên ứng dụng web

```
public String resetToken() {  
    generateToken();  
    return getSessionToken();  
}
```

Reset Token



```
public String getSessionToken() {  
    if (request != null  
        && request.getSession() != null  
        && request.getSession().getAttribute(  
            "SESSION_TOKEN") != null) {  
        return request.getSession()  
            .getAttribute("SESSION_TOKEN").toString();  
    }  
    return null;  
}
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ CSRF (Cross-Site Request Forgery)

❖ Struts2

SecurityInterceptor.java

Override

```
public String intercept(ActionInvocation invocation) throws Exception {  
    final ActionContext context = invocation.getInvocationContext();  
    HttpServletRequest request = (HttpServletRequest) context.get(HTTP_REQUEST);  
    Object reqToken = request.getParameter("token");  
    Object sesToken = request.getSession().getAttribute("SESSION_TOKEN");  
    if(reqToken != null && sesToken != null && reqToken.equals(sesToken)) {  
        return invocation.invoke();  
    } else {  
        return "invalid.token";  
    }  
}
```



```
<interceptor name="securityInterceptor"
    class="com.java.interceptor.SecurityInterceptor"/>
<interceptor-stack name="securityInterceptStack">
    <interceptor-ref name="servletConfig" />
    <interceptor-ref name="params" />
    <interceptor-ref name="staticParams" />
    <interceptor-ref name="securityInterceptor" />
    <interceptor-ref name="prepare" />
    <interceptor-ref name="chain" />
    <interceptor-ref name="modelDriven" />
    <interceptor-ref name="fileUpload" />
    <interceptor-ref name="checkbox" />
    <interceptor-ref name="staticParams" />
    <interceptor-ref name="params" />
    <interceptor-ref name="conversionError" />
    <interceptor-ref name="validation" />
    <interceptor-ref name="workflow" />
</interceptor-stack>
```



Struts.xml

03

Các lỗi hổng ATTT trên ứng dụng web

❖ CSRF (Cross-Site Request Forgery)

❖ Struts2

Sử dụng Interceptor trong action

```
<action name="deleteUser" class="com.java.actions.account.UserAction"
    method="deleteUser">
    <interceptor-ref name="securityInterceptStack"></interceptor-ref>
    <result name="error">/Error.jsp</result>
    <result type="json" name="success">
        <param name="root">result</param>
    </result>
</action>
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ CSRF (Cross-Site Request Forgery)



```
deleteUser = function(id) {  
    fancyConfirm("Bạn muốn xóa user??", function(ret) {  
        if (ret == true) {  
            var tk = '${sessionScope.SESSION_TOKEN}';  
            var kData = {"id":id, "token":tk};  
            $.ajax({  
                type : "POST",  
                url : "deleteUser.htm",  
                data : kData,  
                dataType : "json",  
                success : function(msg) {  
                    if (msg=='OK') {  
                        $.fancybox('Delete thành công!!!',{
```

Code xử lý gửi request



03

Các lỗi hổng ATTT trên ứng dụng web

❖ CSRF (Cross-Site Request Forgery)

❖ Struts2

```
try {  
    Users u = new Users();  
    u.setId(id);  
    usersMgr.deleteUser(u);  
    result = "OK";  
    resetToken();  
    return SUCCESS;  
} catch (Exception ex) {  
    Logger.getLogger(UserAction.class.getName())  
        .log(Level.SEVERE, null, ex);  
}
```

Action xử lý Delete

03

Các lỗi hổng ATTT trên ứng dụng web

```
❖ [ValidateAntiForgeryToken]
❖ [HttpPost]
❖ public ActionResult UpdateUser(int id, string Username,
{
    try
    {
        UserRegister user = new UserRegister();
        user.Id = id;
        user.UserName = Username;
        user.Password = Password;
        user.Email = Email;
        Repositories.updateUser(user);
        return Json(new { Type = "SUCCESS" });
    }
}
```

Action xử lý Update



03

Các lỗi hổng ATTT trên ứng dụng web

```
update = function () {  
    var id = @Model.Id;  
    var un = $("#Username").val();  
    var pw = $("#Password").val();  
    var em = $("#Email").val();  
    var token = $("input[name=__RequestVerificationToken]").val()  
    var kData = {"id":id,"username":un, "password":pw,  
                "email":em, "__RequestVerificationToken":token}  
    $.ajax({  
        url: '@Url.Action("UpdateUser", "Manager")',  
        data: kData,  
        type: 'POST',  
        datatype: 'json',  
        success: function (msg) {  
            if (msg.Type == "SUCCESS") {
```

Code xử lý tại View



03

Các lỗi hỏng ATTT trên ứng dụng web

❖ CSRF



03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ Kiểm soát file upload lên hệ thống
- ❖ **Nguyên cơ:** Các chức năng upload file, dữ liệu lên server nếu không kiểm soát tốt dẫn đến upload các file không hợp lệ (như webshell, file cấu hình,...)
- ❖ **Biện pháp:** Kiểm soát phía server:
 - Extension của file
 - Lọc các ký tự '/', '\', ký tự null
 - Sinh ngẫu nhiên tên file

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Kiểm soát file upload lên hệ thống

Code xử lý Upload

```
public static boolean checkExtentionImgUpload(String fileName) {  
    St public static String getSafeFileName(String input) {  
        in    StringBuilder sb = new StringBuilder();  
        if    for (int i = 0; i < input.length(); i++) {  
            char c = input.charAt(i);  
        }    if (c != '/' && c != '\\ ' && c != 0) {  
        if    sb.append(c);  
            }  
        }  
    }    return sb.toString();  
    re }  
}
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Kiểm soát file upload lên hệ thống



03

Các lỗi hổng ATTT trên ứng dụng web

- ❖ **Path Traversal – Download file**
- ❖ **Nguyên cơ:** Các xử lý download qua action truyền filename nếu không xử lý filename dẫn đến attacker có thể download các file config của ứng dụng hay cấu hình server.
- ❖ **Biện pháp:** Trước khi xử lý, cần lọc các ký tự ‘/’, ‘\’, ký tự null.

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Path Traversal – Download file

```
GET /download?fileName=../META-INF/context.xml HTTP/1.1
Host: 10.61.59.253:8084
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5

Response

Raw Headers Hex XML
testOnBorrow="true" testOnReturn="false" validationQuery="SELECT 1 FROM DUAL"
validationInterval="30000" timeBetweenEvictionRunsMillis="30000"
maxActive="50" minIdle="50" maxWait="-1" initialSize="10"
removeAbandonedTimeout="100" removeAbandoned="true" logAbandoned="true"
minEvictableIdleTimeMillis="30000" jmxEnabled="true"
jdbcInterceptors="org.apache.tomcat.jdbc.pool.interceptor.ConnectionState;org.apache.tomcat.jdbc.pool.interceptor.StatementFinalizer"
username="242-145-244-171-79-69-142-164"
password="172-141-160-58-249-191-28-58"
driverClassName="com.mysql.jdbc.Driver"

url="128-157-97-173-134-91-229-245-38-125-176-19-154-39-192-151-155-121-185-185-137-237-158-63-97-20-97-89-10-130-133-129-172-141-160-58-249-191-28-58" /> -->
<Resource auth="Container" driverClassName="com.mysql.jdbc.Driver" factory="org.apache.tomcat.jdbc.pool.DataSourceFactory" initialSize="10"
jdbcInterceptors="org.apache.tomcat.jdbc.pool.interceptor.ConnectionState;org.apache.tomcat.jdbc.pool.interceptor.StatementFinalizer" jmxEnabled="true"
logAbandoned="true" maxActive="50" maxIdle="50" maxWait="-1" minEvictableIdleTimeMillis="30000" minIdle="10" name="jdbc/java" password="" removeAbandoned="true"
removeAbandonedTimeout="60" testOnBorrow="true" testOnReturn="false" testWhileIdle="true" timeBetweenEvictionRunsMillis="30000" type="javax.sql.DataSource"
url="jdbc:mysql://localhost:3306/java" username="root" validationInterval="30000" validationQuery="SELECT 1 FROM DUAL"/>
</Context>
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Path Traversal – Download file

Fix lỗi

```
protected void  
    if (request.getParameter("fileName") != null) {  
        String fileName = request.getParameter("fileName");  
        String filePath = "C:\\Users\\Admin\\Desktop\\file.txt";  
        logger.info("Attempting to download file: " + fileName);  
        DownloadFile(filePath, fileName);  
    }  
}  
  
public static String getSafeFileName(String input) {  
    StringBuilder sb = new StringBuilder();  
    for (int i = 0; i < input.length(); i++) {  
        char c = input.charAt(i);  
        if (c != '/' && c != '\\ ' && c != 0) {  
            sb.append(c);  
        }  
    }  
    return sb.toString();  
}
```

03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ Mã hóa dữ liệu nhạy cảm
- ❖ **Nguy cơ:** Bằng một cách nào đó (khai thác lỗ hổng hoặc có quyền truy cập DB), Attacker lấy được các thông tin nhạy cảm trong DB. Các thông tin sẽ bị lộ nếu không mã hóa hoặc mã hóa không an toàn.
- ❖ **Biện pháp:**
 - Mã hóa dữ liệu nhạy cảm trong DB
 - Các hàm mã hóa 1 chiều phải dùng Salt

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Mã hóa dữ liệu nhạy cảm

Code lỗi

```
try {
    if (usersMgr.checkUserExist(username) != null) {
        result = "NotOK";
        return SUCCESS;
    }
    Users u = new Users();
    u.setUsername(username);
    u.setPassword(EncryptionUtils.encryptMD5(password));
    usersMgr.createUser(u);
    result = "OK";
    return SUCCESS;
} catch (Exception ex) {
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Mã hóa dữ liệu nhạy cảm

Fix lỗi

```
try {  
    if (usersMgr.checkUserExist(username) != null) {  
        result = "NotOK";  
        return SUCCESS;  
    }  
    Users u = new Users();  
    u.setUsername(username);  
    u.setPassword(EncryptionUtils.encryptMD5(password  
        + EncryptionUtils.encryptMD5(username)));  
    usersMgr.createUser(u);  
    result = "OK";  
    return SUCCESS;  
} catch (Exception ex) {
```


03

Các lỗi hỏng ATTT trên ứng dụng web

- ❖ Phân quyền
- ❖ **Nguyên cơ:** Một hệ thống phân quyền không tốt, dẫn đến người dùng có thể truy cập đến các chức năng, dữ liệu không được phép.
- ❖ **Biện pháp:** Kiểm tra quyền trong request gửi lên server
 - Người dùng có được phép thực hiện chức năng?
 - Người dùng thực hiện chức năng trên vùng dữ liệu cho phép?

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Phân quyền

```
public String getListUser() {  
    try {  
        if (!session.getAttribute("USER_SESSION_HANDLE").equals("admin")) {  
            result = "Không có quyền thực hiện action này";  
            return ERROR;  
        }  
        listUsers = usersMgr.getListUsers();  
        listUsers.remove(0);  
        return SUCCESS;  
    } catch (Exception e) {  
        e.printStackTrace();  
        return ERROR;  
    }  
}
```

Check quyền thực hiện chức năng



03

Các lỗi hổng ATTT trên ứng dụng web

❖ Phân quyền

Check phân quyền dữ liệu

```
public String userDetails() {  
    try {  
        if (id != null) {  
            if (id == session.getAttribute("USER_SESSION_ID")) {  
                return ERROR;  
            }  
            user = usersMgr.getUserDetail(id);  
            if (user != null) {  
                return SUCCESS;  
            }  
        }  
    } catch (Exception e) {
```

03

Các lỗ hổng ATTT trên ứng dụng web

❖ User enumeration

❖ Nguy cơ:

- Trong chức năng đăng nhập, nếu thông báo lỗi quá chi tiết dẫn đến attacker có thể thử và tìm ra thông tin user có trên hệ thống.
- Với chức năng như reset password, forgot password, đăng ký cho phép thông báo user đúng hay sai dẫn đến attacker có thể thử và tìm ra thông tin user có trên hệ thống

03

Các lỗ hổng ATTT trên ứng dụng web

❖ User enumeration

❖ Biện pháp:

- Sử dụng chung thông báo lỗi cho trường hợp login sai username hay password.
- Sử dụng captcha cho các chức năng đăng ký, reset, forgot password

03

Các lỗi hổng ATTT trên ứng dụng web

```
Users u = usersMgr.loginHandle(username, EncryptionUtils.encryptMD5(password));
if (u != null) {
    session.invalidate();
    session = request.getSession(true);
    session.removeAttribute("USER_SESSION_HANDLE");
    session.setAttribute("USER_SESSION_NAME", u.getUsername());
    generateToken();
    errorStr = "Login thành công!";
    result = "OK";
    return SUCCESS;
} else {
    errorStr = "Sai thông tin đăng nhập!";
    result = "NotOK";
    return SUCCESS;
}
```

03

Các lỗ hổng ATTT trên ứng dụng web

❖ Session Fixation

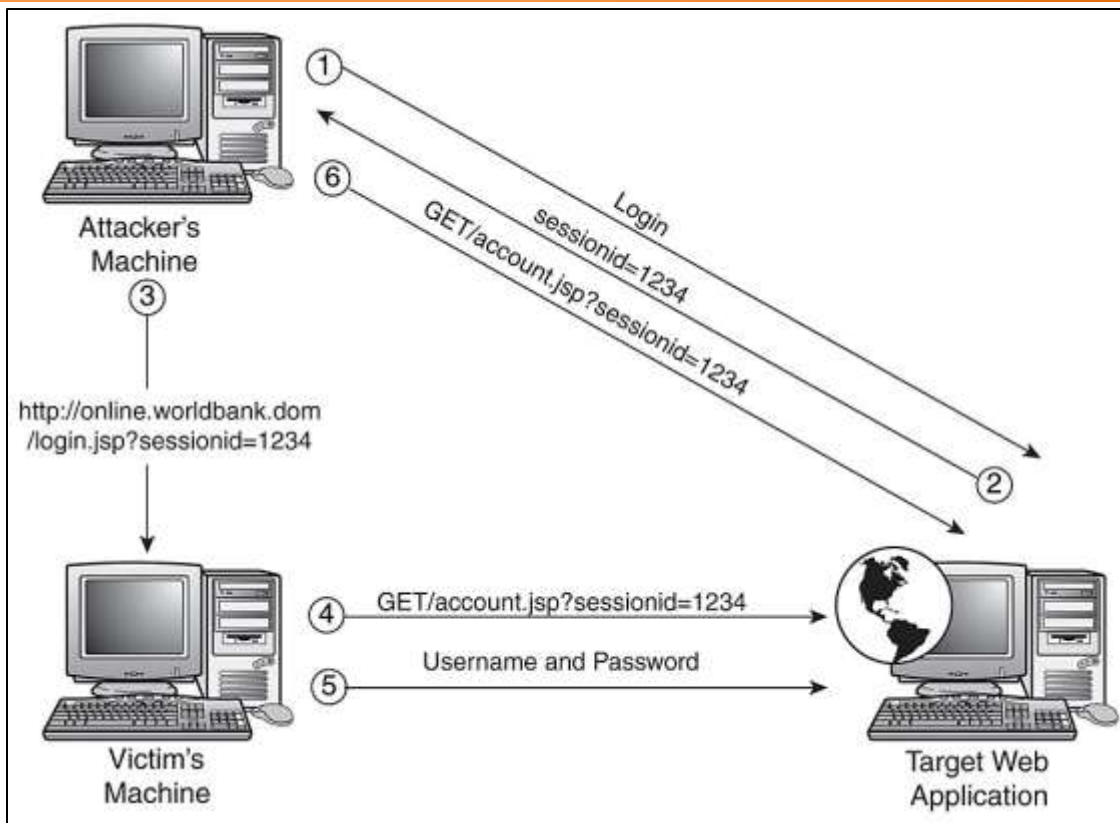
❖ **Nguyên cơ:** Attacker có thể truy cập account người dùng dựa vào SessionId bằng cách gửi SessionId hợp lệ cùng link đăng nhập tới người dùng.

❖ Biện pháp:

- Hủy session và sinh mới session sau khi người dùng đăng nhập thành công.
- Xóa bỏ session sau khi logout
- Đặt timeout cho session

03

Các lỗi hổng ATTT trên ứng dụng web



03

Các lỗi hổng ATTT trên ứng dụng web

❖ Session Fixation

```
Users u = usersMgr.loginHandle(username, EncryptionUtils.encryptMD5(password));
if (u != null) {
    session.invalidate();
    session = request.getSession(true);
    session.removeAttribute("USER_SESSION_HANDLE");
    session.setAttribute("USER_SESSION_NAME", u.getUsername());
    generateToken();
    errorStr = "Login thành công!";
    result = "OK";
    return SUCCESS;
} else {
    errorStr = "Sai thông tin đăng nhập!";
    result = "NotOK";
    return SUCCESS;
}
```

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Session Fixation

```
[HttpPost]
public ActionResult Login(models.User user)
{
    if (ModelState.IsValid)
    {
        if (Repositories.IsValid(user.UserName, user.Password))
        {
            Session.Clear();
            Session.Abandon();
            Response.Cookies.Add(new HttpCookie("ASP.NET_SessionId", ""));
            SessionIDManager manager = new SessionIDManager();
            string newSessionId = manager.CreateSessionID(System.Web.HttpContext.Current);
            HttpCookie cookie = new HttpCookie("ASP.NET_SessionId");
            cookie.HttpOnly = true;
            cookie.Value = newSessionId;
            Response.Cookies.Add(cookie);
        }
    }
}
```

03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ Sử dụng Cookie an toàn
- ❖ **Nguy cơ:** Nếu ứng dụng không thiết lập thuộc tính cookie an toàn (HTTPOnly và Secure), attacker có thể tìm cách lấy session cookie của người dùng.
- ❖ **Biện pháp:** Thiết lập thuộc tính “HTTPOnly” cho session cookie. Các website sử dụng HTTPS, cần thiết lập thuộc tính “Secure”.

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Sử dụng Cookie an toàn

❖ Java:

```
response.setHeader("SET-COOKIE", "JSESSIONID=" + sessionid +  
"; HttpOnly");
```

```
response.setHeader("SET-COOKIE", "JSESSIONID=" + sessionid +  
"; secure");
```

Java code

```
<session-config>  
  <cookie-config>  
    <http-only>true</http-only>  
  </cookie-config>  
</session-config>
```

```
<session-config>  
  <cookie-config>  
    <secure>true</secure>  
  </cookie-config>  
</session-config>
```

Web.xml

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Sử dụng Cookie an toàn

❖ .NET:

```
SessionIDManager manager = new SessionIDManager();  
string newSessionId = manager.CreateSessionID(System.Web.HttpContext.Current);  
HttpCookie cookie = new HttpCookie("ASP.NET_SessionId");  
cookie.HttpOnly = true;  
cookie.Value = newSessionId;  
Response.Cookies.Add(cookie);
```

<httpCookies httpOnlyCookies="true" ...>

Web.config



03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ Sử dụng lib – mã nguồn phiên bản cũ
- ❖ **Nguyên cơ:** Các thư viện, mã nguồn phiên bản cũ thường tồn tại các lỗ hổng bảo mật, attacker có thể khai thác tấn công hệ thống
- ❖ **Biện pháp:** Sử dụng các phiên bản mới nhất hoặc phiên bản được khuyến cáo sử dụng. Download tại trang chủ hoặc nguồn tin cậy.

03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ Chuyển tiếp thiếu thẩm tra
- ❖ **Nguy cơ:** Attacker có thể lợi dụng lỗ hổng để chuyển hướng người dùng tới trang chứa mã độc, hoặc lừa người dùng tới các trang giả mạo.
- ❖ **Biện pháp:** Hạn chế việc chuyển hướng dựa vào các biến gửi từ client. Nếu sử dụng, cần validate biến này (sử dụng whitelist).

03

Các lỗi hổng ATTT trên ứng dụng web

- ❖ Kiểm soát ngoại lệ không tốt
- ❖ **Nguyên cơ:** Việc hiển thị thông tin lỗi quá chi tiết giúp ích cho attacker có thông tin về server - ứng dụng hỗ trợ cho việc khai thác lỗi.
- ❖ **Biện pháp:** Tất cả Exception phải được xử lý và lưu vào log để xử lý.

03

Các lỗi hổng ATTT trên ứng dụng web

❖ Kiểm soát ngoại lệ không tốt

```
public String getListUser() {  
    try {  
        if (!session.getAttribute("USER_SESSSION_HANDLE").equals("admin")) {  
            result = "Không có quyền thực hiện action này";  
            return ERROR;  
        }  
        listUsers = usersMgr.getListUsers();  
        listUsers.remove(0);  
        return SUCCESS;  
    } catch (Exception ex) {  
        Logger.getLogger(UserAction.class.getName()).log(Level.SEVERE, null, ex);  
        return ERROR;  
    }  
}
```

03

Các lỗi hỏng ATTT trên ứng dụng web

❖ Kiểm

❖ Stru

```
<error-page>
  <error-code>404</error-code>
  <location>/Error.jsp</location>
</error-page>
<error-page>
  <error-code>505</error-code>
  <location>/Error.jsp</location>
</error-page>
<error-page>
  <exception-type>java.lang.Throwable</exception-type>
  <location>/Error.jsp</location>
</error-page>
<error-page>
  <exception-type>java.lang.NullPointerException</exception-type>
  <location>/Error.jsp</location>
</error-page>
```

03

Các lỗi hỏng ATTT trên ứng dụng web

- ❖ Kiểm soát ngoại lệ không tốt
- ❖ .NET

```
<customErrors mode="On">  
  <error statusCode="404" redirect="/error/show" />  
  ...  
</customErrors>
```

03

Các lỗ hổng ATTT trên ứng dụng web

- ❖ Sử dụng captcha an toàn
- ❖ **Nguy cơ:** Với các chức năng quan trọng, attacker có thể sử dụng công cụ tự động đến khi đạt được mục đích.
- ❖ **Biện pháp:** Sử dụng Captcha cho các chức năng quan trọng này.

04

Thảo luận và hỏi đáp



THANK YOU! ☺



TRUNG TÂM GIẢI PHÁP CÔNG NGHỆ THÔNG TIN & VIỄN THÔNG VIETTEL

Hãy nói theo cách của bạn

4/25/2015

62