**فصل سوم**

بررسی کاربردهای پیشینه تحقیق

مقدمه

هدف این پایان‌نامه، تشخیص کدهای آسیب‌پذیر در برنامه‌های کاربردی تحت وب است، که مهاجمان با تغییر و دست‌کاری این کدها، حملات XSS را طراحی نموده و به‌وسیله آن‌ها به سرورهای میزبان این صفحات، نفوذ کرده و اقدامات مخربی را انجام می‌دهند.

الگوریتم پیشنهادی، کد برنامهٔ کاربردی را دریافت کرده و با استفاده از فازر تقویتی، تک‌تک خطوط برنامه را مورد جستجو قرار می‌دهد و در صورت وجود آسیب‌پذیری، کد آسیب‌پذیر را مشخص نموده و به برنامه‌نویس اعلام می‌کند. با توجه به گستردگی نوع حملات و پیچیدگی فراوان برنامه‌های کاربردی تحت وب این تحقیق با مسائل غیرقابل‌حل و سختی روبرو خواهد بود. ازآنجاکه راه‌حل پیشنهادی باید برای تمامی حالت‌های ممکنه جواب قابل‌قبول داشته باشد، از الگوریتم‌های تقویتی درروش پیشنهادی استفاده‌شده است. برای این کار توسط فازر تقویتی که با استفاده از الگوریتم تقویتی ABC\_RL ب طراحی‌شده است، کد برنامه با الگوهای آسیب‌پذیر مقایسه شده و درصورتی‌که با یکی از این الگوها مطابقت داشته باشد، کد موردنظر را به‌عنوان یک آسیب‌پذیری نمایش داده و به اطلاع برنامه‌نویس می‌رساند. در شکل زیر فلوچارت الگوریتم پیشنهادی را مشاهده می‌کنید.

**مجموعه داده‌های مورداستفاده**

فازر طراحی‌شده در این پایان‌نامه نیاز به ورودی‌هایی دارد که از آن‌ها برای آموزش الگوریتم استفاده‌شده در هسته خود جهت شناسایی کدهای آسیب‌پذیر در برنامه کاربردی موردبررسی استفاده نماید. این ورودی‌ها که درواقع الگوهای آسیب‌پذیری هستند که در پَروَنجا XSS\_dataset1\_engineered ذخیره‌شده‌اند.این مجموعه داده از مجموعه داده‌های به اشتراک گذاشته در پایگاه داده دانشگاه کارنگی ملون دریافت شده است. اما ازآنجاکه همه ویژگی‌هایی که در این مجموعه داده وجود دارد دارای اهمیت یکسانی نیستند. بنابراین با حذف ویژگی‌های نامربوط، تکراری و کم‌اهمیت با کمک نرم‌افزار Weka ابعاد مسئله کاهش می‌دهم. بدین ترتیب که در فاز پیش‌پردازش سعی می‌نماییم تا ویژگی‌های بااهمیت‌تر رابین 64 ویژگی استخراج‌کنیم.

**روش پیشنهادی**

پیاده‌سازی الگوریتم، روش پیشنهادی ما توسط زبان پایتون انجام‌شده است. درروش ارائه‌شده، یک فازر تقویتی با هستهٔ مرکزی الگوریتم ABC\_RL برای شناسایی کدهای آسیب‌پذیر طراحی‌شده است درروش پیشنهادی ما علاوه بر انتخاب ویژگی‌های موثر در پیش‌بینی آسیب‌پذیری، وزن هر یک از ویژگی‌ها یافت می‌شود. هدف انتخاب ویژگی‌هایی است که صحت طبقه‌بندی را بهبود بخشد. انتخاب ویژگی به‌منظور حذف ویژگی‌های نامناسب در مجموعه داده‌ها است. چراکه انتخاب ویژگی‌های نامناسب باعث بروز خطا در کلاسه‌بندی ویژگی‌ها می‌شود. در انتخاب ویژگی‌ها، زیرمجموعه‌ای از کل ویژگی‌ها، انتخاب می‌گردد. استفاده از روش انتخاب ویژگی باعث کاهش زمان آموزش، کاهش زمان محاسباتی و افزایش قابلیت تعمیم کلاسه‌بندی می‌شود. یک الگوریتم انتخاب ویژگی از یک روش جستجو برای انتخاب زیرمجموعه‌ای از ویژگی‌ها و یک معیار ارزیابی برای امتیازدهی به این زیرمجموعه استفاده می‌کند. در ساده‌ترین الگوریتم، تمام زیرمجموعه‌های ممکن ویژگی‌ها موردبررسی قرارگرفته و زیرمجموعه‌ای با کمترین مقدار نرخ خطای کلاسه‌بندی انتخاب می‌شود. جستجوی کامل فضای ویژگی‌ها دارای بار محاسباتی بالایی است. در این تحقیق از الگوریتم‌های تقویتی برای انتخاب ویژگی استفاده‌شده است. درروش پیشنهادی برای مشخص کردن ارزش و نقش هر یک از ویژگی‌ها در تشخیص کد آسیب‌پذیر به‌طور تصادفی برای هر ویژگی‌ها وزن صفر یا یک اختصاص داده می‌شود که نشان‌دهنده انتخاب یا عدم انتخاب ویژگی است. مقادیر وزن‌دار ویژگی‌ها به‌عنوان ورودی به الگوریتم‌های کلاسه‌بندی داده‌شده و توسط الگوریتم ABC\_RL بهینه می‌گردد.



**الگوریتم ABC\_RL پیشنهادی**

الگوریتم ABC مبتنی بر RL پیشنهادی در این بخش معرفی می‌شود. در ABC\_RL، RL در فاز زنبور کارگر گنجانده‌شده است تا تعداد ابعادی که باید هر بار به‌روز شوند ($nb\_{UP}$) را تنظیم کند از سوی دیگر استراتژی جستجو در زنبورهای ناظر را با جمع‌آوری اطلاعات از بهترین‌های جهانی و دو همسایه بهبود می‌یابد و از اعداد تصادفی بر اساس توزیع دم‌سنگین به‌عنوان عامل مقیاس در استراتژی‌های جستجو الگوریتم بهبودیافته استفاده می‌شود.

**عامل مقیاس بر اساس heavy-tailed distribution**

تصادفی بودن تأثیر نسبتاً زیادی بر تنوع الگوریتم‌های فرا ابتکاری دارد. در ABC پایه، پارامتر 𝜙 یک عدد تصادفی توزیع‌شده یکنواخت در محدوده [1-، 1] است. درواقع، این محدوده سفت‌وسخت نسبتا کوچکی است، که کارایی جستجو را محدود می‌کند. علاوه بر این، زمانی که ABC در بهینه محلی به دام می‌افتد، این تعریف ممکن است کمک بسیاری نماید.

پرواز لوی یک **heavy-tailed distribution است درواقع** قدم زدن تصادفی است که در آن طول گام‌ها از توزیع لوی پیروی می‌کند. توزیع لوی که توسط پائو پیر لوی ارائه ش‍د، یکی از معدود توزیع‌هایی است که پایدار بوده و دارای توابع چگالی احتمال تحلیلی هست] 25[ به‌منظور یپاده‌سازی و استفاده از این توزیع، تولید اعداد تصادفی با پرواز لوی از دو مرحله تشکیل می‌شود که شامل انتخاب جهت تصادفی و تولید اندازه هر گام که از توزیع لیوی پیروی کند، است. تولید جهت می‌تواند از توزیع یکنواخت باشد. برای تولید گام از توزیع لوی، چند راه وجود دارد. یکی از راه‌های کارآمد و آسان استفاده از الگوریتم مانتگنا برای توزیع پایدار و متقارن لوی است. البته منظور از متقارن این است که طول گام‌ها می‌تواند مثبت یا منفی باشد] 25[ مطالعات نشان داده است که پرواز لوی می‌تواند کارایی فر آیند جستجو در شرایط عدم اطمینان را بیشینه نمایید [27.]در واقع پرواز لوی در رفتار حیواناتی نظیر کرم میوه، نوعی مرغابی و میمون‌های عنکبوتی در جستجوی غذا، مشاهده شده است. به‌علاوه توزیع لوی کاربردهای زیادی در پدیده‌های فیزیکی مانند انتشار مولکولی فلوروسنت تحت شرای مطلوب دارد. تولید جواب تصادف(t+1) X بدین صورت است، که ابتدا اعداد تصادفی را از توزیع احتمال لوی تولید کرده، سپس در فضای جواب‌های شدنی، جوابی را با استفاده از آن عدد تصادفی همانند معادله زیر تولید می‌نمائیم.

|  |  |
| --- | --- |
|  | $$x\left(t+1\right)=α×Levy(β)×(X(t)-best)$$ |

که در آن α>0 طول گام بوده که متناسب با مقیاس مسئله و اندازه فضای جواب انتخاب می‌شود و در بیشتر موارد نیز می‌توان آن را مساوی ی قرار داد. رابطه بالا در اصل یک رابطه تصادفی برای قدم زدن یا پرواز تصادفی است. در واقع یک زنجیره مارکوف است که جواب بعدی در آن فقط به جواب جاری بستگی دارد. برخی از جواب‌های جدید با پرواز لوی در همسایگی بهترین جوابی best که در هر مرحله به دست می‌آید انتخاب می‌شوند، تا جستجوی محلی سرعت بیشتری داشته باشد. همچنین، بخشی از جواب‌های جدید را باید در میدانی دورتر از بهترین جواب‌های جاری به تصادف انتخاب کرد، تا فرآیند جستجو در دام بهینه‌های محلی نیفتد.

در این مقاله، از توزیع دم‌سنگین برای کمک به انتخاب تصادفی بودن استفاده شده‌است. توزیع Mittag Leffler، یکی از رایج‌ترین توزیع‌های دم‌سنگین، در الگوریتم پیشنهادی برای تولید فاکتورهای مقیاس در معادلات جستجوی راه‌حل استفاده می‌شود. تعریف توزیع Mittag Leffler به شرح زیر است. گفته می‌شود که یک متغیر تصادفی مشمول توزیع Mittag-Leffler می‌شود که توابع توزیع آن به شکل زیر باشد]28 [

|  |  |
| --- | --- |
| $$F\_{β}(x)=\sum\_{k=1}^{\infty }\frac{(-1)^{k-1}x^{kβ}}{Γ(1+kβ)}$$ | 7 |

|  |  |
| --- | --- |
| $$τ\_{β}=-γ×lnu\left(\frac{\sin(\left(βπ\right))}{\tan(\left(βπυ\right))}-cos⁡(βπ)\right)^{^{1}/\_{β}}$$ | 8 |

عامل تصادفی 𝜏 تولید شده از طریق معادله. (8) در هر دو فاز زنبور عسل کارگر و فاز زنبور ناظر پذیرفته شده است. و برای برازش مقیاس مسئله، 𝜏 با یک ضریب ضرب می‌شود. با توجه به مسئولیت متفاوت دو فاز، ضرایب مربوطه به‌طور متفاوت تعیین می‌شود. 0.07×𝜏 در فاز زنبور کارگر استفاده می‌شود در حالی که فاز زنبور ناظر از 0.05 × 𝜏 استفاده می‌کند. فاز زنبور ناظر قرار است به صورت محلی مورد بهره برداری قرار گیرد، بنابراین مقدار ضریب آن برای جلوگیری از پرش از روی بهینه نسبتاً کوچک است.

**فاز زنبور عسل کارگر به کمک یادگیری تقویتی**

* **استراتژی جستجوی تفاضلی**

درواقع، روشی که معادله جستجو اطلاعات مفید را جمع‌آوری می‌کند، می‌تواند تأثیر زیادی بر اثربخشی جستجوی آن داشته باشد. همانطور که قبلاً ذکر شد، معادله جستجوی ABC استاندارد برای جمع‌آوری اطلاعات محدود در یک زمان پیدا شد. اما همانطور که دیدیم این الگوریتم تنها به بررسی یک بعد یک همسایه هنگام تولید یک راه‌حل امکان پذیر جدید می‌پردازد. توجه داشته باشید که با توجه به بررسی‌های انجام شده در الگوریتم‌های ABC و DE نشان داده شده که الگوریتم DE قادر به جستجو و همگرایی کارآمدتر از الگوریتم ABC است زیرا معادله جستجوی راه‌حل DE امکان بیشتری برای جمع‌آوری اطلاعات مفید از سایر دسته‌ها دارد. از این رو این الگوریتم برای تعریف$nb\_{up}$ انعطاف بیشتری دارد. از آن‌جا که اپراتور جهش الگوریتم DE در ابتدا امر دسته‌هارا در ابعاد کلی به روزرسانی کرده پیام‌هایی را از نسل قبلی و آخرین جهش جمع‌آوری کرده و سپس به اعضا اجازه انتخاب می‌دهد. معادله (9) یک نسخه اساسی از استراتژی جستجوی DE، "rand/1/bin" را ارائه می دهد.

|  |  |
| --- | --- |
| 9 | $$u\_{i.j}=\left\{\begin{array}{c}x\_{r\_{1}j}+F×\left(x\_{r\_{2}j}-x\_{r\_{3}j}\right). if rand\leq C R or j=j\_{rond}\\x\_{i.j} \&otherwise\end{array}\right.$$ |

برای به اشتراک گذاری بهتر اطلاعات در داخل کلونی، یک پارامتر 𝑀 برای تعیین تعداد متغیرهایی که باید به‌روز شوند، اتخاذ شده است ]29[. و با این‌که محدوده‌ای را برای تولید تصادفی تعداد اجزایی که باید اصلاح شوند تعریف شده با این حال، بیشتر پیشرفت‌ها به تنظیمات پارامترها بستگی دارد اما تقریباً غیرممکن است که یک تنظیم ثابت را تعیین کنید که تمام مشکلات را برآورده کند]30[. در این پایان‌نامه سعی خواهد شد که در به بهبود دو جنبه از جنبه‌های جستجو در فاز زنبور عسل کارگر یعنی مقدار اطلاعاتی که از دسته‌ها به دست می‌آید و همچنین$ nb\_{ up}$پرداخته شود و سعی مینماید که از الگوریتمRL برای تنظیم$ nb\_{ up}$ استفاده کند.

|  |  |
| --- | --- |
| 10 | $$υ\_{i.j}=x\_{k\_{1.}j}+c\_{1}×(x\_{k\_{2}.j}-x\_{k\_{3.}j})$$ |

**تنظیم پارامتر**$ nb\_{ up}$ **با Q-learning**

در این قسمت، روش استفاده از یادگیری Q برای تعریف$ nb\_{ up}$ در طول فرآیند جستجو شرح داده‌ می‌شود. برای حل مسائل در مقیاس‌های مختلف،$ nb\_{ up}$ با تغییر نسبت و مقادیر ممکن پارامتر$d\_{ ratio}$ را می‌توان از مجموعه {0.1,0.2,….,0.9} در نظر گرفت.

مجموعه اکشن مربوط به تنظیم مقدار$d\_{ ratio}$ به شکل زیر تعریف می‌شود.



 همانطور که قبلاً ذکر شد، پاداش انجام اقدام 𝑎 در حالت 𝑠 به‌عنوان مقدار 𝑄(𝑠,𝑎) در جدول 𝑄 ثبت می‌شود. در الگوریتم پیشنهادی، هر زنبور کارگر یک عمل را با توجه به جدول 𝑄 مرتبط خود انتخاب می‌کند و سپس انجام عمل روی$d\_{ ratio }$، $ nb\_{ up}$ به صورت زیر محاسبه می شود:

|  |  |
| --- | --- |
| 10 | $$nb\_{up}=ceil(d\_{ratio}×D)$$ |

متغیرهای$ nb\_{ up}$ به‌طور تصادفی از بین متغیرهای 𝐷 برای اصلاح انتخاب می‌شوند. پس از آن، راه‌حل کاندید جدید از طریق معادله بالا تولید می‌شود و با توجه به نتایج مقایسه بین راه‌حل جدید و قبلی، پاداش داده می‌شود و بعد از آن وضعیت به‌روز می‌شود. آخرین مرحله Q-learning به‌روز رسانی مقدار 𝑄 مربوطه از طریق معادله (6) بدست می‌آید توجه داشته باشیم برای جلوگیری از تاثیر گذاری زنبورها بر هم هر یک از آن‌ها دارای جدول 𝑄 مربوط به خود است.

**مجموعه دسته‌ها و پاداش**

از آنجایی که هدف به حداقل رساندن مقدار توابع هدف است، از این رو دو حالت با توجه به نتیجه به‌روز رسانی‌ها تعریف می‌شود.

$$\left\{\begin{array}{c}s1: f\left(vi\right)< f\left(xi\right) \\s2: f(vi) ⩾ f(xi)،\end{array}\right.$$

S1راه‌حل جدید تولید شده دارای عملکرد بهتری از قبلی است.

S2راه‌حل جدید تولید شده از عملکرد بهتری برخوردار نیست.

**انتخاب استراتژی**

سعی کنید همیشه استراتژی با بیشترین برآورد 𝑄 انتخاب کنید. با این حال، انتخاب حریصانه ممکن است استراتژی خاصی را با پتانسیل بهتر نادیده بگیرد. برای غلبه بر این ضعف، 𝜖 − 𝑔𝑟𝑒𝑒𝑑𝑦 پیشنهاد می‌شود که گه‌گاه استراتژی با مقادیر کوچک‌تر انتخاب کنید، در حالی که در بیشتر مواقع هنوز هم اعمال حریصانه انتخاب می‌شوند و 𝜖 به‌عنوان میزان حریص نامگذاری شده‌است. این مکانیسم به روش یادگیری Q اجازه می‌دهد تا اکتشاف و بهره برداری را به خوبی متعادل کند. ]31[ 𝜖 − 𝑔𝑒𝑒𝑑𝑦 به صورت زیر تعریف می‌شود.

|  |  |
| --- | --- |
| 12 | $$π\left(s\_{t}.a\_{t}\right)=\left\{\begin{array}{c} maxQ\left(s\_{t}.a\right). if 1-ϵ\geq rand\\a\_{rand otherwise}\end{array}\right.$$ |

که در آن 𝑟𝑎𝑛𝑑 یک عدد تصادفی در بازه [0، 1] است و $a\_{rand}$ به صورت تصادفی از میان مجموعه {$a\_{1}.….a\_{q}$ { انتخاب می‌شود. $nb\_{up}$ را با توجه به‌روز رسانی تغییر پیدا می‌کند. شبه کد فاز زنبور کارگر اصلاح شده در الگوریتم 3 نمایش داده‌شده است. برای هر راه‌حل نامزد، ابتدا یک عمل بر اساس 𝑄-table آن انتخاب می‌شود (خط 4). از این رو، ابعادی که به‌روز می‌شوند را می‌توان از طریق خطوط5-6 تعیین کرد. پس از به‌روز رسانی راه‌حل‌های نامزد، وضعیت و ارزش پاداش مربوط به عمل بر اساس نتیجه انتخاب حریصانه (خطوط 10-17) داده می‌شود. در پایان، 𝑄-table همانطور که در خط 18 بیان شده است به‌روز می‌شود. مقداردهی اولیه پارامتر $d\_{ratio}$ است.



از سوی دیگر باید بدانید که با استفاده از الگوریتم RL، الگوریتم پیشنهادی نه تنها می‌تواند$nb\_{up}$ را بزرگ کند، بلکه می‌توان مقدار پارامتر را با در نظر گرفتن یادگیری آنلاین بدون نیاز به تعریف مقدار اولیه توسط کاربران تعیین نمود.

**فاز زنبور عسل ناظر**

با در نظر گرفتن ضعف الگوریتم پایه ABC، الگوریتم ABC\_RL پیشنهادی ما به‌عنوان یک راه‌حل پیشرفته در فاز زنبور ناظر استفاده می‌شود. در استراتژی جستجوی پیشنهادی معادله (13)، اطلاعات بهترین راه‌حل جهانی و همچنین دو همسایه در نظر گرفته شده است. علاوه بر این، ضریب مقیاس نیز با توزیع Mittag-Leffler به‌منظور افزایش تصادفی تولید می‌شود. در همین حال، عامل‌ سوم به‌عنوان یک عدد تصادفی در محدوده [1و0] تعریف می‌شود تا به راه‌حل جدید تولید شده کمک نمامید تا از بهترین راه‌حل جهانی به‌طور پایدار یاد بگیرد.

|  |  |
| --- | --- |
| 13 | $$υ\_{i.j}=x\_{i.j}+c\_{2}×τ×\left(x\_{k\_{1}.j}-x\_{k\_{2}.j}\right)+θ×(x\_{best.j}-x\_{i.j})$$ |

**چارچوب الگوریتم ABC\_RL**

به‌منظور توضیح کل فرآیند الگوریتم پیشنهادی، چارچوب آن و نمودار جریان ABC\_RL به ترتیب در الگوریتم 4 نشان داده‌شده است. در ابتدا، راه‌حل‌های نامزد و پارامترهای Q-learning مقداردهی اولیه می‌شوند. سپس در هر تکرار، فاز زنبور به کارگر با RL ابتدا برای به‌روز رسانی راه‌حل‌های کاندید اجرا می‌شود. برای هر راه‌حل نامزد، Q-learning برای تعریف و تنظیم تعداد ابعاد در حال به‌روز رسانی اتخاذ می‌شود. و مشابه روش سنتی ABC، فاز زنبور پیشاهنگ و فاز زنبور ناظر پس از آن اجرا می‌شوند. توجه داشته باشید که به جز پارامترهای کنترل اصلی ABC، پارامترهای 𝛼، 𝛾، 𝜖 و ضرایب 1 و 2 توسط کاربران تعریف می‌شوند. ضرایب مقیاس 𝜃 و 𝜏 اعداد تصادفی هستند.





در الگوریتم ABC\_RL هر زنبور که کلاسه‌بندی را با خطای پیش‌بینی کمتری انجام دهد، شایسته‌تر خواهد بود. مراحل الگوریتم تنظیم تابع برازندگی روش پیشنهادی به شرح زیر هست.

1. مجموعه داده‌ها فراخوانی می‌شود.
2. مقادیر ویژگی‌ها نرمال می‌شود.
3. مجموعه‌ای از مقادیر (صفر یا یک) به هریک از ویژگی‌ها اختصاص داده می‌شود.
4. بر اساس ویژگی‌هایی که وزن یک دارند، طبقه‌بندی انجام می‌شود
5. میزان خطای طبقه‌بند محاسبه می‌شود.

**Herminy**

 **Herminy** یک برنامه کاربردی ساده است که در python ساده نوشته شده است که محموله‌های XSS را بر اساس بردارهای تعریف شده توسط کاربر با استفاده از متغیرهایی که با لیست‌های فازی جایگزین می‌شوند تولید می‌کند. این امکان را فراهم می‌کند که فقط بارها را به صورت متن ساده تولید کنید یا آنها را در داخل یک iframe اجرا کنید. در داخل iframe، امکان ارسال درخواست‌های GET یا POST از مرورگر به URL های دلخواه با استفاده از بارهای تولید شده وجود دارد.



**کدهای html فازر ساخته شده که باید به پایتون برگردد و توسط الگورتیم abc\_al بهبود پیدا کند**

<html>

<head>

 <title>XSS Fuzzer</title>

 <meta charset="utf-8" />

 <style>

.displayinline

{

 display: inline-block;

}

.round {

 border: 2px solid #765942;

 border-radius: 10px;

 padding: 10px 10px 10px 10px;

}

table.blueTable {

 border: 1px solid #1C6EA4;

 background-color: #EEEEEE;

 width: 90%;

 text-align: left;

 border-collapse: collapse;

}

table.blueTable td, table.blueTable th {

 border: 1px solid #AAAAAA;

 padding: 3px 2px;

}

table.blueTable tbody td {

 font-size: 13px;

}

table.blueTable thead {

 background: #1C6EA4;

 background: -moz-linear-gradient(top, #5592bb 0%, #327cad 66%, #1C6EA4 100%);

 background: -webkit-linear-gradient(top, #5592bb 0%, #327cad 66%, #1C6EA4 100%);

 background: linear-gradient(to bottom, #3c4b56 0%, #327cad 66%, #1C6EA4 100%);

 border-bottom: 2px solid #444444;

}

table.blueTable thead th {

 font-size: 15px;

 height: 40px;

 font-weight: bold;

 color: #FFFFFF;

 border-left: 2px solid #D0E4F5;

}

table.blueTable thead th:first-child {

 border-left: none;

}

/\* https://purecss.io/buttons/ \*/

.button-success {

 color: white;

 font-size: 12pt;

 border-radius: 4px;

 margin: 5px;

 padding-left: 10px;

 padding-right: 10px;

 text-shadow: 0 1px 1px rgba(0, 0, 0, 0.2);

 background: #327cad; /\* this is a green \*/

}

.button-simple {

 color: white;

 font-size: 12pt;

 border-radius: 4px;

 text-shadow: 0 1px 1px rgba(0, 0, 0, 0.2);

 background: #327cad; /\* this is a green \*/

}

 </style>

 <script>

 // Class for fuzzing lists

 class FuzzList {

 constructor(listname, textvalues) {

 this.name = listname;

 this.list = textvalues;

 }

 }

 // Array of fuzzing lists

 fuzzing\_array = [];

 // Default fuzzing lists

 var htmllist = new FuzzList("[TAG]", "img\nsvg");

 fuzzing\_array.push(htmllist);

 var eventlist = new FuzzList("[EVENT]", "onerror\nonload");

 fuzzing\_array.push(eventlist);

 var attrlist = new FuzzList("[ATTR]", "src\nvalue");

 fuzzing\_array.push(attrlist);

 // Array Remove - By John Resig (MIT Licensed)

 Array.prototype.remove = function(from, to) {

 var rest = this.slice((to || from) + 1 || this.length);

 this.length = from < 0 ? this.length + from : from;

 return this.push.apply(this, rest);

 };

 // Initialization

 function Initialization() {

 for (var i = 0; i < fuzzing\_array.length; i++) {

 var opt = document.createElement('option');

 opt.value = fuzzing\_array[i].name;

 opt.innerHTML = fuzzing\_array[i].name;

 var select = document.getElementById("fuzzing");

 select.appendChild(opt);

 }

 }

 // Save new fuzzing list

 function SaveFuzzingList() {

 var myselect = document.getElementById("fuzzing");

 if(myselect.selectedIndex == -1) return;

 var listname = myselect.options[myselect.selectedIndex].value;

 var list\_data = document.getElementById("fuzz\_list\_data").value;

 // Check if the list is already there

 var found = false;

 for (var i = 0; i < fuzzing\_array.length; i++) {

 if (fuzzing\_array[i].name == listname) {

 fuzzing\_array[i].list = list\_data;

 found = true;

 }

 }

 if(!found)

 {

 var new\_list = new FuzzList(listname, list\_data);

 fuzzing\_array.push(new\_list);

 }

 }

 // On changing the fuzzing list

 function SelectChanged() {

 var myselect = document.getElementById("fuzzing");

 if(myselect.selectedIndex == -1) return;

 var list\_name = myselect.options[myselect.selectedIndex].value;

 var list\_data\_elem = document.getElementById("fuzz\_list\_data");

 // Parse list data and check value

 var found = false;

 for (var i = 0; i < fuzzing\_array.length; i++) {

 if (fuzzing\_array[i].name == list\_name) {

 list\_data\_elem.value = fuzzing\_array[i].list;

 found = true;

 }

 }

 if (!found) list\_data\_elem.value = "";

 }

 // Function to add a new fuzz list

 function AddPlaceholder() {

 var placeholder = document.getElementById("placeholder").value;

 var opt = document.createElement('option');

 opt.value = placeholder;

 opt.innerHTML = placeholder;

 var select = document.getElementById("fuzzing");

 select.appendChild(opt);

 }

 // Function used to delete a placeholder

 function DeletePlaceholder()

 {

 var myselect = document.getElementById("fuzzing");

 if(myselect.selectedIndex == -1) return;

 var list\_name = myselect.options[myselect.selectedIndex].value;

 myselect.remove(myselect.selectedIndex);

 var list\_data\_elem = document.getElementById("fuzz\_list\_data");

 list\_data\_elem.value = "";

 // Parse list data and check value

 for (var i = 0; i < fuzzing\_array.length; i++) {

 if (fuzzing\_array[i].name == list\_name) {

 fuzzing\_array.remove(i);

 }

 }

 }

 // Get fuzzlist array content by name

 function GetFuzzList(list\_name)

 {

 // Parse list data and check value

 for (var i = 0; i < fuzzing\_array.length; i++) {

 if (fuzzing\_array[i].name == list\_name) {

 var list\_string = fuzzing\_array[i].list;

 list\_string = list\_string.replace("\r\n","\n");

 return list\_string.split("\n");

 }

 }

 return NULL;

 }

 // Function that saves the executed payloads

 function js\_callback(payload)

 {

 alert(unescape(payload));

 }

 // Basic HTML encode

 function html\_entities(str)

 {

 return str.replace(/</g,"&lt;").replace(/>/g,"&gt;").replace(/'/g,"&apos;").replace(/"/g,"&quot;");

 }

 var current\_iframe = 0;

 var current\_payload = 0;

 var process\_interval;

 // If required to use GET/POST request

 var urlmode;

 var urlmethod;

 var get\_url;

 var post\_data;

 var nr\_iframes;

 var selected\_name;

 var save\_payload;

 // Function that processes the results in batches

 function ProcessResultsInBatches(payload\_results)

 {

 current\_iframe = 0;

 // Finished fuzzing

 if(current\_payload == payload\_results.length)

 {

 clearInterval(process\_interval);

 current\_payload = 0;

 alert("Fuzzing session finished!");

 return;

 }

 // Create iframes

 CreateIframes(nr\_iframes);

 // Iterate through payloads

 while(current\_payload < payload\_results.length)

 {

 if(current\_iframe == nr\_iframes)

 {

 current\_iframe = 0;

 break;

 }

 var clean\_payload = payload\_results[current\_payload++];

 var updated\_payload = clean\_payload.replace("[SAVE\_PAYLOAD]", save\_payload);

 // Check run mode

 if(selected\_name == "1")

 {

 var doc = all\_iframes[current\_iframe++].contentWindow.document;

 doc.open();

 doc.write(updated\_payload.replace("[PAYLOAD]",escape(clean\_payload)));

 doc.close();

 }

 else if(selected\_name == "2")

 {

 // GET

 if(urlmethod == "GET")

 {

 all\_iframes[current\_iframe++].src = get\_url.replace("[PAYLOAD]", escape(updated\_payload.replace("[PAYLOAD]",escape(clean\_payload))));

 }

 else

 {

 // Create HTML form

 var form\_content = '<form method="post" id="post\_form" action="' + get\_url.replace("[PAYLOAD]",escape(updated\_payload.replace("[PAYLOAD]",escape(clean\_payload)))) + '">\n';

 // Parse POST data

 var params = post\_data.split("&");

 for(var p = 0; p < params.length; p++)

 {

 var each\_param = params[p].split("=");

 form\_content += '<input type="hidden" name="'

 + html\_entities(each\_param[0].replace("[PAYLOAD]", html\_entities(updated\_payload.replace("[PAYLOAD]",escape(clean\_payload))))) + '" value="'

 + html\_entities(each\_param[1].replace("[PAYLOAD]", html\_entities(updated\_payload.replace("[PAYLOAD]",escape(clean\_payload))))) + '" />\n';

 }

 var doc = all\_iframes[current\_iframe++].contentWindow.document;

 doc.open();

 doc.write(form\_content + '</form><scr' + 'ipt>document.getElementById(\'post\_form\').submit();</scr' + 'ipt>\n');

 doc.close();

 }

 }

 }

 }

 // Function used to create the iframes

 var all\_iframes = [];

 var ifrm;

 function CreateIframes(nr\_iframes)

 {

 // Cleanup

 all\_iframes = [];

 ifrm = document.getElementById("runiframe");

 ifrm.innerHTML = "";

 // Create new iframes

 for(var i = 0; i < nr\_iframes; i++)

 {

 var iframe = document.createElement("iframe");

 iframe.width = 140;

 iframe.height = 140;

 ifrm.appendChild(iframe);

 all\_iframes.push(iframe);

 // Add a <br />

 if( (i + 1) % 10 == 0)

 {

 var br = document.createElement("br");

 ifrm.appendChild(br);

 }

 }

 }

 // Function that does something with the results

 function ProcessResults(payload\_results)

 {

 var runmode = document.getElementById("runmode");

 selected\_name = runmode.options[runmode.selectedIndex].value;

 save\_payload = document.getElementById("save\_payload").value;

 // Print them

 if(selected\_name == "0")

 {

 for(var f = 0; f < payload\_results.length; f++)

 {

 var print\_result = payload\_results[f].replace("[SAVE\_PAYLOAD]", save\_payload).replace(/</g,"&lt;").replace(/>/g,"&gt;");

 print\_result = print\_result.replace("[PAYLOAD]", escape(payload\_results[f]));

 output.innerHTML += print\_result;

 output.innerHTML += "<br />\r\n";

 }

 output.innerHTML += "<br />\r\n";

 }

 // Run in iframes

 if(selected\_name == "1" || selected\_name == "2")

 {

 ifrm = document.getElementById("runiframe");

 nr\_iframes = parseInt(document.getElementById("nr\_iframes").value);

 nr\_delay = parseInt(document.getElementById("nr\_delay").value);

 // If required to use GET/POST request

 urlmode = document.getElementById("urlmode");

 urlmethod = urlmode.options[urlmode.selectedIndex].value;

 get\_url = document.getElementById("get\_url").value;

 post\_data = document.getElementById("post\_data").value;

 // Run in batches

 ProcessResultsInBatches(payload\_results);

 process\_interval = setInterval( function()

 {

 ProcessResultsInBatches(payload\_results)

 }, nr\_delay);

 }

 }

 // Run function

 function Run()

 {

 // Get the payloads

 var payloadstext = document.getElementById("payloads").value;

 payloadstext = payloadstext.replace("\r\n","\n");

 var payloads = payloadstext.split("\n");

 var output = document.getElementById("output");

 // Clean previous output

 output.innerHTML = "<br />\r\n";

 document.getElementById("runiframe").innerHTML = "\r\n";

 // Iterate through payloads

 for(var i = 0; i < payloads.length; i++)

 {

 // Init payload list

 payload\_results = [];

 payload\_results.push(payloads[i]);

 var total\_payloads = 1;

 // Check all placeholders

 for(var j = 0; j < fuzzing\_array.length; j++)

 {

 total\_payloads = payload\_results.length;

 var list\_found = false;

 // Iterate through all previously stored payloads

 for(var p = 0; p < total\_payloads; p++)

 {

 // Check if we have the placeholder in the tag

 if(payload\_results[p].includes(fuzzing\_array[j].name))

 {

 // Navigate through fuzz list

 var fuzz = GetFuzzList(fuzzing\_array[j].name);

 for(var k = 0; k < fuzz.length; k++)

 {

 // Sava a list with all payloads

 var one\_payload = payload\_results[p].replace(fuzzing\_array[j].name, fuzz[k]);

 payload\_results.push(one\_payload);

 }

 }

 else list\_found = true;

 }

 // Remove unused payloads

 if(!list\_found) for(var r = 0; r < total\_payloads; r++) payload\_results.remove(0);

 }

 // Print the results

 ProcessResults(payload\_results);

 }

 }

 // When run mode is changed

 function RunModeChanged()

 {

 var runmode = document.getElementById("runmode");

 var selected\_name = runmode.options[runmode.selectedIndex].value;

 var nr\_iframes = document.getElementById("run\_nr\_iframes");

 var nr\_delay = document.getElementById("run\_nr\_delay");

 var req\_iframes = document.getElementById("request\_iframes");

 var req\_iframes\_url = document.getElementById("request\_iframes\_url");

 var req\_iframes\_data = document.getElementById("request\_iframes\_data");

 // Print them

 if(selected\_name == "0")

 {

 nr\_iframes.style.display = "none";

 nr\_delay.style.display = "none";

 req\_iframes.style.display = "none";

 req\_iframes\_url.style.display = "none";

 req\_iframes\_data.style.display = "none";

 }

 // Run in iframe

 if(selected\_name == "1")

 {

 nr\_iframes.style.display = "";

 nr\_delay.style.display = "";

 req\_iframes.style.display = "none";

 req\_iframes\_url.style.display = "none";

 req\_iframes\_data.style.display = "none";

 }

 // Run in request

 if(selected\_name == "2")

 {

 nr\_iframes.style.display = "";

 nr\_delay.style.display = "";

 req\_iframes.style.display = "";

 req\_iframes\_url.style.display = "";

 req\_iframes\_data.style.display = "";

 }

 }

 </script>

</head>

<body onload="Initialization()">

 <table class="blueTable" align="center">

 <thead>

 <tr align="center">

 <th colspan="3">

 XSS Fuzzer

 </th>

 </tr>

 <tr align="center">

 <th width="50%">

 Payloads

 </th>

 <th width="25%">

 Fuzzing lists

 </th>

 <th width="25%">

 List

 </th>

 </tr>

 </thead>

 <tr align="center">

 <td>

 <textarea class="round" placeholder="Payloads here..." id="payloads" rows="10" cols="80">&lt;[TAG] [ATTR]=Something [EVENT]=[SAVE\_PAYLOAD] /&gt;</textarea>

 </td>

 <td>

 <br />

 <select id="fuzzing" size="10" onchange="SelectChanged()" style="width: 200px;">

 </select>

 <br /><br />

 <input type="button" class="button-simple" value="Delete" onclick="DeletePlaceholder()" />

 <br /><br />

 Placeholder: <input type="text" id="placeholder" value="" />

 <input type="button" class="button-simple" value="Add" onclick="AddPlaceholder()" />

 <br /><br />

 </td>

 <td>

 <textarea placeholder="Select a list" id="fuzz\_list\_data" class="round" rows="10" cols="30"></textarea>

 <br /><br />

 <input type="button" class="button-simple" value="Save" onclick="SaveFuzzingList()" />

 </td>

 </tr>

 <tr>

 <td colspan="2" style="padding: 20px;">

 <table>

 <tr>

 <td style="border:none" width="100">Run mode: </td>

 <td style="border:none" width="400">

 <select id="runmode" onchange="RunModeChanged()">

 <option value="0">Print results</option>

 <option value="1">Run in iframe(s)</option>

 <option value="2">Run in GET/POST data in iframe(s)</option>

 </select>

 </td>

 </tr>

 <tr id="run\_nr\_iframes\_payload">

 <td style="border:none">

 [SAVE\_PAYLOAD]:

 </td>

 <td style="border:none">

 <input type="text" style="width: 300px;" id="save\_payload" value="alert(unescape('[PAYLOAD]'));" />

 </td>

 </tr>

 <tr id="run\_nr\_iframes" style="display: none">

 <td style="border:none">

 Nr. of iframes:

 </td>

 <td style="border:none">

 <input type="text" style="width: 50px;" id="nr\_iframes" value="100" />

 </td>

 </tr>

 <tr id="run\_nr\_delay" style="display: none">

 <td style="border:none">

 Iframes delay:

 </td>

 <td style="border:none">

 <input type="text" style="width: 50px;" id="nr\_delay" value="5000" />

 </td>

 </tr>

 <tr id="request\_iframes" style="display: none">

 <td style="border:none">Request type: </td>

 <td style="border:none">

 <select id="urlmode">

 <option value="GET">GET</option>

 <option value="POST">POST</option>

 </select>

 </td>

 </tr>

 <tr id="request\_iframes\_url" style="display: none">

 <td style="border:none">URL:</td>

 <td style="border:none"><input type="text" style="width: 300px;" id="get\_url" value="" /></td>

 </tr>

 <tr id="request\_iframes\_data" style="display: none">

 <td style="border:none">POST data: </td>

 <td style="border:none"><input type="text" style="width: 300px;" id="post\_data" value="" /></td>

 </tr>

 </table>

 </td>

 <td align="center">

 <input type="button" onclick="Run()" value="Run" class="button-success" />

 </td>

 </tr>

 <tr>

 <td colspan="3" style="padding-left: 20px;">

 <br />Output: <br />

 <div id="runiframe"></div><br />

 <div id="output"></div>

 </td>

 </tr>

 </table>

</body>

</html>

**جمع‌بندی**

در این فصل، الگوریتم پیشنهادی برای تشخیص آسیب‌پذیری، برای کد برنامه‌های کاربردی تحت وب توضیح داده شد. بطور خلاصه، در این روش، ما کد برنامهٔ نوشته شده توسط برنامه‌نویس و الگوهای آسیب‌پذیری را به سیستم پیشنهادی وارد می‌کنیم و در نهایت، مشخص می‌کند چه خطوطی از برنامه ممکن است مورد سوء استفادهی مهاجمان قرار گیرد. روش پیشنهادی، بصورت کاملاً هوشمند با تولید ورودی‌های کاملاً تصادفی برای فازر تقویتی و محاسبهٔ میزان تشابه برنامهٔ کاربردی تحت وب در فضای حالت انواع آسیب‌پذیری‌ها را جستجو کرده و درصورتی‌که میزان تشابه، بیش از؟ درصد باشد، کد موردبررسی را به‌عنوان یک عبارت آسیب پذیر به برنامه‌نویس معرفی می‌کند. البته برنامه‌نویس، آزاد به تغییر این قسمت از کد برنامهٔ خود است. ولی درصورتی‌که این قسمت با عبارات مطمئن‌تر جایگزین نگردد، به احتمال زیاد مهاجمان قادر خواهند بود، با تغییراتی در این قسمت از کد، اقدام به حملات XSS نمایند. درروش پیشنهادی، از الگوریتم ABC\_RL با کمک حرکت تصادفی به‌عنوان قلب سیستم فازینگ تقویتی استفاده کرده و اقدام به شناسایی آسیب‌پذیری‌ها می‌نماید. این روش، نتایجی به مراتب بهتر از سایر روشهای تقویتی که تاکنون ارائه‌شده، بدست آورده است. در این روش علاوه بر اینکه خود روش ABC\_RL کارایی بهتری نسبت به سایر روشهای یادگیری ماشین مثل الگوریتم ژنتیک دارد، مشکل بیشینهٔ محلی، با ایدی حرکت تصادفی برطرف شده است و همین امر کمک شایانی در بهبود نتایج داشته است.