SAIVS

- **S**pider **A**rtificial **I**ntelligence **V**ulnerability **S**canner -

ISAO TAKAESU TAKESHI TERADA

Abstract: SAIVS is an artificial intelligence we developed to detect vulnerabilities in Web applications. The goal of SAIVS is to perform vulnerability assessment like a human security engineer. Currently, our SAIVS prototype demonstrates some human-like reasoning and learning abilities using machine learning algorithms: it recognizes the type of the page, it creates a login credential in the “create account” page to log in, and it crawls the rest of the pages to scan for and report vulnerabilities. When improved SAIVS can be applied to actual security assessments and bug bounty programs.

1. Introduction

SAIVS is an artificial intelligence designed to detect vulnerabilities in Web applications. Developed in January 2016, the SAIVS prototype can crawl simple Web applications that include dynamic pages such as "login," "create account," and "information search" and also report vulnerabilities such as Cross Site Scripting and SQL Injections. This paper will explain how machine learning enabled these abilities.

2. Objectives of SAIVS

SAIVS can crawl simple Web applications using its human-like reasoning skills. Crawling is a process in which one follows and browses the links in an HTML document on a Web application. Consider the case when a person is trying to access the page in Figure 1. You will demonstrate the following thinking pattern:

```
<form action="/cyclone/sessions" method="post">
  <label for="email">Email</label>
  <input id="email" name="email" type="text" />
  <label for="password">Password</label>
  <input id="password" name="password" type="password" />
  <input name="commit" type="submit" value="Sign in" />
</form>
```

Figure 1 Example of FORM tag element

When you see the word "sign in," you **recognize this is a login form**. You realize that you need to prepare a login credential in advance, and if you don’t have a credential yet, you will first need to register and create an account from “create account” page. You will also process that you need to **enter the optimal values** in the input forms specified with the INPUT tags to move to the next page. So, you will enter the appropriate values, in this case an e-mail address and password, in the corresponding input forms. Even by doing so, you may see an error message. In that case, you will **recognize that you have failed to move to the next page** and that you will need to re-enter new, better values in the input forms.

This example shows that crawling requires a complex thought process. Therefore, to perform such reasoning, SAIVS must also demonstrate at least the following thinking patterns.

- Recognize the page type
- Learn the optimal parameter values
- Recognize the success or failure of a page transition

SAIVS uses three machine learning algorithms and acquired the thinking patterns as shown in Table 1.

Thinking pattern	Algorithm
Recognize the page type	Naive Bayes
Recognize the success or failure of a page transition	
Learn the optimal parameter values	Multilayer Perceptron Q-Learning

Table 1 Thinking patterns and algorithms

2.1 Recognizing page contents: Naive Bayes

Naive Bayes is used for text classification. It uses some key information that determines the characteristics of a text and automatically classifies the text into pre-defined categories.

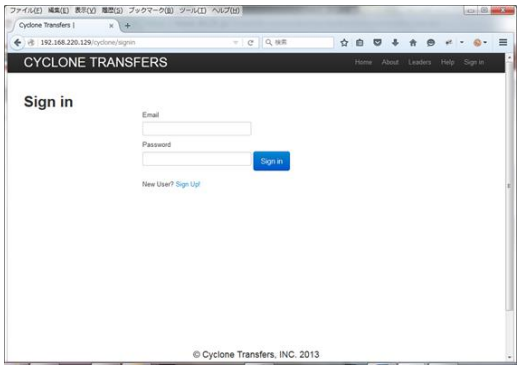

Figure 2 Login page

A person looks at Figure 2 and visually recognizes this is a “login” page because they associate the texts on the page "Sign in," "Email," and "Password" with a “login” page. In their thought process, they link the information that indicates the page type with a category of a page, concluding the page containing these texts must be a “login” page.

SAIVS follows the same logic to identify the page type using Naive Bayes. The information that indicates the page type can be obtained by parsing keywords such as **Sign in**, **Email**, and **Password** in the HTML source of the page. For the pre-defined

categories, we prepared the three page types as shown in Table 2. We also numerically defined the likelihoods of the keywords appearing in each page type as probabilities as listed in Table 3.

```
<h1>Sign in</h1>
<form action="/cyclone/sessions" method="post">
<label for="email">Email</label>
<input id="email" name="email" type="text" />
<label for="password">Password</label>
<input id="password" name="password" type="password" />
</form>
```

Figure 3 HTML source of the login page

Category	Words used for classification
Login page	Email, User ID, Password, Sign in…
Create account page	Email, Password, Confirm, Sign up…
Search page	Word, Text, String, Search…

Table 2 Category table for the page type

Category (Page Type)	Likelihoods of Keywords Appearing		
	Sign in	Email	Password
Login page	90 %	50 %	50 %
Create account page	20 %	50 %	50 %
Search page	5 %	20 %	10 %

Table 3 Probabilities of keywords appearing in each category

SAIVS will recognize the page type using the three keywords, "Sign in," "Email," and "Password" in Figure 3. By using the probabilities of the keywords appearing in a page type in Table 3, the probabilities of all three keywords appearing in each page type are calculated as the following:

- Login : 0.9 x 0.5 x 0.5 x 100 = **22.5**
- Create account : 0.2 x 0.5 x 0.5 x 100 = **5**
- Search : 0.05 x 0.2 x 0.1 x 100 = **0.1**

The probability of category "login" page containing all three keywords is the highest. Therefore, the page that contains the texts "Sign in," Email," and "Password" is determined as the "login page."

The same logic can be applied to classify the page transitions into "failure" or "success" by using the texts in the HTML source that characterize the results of a page transition. Figure 4 shows a HTML source of a failed login page.

```
<div>Invalid email or password </div>
```

Figure 4 HTML source of failed login

2.2 Learning optimal values: Multilayer Perceptron and Q-learning

A multilayer perceptron can model the complex relationships between input and output. Q-Learning can determine the optimal action in a current state using Q-value. Combined, the two algorithms can calculate an optimal parameter value for a page transition. Figure 5 is a model that combines a multilayer perceptron and Q-learning.

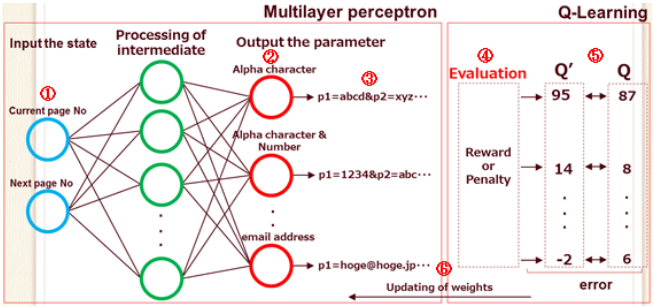

Figure 5 Learning model of the optimal parameter values

Using this model, SAIVS learns the optimal behavior patterns (optimal parameter configuration) to transition to the next page by repeating the following steps (numbers correspond to the numbers in Figure 5):

① Input the current state.
② Output some parameter values.
③ Use the output parameter values to transition to the next page.
④ Observe the success or failure of the page transition.
⑤ Update the Q-value. Calculate error for the Q-value before and after the update.
⑥ Let multilayer perceptron learn, so that error is minimized.

Initially, SAIVS outputs random parameter values but gradually starts to output better parameter values as it learns from the results of the "failed" and "successful" transitions. By the trial-and-error process, SAIVS eventually learns to create an appropriate credential and use it to log in the application.

3. Demonstration

The target Web application is Cyclone (OWASP BWA) which contains dynamic “login,” “create account,” and “search” pages.

Please visit the following link to watch the demo video:

https://www.mbsd.jp/blog/img/20160113_4.mp4

The command prompt at the upper left corner of the video is the console that is running SAIVS. It scans Cyclone while crawling it and detects an SQL injection vulnerability in the "search" page after logging in.

4. Future Prospects

With enhanced scanning and page transitioning abilities, SAIVS can be adopted to real world businesses such as Web application vulnerability assessments and bug bounty programs. We plan to incorporate Natural Language Processing in the AI to speed up the learning processes. Our goal for SAIVS is to detect vulnerabilities that require human perception to identify.

Contact us

Email : isao.takaesu.eq@d.mbsd.jp