



Software Architecture and Design

Open-Source Analysis Assignment

Zed Attack Proxy (ZAP)

เสนอ

อ.ปริญญา เอกปริญญา

สมาชิก

62010453	นนทพันธุ์ รุจิรกาล
62010472	นวพรรษ ศรีบุญเรือง
62010474	นวพล กรุดพันธ์
62010494	นิติพัฒน์ บุญเกตุ
62010496	นิติภูมิ คล้ายเนียม

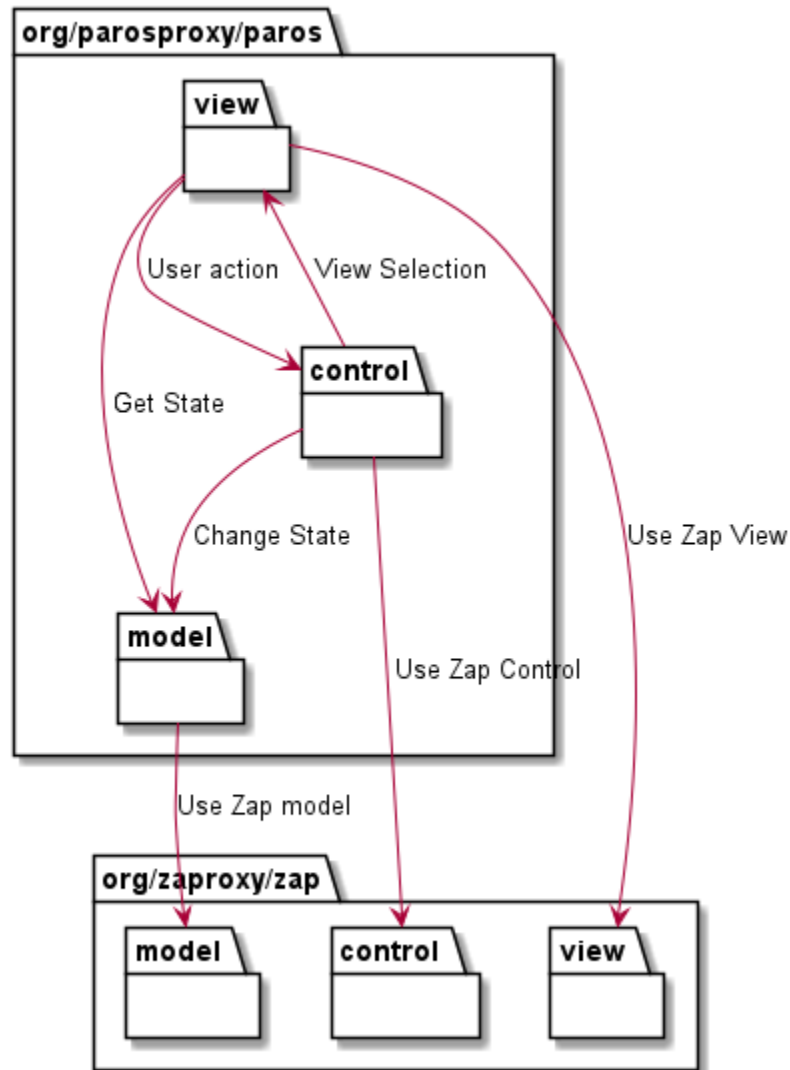
รายงานนี้เป็นส่วนหนึ่งของรายวิชา 01076024

SOFTWARE ARCHITECTURE AND DESIGN ภาคเรียนที่ 1 ปีการศึกษา 2564

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Zed Attack Proxy (ZAP)

GitHub Project Repository: <https://github.com/zaproxy/zaproxy>

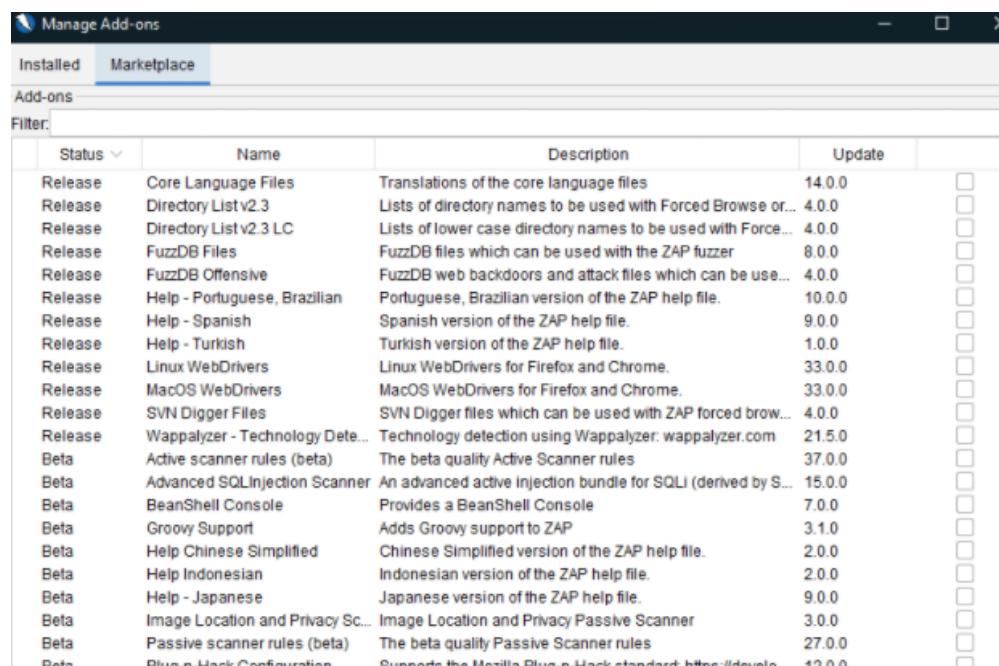


Model , **View** และ **Control** เป็น Module โดย **Model** จะทำงานด้วยการจัดการ State และมี **View** ที่ติดต่อกับ User โดยที่จะรับ State มาจาก **Model** และแสดงผลให้ User และ **Control** รับ User Action มาจาก **View** และจัดการกับ State ของ **Model**

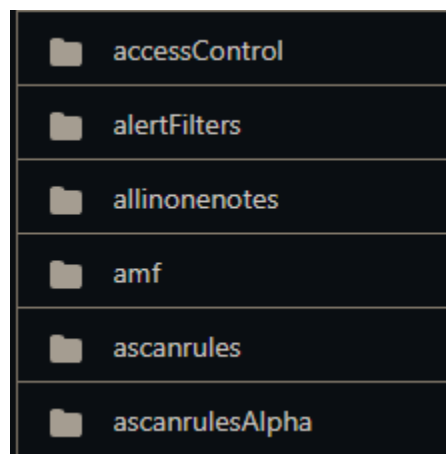
Quality Attributes

1. Usability

- Tactic : AddOns
- Evidences :



Status	Name	Description	Update	
Release	Core Language Files	Translations of the core language files	14.0.0	<input type="checkbox"/>
Release	Directory List v2.3	Lists of directory names to be used with Forced Browse or...	4.0.0	<input type="checkbox"/>
Release	Directory List v2.3 LC	Lists of lower case directory names to be used with Force...	4.0.0	<input type="checkbox"/>
Release	FuzzDB Files	FuzzDB files which can be used with the ZAP fuzzer	8.0.0	<input type="checkbox"/>
Release	FuzzDB Offensive	FuzzDB web backdoors and attack files which can be use...	4.0.0	<input type="checkbox"/>
Release	Help - Portuguese, Brazilian	Portuguese, Brazilian version of the ZAP help file.	10.0.0	<input type="checkbox"/>
Release	Help - Spanish	Spanish version of the ZAP help file.	9.0.0	<input type="checkbox"/>
Release	Help - Turkish	Turkish version of the ZAP help file.	1.0.0	<input type="checkbox"/>
Release	Linux WebDrivers	Linux WebDrivers for Firefox and Chrome.	33.0.0	<input type="checkbox"/>
Release	MacOS WebDrivers	MacOS WebDrivers for Firefox and Chrome.	33.0.0	<input type="checkbox"/>
Release	SVN Digger Files	SVN Digger files which can be used with ZAP forced brow...	4.0.0	<input type="checkbox"/>
Release	Wappalyzer - Technology Dete...	Technology detection using Wappalyzer: wappalyzer.com	21.5.0	<input type="checkbox"/>
Beta	Active scanner rules (beta)	The beta quality Active Scanner rules	37.0.0	<input type="checkbox"/>
Beta	Advanced SQLInjection Scanner	An advanced active injection bundle for SQLi (derived by S...	15.0.0	<input type="checkbox"/>
Beta	BeanShell Console	Provides a BeanShell Console	7.0.0	<input type="checkbox"/>
Beta	Groovy Support	Adds Groovy support to ZAP	3.1.0	<input type="checkbox"/>
Beta	Help Chinese Simplified	Chinese Simplified version of the ZAP help file.	2.0.0	<input type="checkbox"/>
Beta	Help Indonesian	Indonesian version of the ZAP help file.	2.0.0	<input type="checkbox"/>
Beta	Help - Japanese	Japanese version of the ZAP help file.	9.0.0	<input type="checkbox"/>
Beta	Image Location and Privacy Sc...	Image Location and Privacy Passive Scanner	3.0.0	<input type="checkbox"/>
Beta	Passive scanner rules (beta)	The beta quality Passive Scanner rules	27.0.0	<input type="checkbox"/>
Beta	Plug-n-Hack Configuration	Supports the Mozilla Plug-n-Hack standard: https://develo...	12.0.0	<input type="checkbox"/>

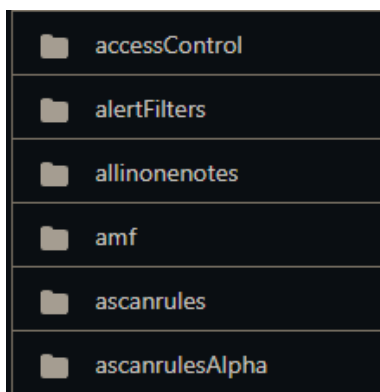
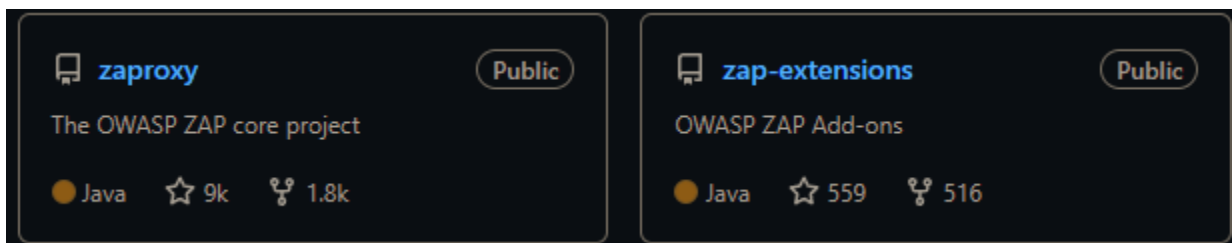


Ref : [zaproxy/zap-extensions: OWASP ZAP Add-ons \(github.com\)](https://github.com/zaproxy/zap-extensions)

- Reason : เพิ่มความหลากหลายในการใช้งาน Tool ให้ตรงตามจุดประสงค์การใช้งานของผู้ใช้ ผู้พัฒนาสามารถพัฒนา Addon ที่เหมาะสมกับการใช้งานของตนเองได้ และส่ง Pull request ให้ผู้พัฒนา Tool หลักได้

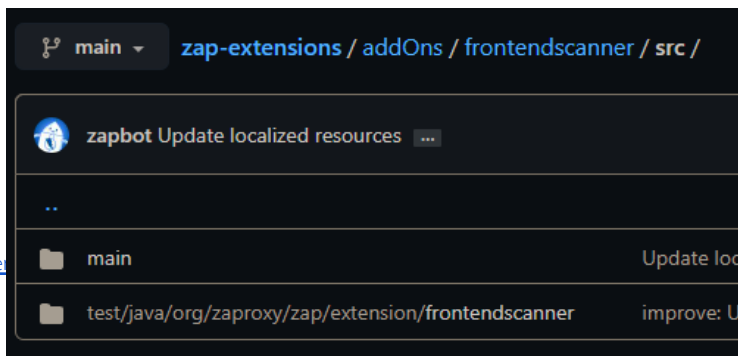
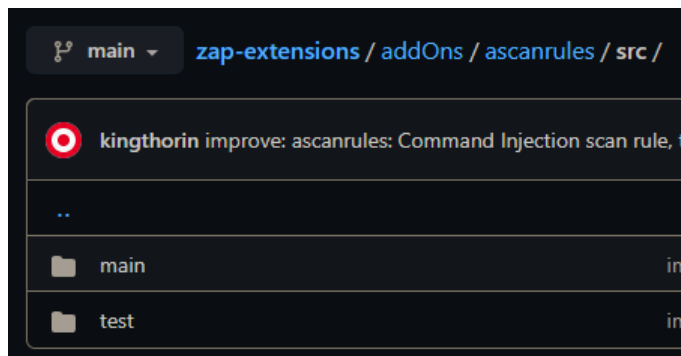
2. Modifiability

- **Tactic** : Coupling, Cohesion, Split Module
- **Evidences** :



Ref : [zapproxy/zap-extensions: OWASP ZAP Add-ons \(github.com\)](https://github.com/zaproxy/zap-extensions)

มีการแยก Repository ระหว่าง Module หลักกับตัว AddOns อย่างชัดเจน และมีการแยก Folder แต่ละ Module ของ AddOns (Support Tactic Split Module)



ในแต่ละ AddOns จะมี Main ของแต่ละ AddOn สามารถบอกได้ว่าการแยกการทำงานของแต่ละ AddOns ชัดเจน และสามารถพัฒนา AddOn ได้โดยไม่ส่งผลกระทบ

Name	Version	Description
Active scanner rules	35.0.0	The release quality Active Scanner rules
Active scanner rules (alpha)	28.0.0	The alpha quality Active Scanner rules
Active scanner rules (beta)	30.0.0	The beta quality Active Scanner rules
Advanced SQLInjection Scanner	13.0.0	An advanced active injection bundle for SQLi (derived by SQLMap)

```
public AddOnCollection(File[] dirs) {  
    if (dirs != null) {  
        for (File dir : dirs) {  
            try {  
                this.addDirectory(dir);  
            } catch (Exception e) {  
                logger.error(e.getMessage(), e);  
            }  
        }  
    }  
}
```

Ref : [zapproxy/AddOnCollection.java at main · zapproxy/zapproxy \(github.com\)](#)

ในโค้ดของ Zapproxy จะไม่มีการใส่ Source Code ของ AddOns เข้าไป แต่จะเป็นการ Import Directory ของ AddOns เข้ามา และจะเรียกใช้จาก AddOns นั้นๆ ทำให้ไม่ส่งผลกระทบต่อ Module หลัก สามารถบอกได้ว่าการแยกการทำงานอย่างชัดเจน ส่วน AddOns จะทำงานแต่ส่วนของ AddOn ส่วน Zapproxy จะทำงานแต่ส่วนของ Zapproxy ต่าง Modules ทำหน้าที่ของตัวเอง และไม่เกี่ยวโยงกับ Modules อื่นๆ

(Support Tactics : Cohesion, Coupling)

- Reason การทำแบบนี้สามารถทำให้ระบบมีความยืดหยุ่นสามารถพัฒนาได้หลากหลาย ผู้ที่ไม่ได้พัฒนาตัวระบบหลัก สามารถพัฒนา AddOns ได้โดยไม่ส่งผลกระทบต่อระบบได้

3. Testability

- Tactic : Specialized Interfaces, Abstract data source, Sandbox
- Evidences :

```
public Builder setHeaders(List<HttpHeaderField> headers) {
    if (headers == null) {
        throw new IllegalArgumentException("Parameter headers must not be null.");
    }
    if (!headers.isEmpty()) {
        List<HttpHeaderField> validHeaders = new ArrayList<>();
        for (HttpHeaderField headerField : headers) {
            if (headerField == null) {
                throw new IllegalArgumentException("Element of headers must not be null.");
            }
            if (headerField.getName() != null
                && !headerField.getName().trim().isEmpty()
                && headerField.getValue() != null) {
                validHeaders.add(headerField);
            }
        }
        this.headers = Collections.unmodifiableList(validHeaders);
    } else {
        this.headers = Collections.emptyList();
    }
    return this;
}
```

```
private void addDirectory(File dir) throws Exception {
    if (dir == null) {
        logger.error("Null directory supplied");
        return;
    }
    if (!dir.exists()) {
        logger.warn(
            "Skipping enumeration of add-ons, the directory does not exist: "
            + dir.getAbsolutePath());
        return;
    }
    if (!dir.isDirectory()) {
        logger.warn("Not a directory: " + dir.getAbsolutePath());
        return;
    }
}
```

Ref : [zapproxy/AddOnCollection.java at main · zapproxy/zapproxy \(github.com\)](#),
[zapproxy/SpiderResourceFound.java at c8afeb8a79283430f89063b12f7bc350d50e15d1 · zapproxy/zapproxy \(github.com\)](#)

มีการใช้ setter() เพื่อที่จะ Validated ค่าไม่ให้เป็น Null ก่อนจะทำงาน Function เพื่อป้องกันไม่ให้ระบบผิดพลาด และมีการ throwExeption เพื่อที่จะให้สามารถป้องกัน Error และสามารถหาจุดที่ผิดพลาดได้

(Support Tactic : Specialized Interfaces)

```
private static final String USER_NAME = "username";  
private static int CONTEXT_ID = 23;
```

```
@Test  
public void shouldEncodeAndDecodeProperly() {  
    // Given  
    User user = spy(new User(CONTEXT_ID, USER_NAME));  
    user.setAuthenticationCredentials(mockedCredentials);  
    doReturn(mockedContext).when(user).getContext();  
    // When  
    String encoded = User.encode(user);  
    User result = User.decode(CONTEXT_ID, encoded, mockedExtension);  
    // Then  
    assertEquals(user.getName(), result.getName());  
    assertEquals(user.isEnabled(), result.isEnabled());  
    assertEquals(user.getId(), result.getId());  
    assertEquals(user.getContextId(), result.getContextId());  
}  
  
@Test  
public void shouldGenerateUniqueIds() {  
    // Given  
    User u1 = new User(CONTEXT_ID, USER_NAME);  
    User u2 = new User(CONTEXT_ID, USER_NAME);  
    User u3 = new User(CONTEXT_ID, USER_NAME);  
    User u4 = new User(CONTEXT_ID, USER_NAME);  
    // When/Then  
    assertThat(u1.getId(), not(anyOf(is(u2.getId()), is(u3.getId()), is(u4.getId()))));  
    assertThat(u2.getId(), not(anyOf(is(u3.getId()), is(u4.getId()))));  
    assertThat(u3.getId(), not(is(u4.getId())));  
    User u5 = new User(CONTEXT_ID, USER_NAME, u4.getId() + 5);  
    User u6 = new User(CONTEXT_ID, USER_NAME);  
    assertThat(u6.getId(), greaterThan(u5.getId()));  
}
```

สร้าง Dummy user เพื่อนำ Object เข้ามาทดสอบในระบบ จำลองข้อมูล และใช้ Assertion เพื่อทดสอบระบบหาข้อผิดพลาด การทำแบบนี้จะใกล้เคียงกับสภาพแวดล้อม OnProduction มากที่สุด

(Support Tactics : Abstract data source, Sandbox)

Ref : [zaproxy/UserUnitTest.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zaproxy/zaproxy \(github.com\)](https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/UserUnitTest.java)

- Reason ผู้พัฒนาสามารถหาข้อผิดพลาดของระบบตัวเองได้ง่าย หรือสามารถแก้ไขได้ง่ายถูกจุดก่อนจะ Deploy ระบบ

4. Availability

- Tactic : Exception handling
- Evidences :

```

107  /**
108   * Main method
109   *
110   * @param args the arguments passed to the command line version
111   * @throws Exception if something wrong happens
112   */
113   public static void main(String[] args) throws Exception {
114       setCustomErrStream();
115
116       CommandLine cmdLine = null;
117       try {
118           cmdLine = new CommandLine(args != null ? Arrays.copyOf(args, args.length) : null);
119
120       } catch (final Exception e) {
121           // Cant use the CommandLine help here as the
122           // i18n messages wont have been loaded
123           System.out.println("Failed due to invalid parameters: " + Arrays.toString(args));
124           System.out.println(e.getMessage());
125           System.out.println("Use '-h' for more details.");
126           System.exit(1);
127       }
128   }

```

```

if (publisher == null) {
    throw new InvalidParameterException("Publisher must not be null");
}
if (eventTypes == null || eventTypes.length == 0) {
    throw new InvalidParameterException("At least one event type must be specified");
}

```

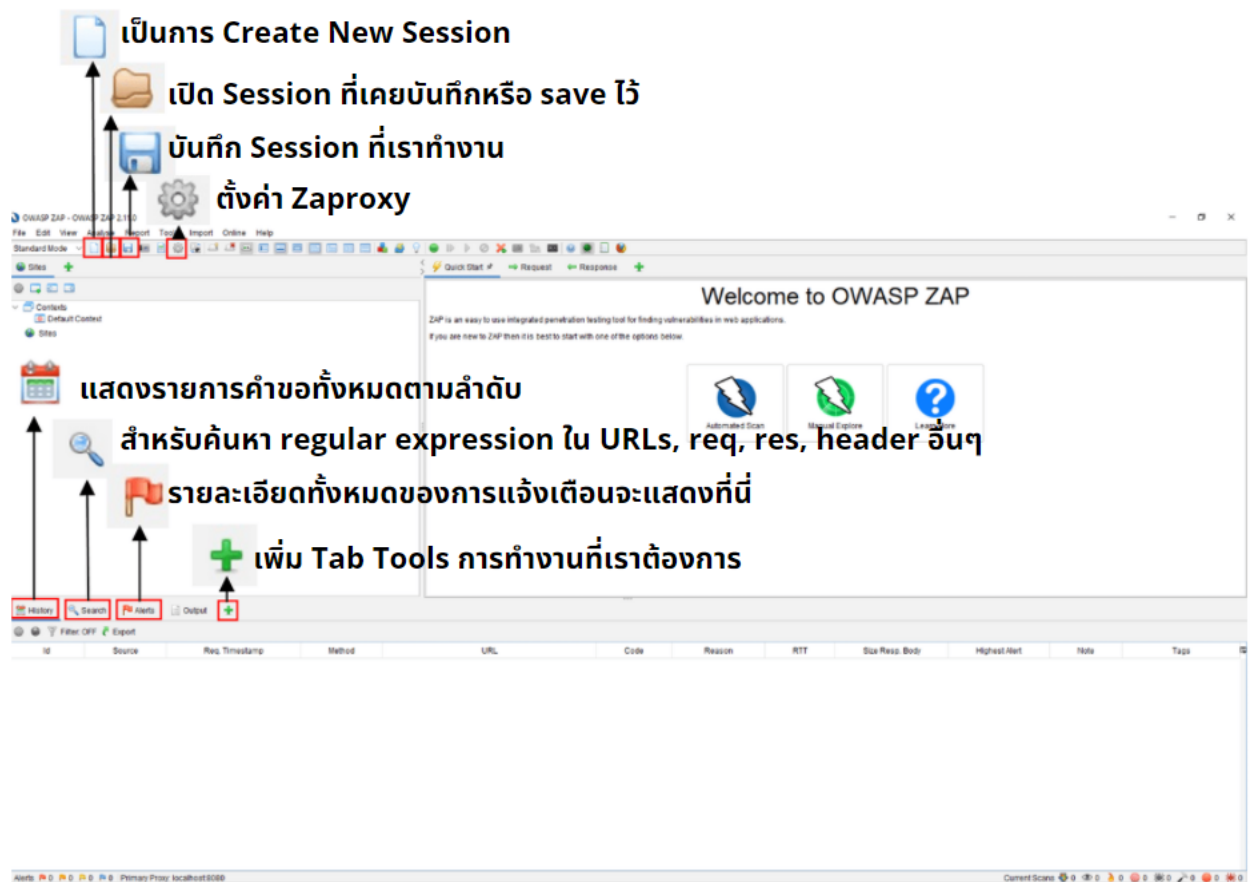
Ref : [zaproxy/ZAP.java at main · zaproxy/zaproxy \(github.com\)](https://github.com/zaproxy/zaproxy/blob/main/zaproxy/ZAP.java),
[zaproxy/SimpleEventBus.java at main · zaproxy/zaproxy \(github.com\)](https://github.com/zaproxy/zaproxy/blob/main/zaproxy/SimpleEventBus.java)

ผู้พัฒนาได้คาดการณ์ข้อผิดพลาดที่อาจจะเกิดขึ้น จึงใส่ Function Try Catch และ Throw Exception เพื่อรองรับ และจัดการกับข้อผิดพลาดที่อาจจะเกิดขึ้น ทำให้ระบบสามารถรับรู้ข้อผิดพลาด และไม่ทำให้เกิดปัญหา ระบบจะสามารถทำงานต่อไปอย่างปกติ แต่จะเป็นการแจ้งเตือนข้อผิดพลาดแทน
 (Support Tactic : Exception handling)

- Reason เป็นการคาดการณ์ปัญหาที่อาจจะเกิดขึ้น ผู้พัฒนาจึงสร้างตัวรองรับข้อผิดพลาดขึ้นมาเพื่อที่จะสามารถระบุข้อผิดพลาด, แก้ไข และทำให้ระบบยังสามารถทำงานต่อไป

5. Recognizability

- Tactic : Specialized Interfaces, Abstract data source, Sandbox
- Evidences :



Ref : <https://www.zaproxy.org/docs/desktop/ui/>

- Reason เป็น แถบเครื่องมือที่ใช้สัญลักษณ์ที่สื่อความหมายในตัวเอง รวมไปถึงในโปรแกรมอื่นๆ ก็ยังใช้สัญลักษณ์ภาพที่เหมือนกัน ทำให้ผู้ใช้งานไม่จำเป็นต้องไปเรียนรู้ว่า Tools นี้คืออะไร

6. Security

- Tactic : Encrypt data, Identity actors
- Evidences :

```
/**
 * Encodes the User in a String. Fields that contain strings are Base64 encoded.
 *
 * @param user the user
 * @return the string
 */
public static String encode(User user) {
    StringBuilder out = new StringBuilder();
    out.append(user.id).append(FIELD_SEPARATOR);
    out.append(user.isEnabled()).append(FIELD_SEPARATOR);
    out.append(Base64.encodeBase64String(user.name.getBytes())).append(FIELD_SEPARATOR);
    out.append(user.getContext().getAuthenticationMethod().getType().getUniqueIdentifier())
        .append(FIELD_SEPARATOR);
    out.append(user.authenticationCredentials.encode(FIELD_SEPARATOR));
    if (log.isDebugEnabled()) log.debug("Encoded user: " + out.toString());
    return out.toString();
}
```

สร้าง User สำหรับการใช้งาน Web Session ระบบจะทำการเข้ารหัส User ของเรา
(Support tactic : Encrypt data)

```
public static HttpMessage prepareMessage(
    SessionManagementMethod sessionManagementMethod, User user) {
    HttpMessage msg = new HttpMessage();
    // Make sure the message will be sent with a good WebSession that can record the changes
    if (user.getAuthenticatedSession() == null)
        user.setAuthenticatedSession(sessionManagementMethod.createEmptyWebSession());
    msg.setRequestingUser(user);

    return msg;
}
```

```
public void setAuthenticatedSession(WebSession session) {
    this.authenticatedSession = session;
}
```

Class AuthenticatedHelper จะทำการ Set Authentication ให้ User

```

136         @Override
137         public WebSession authenticate(
138             SessionManagementMethod sessionManagementMethod,
139             AuthenticationCredentials credentials,
140             User user)
141             throws UnsupportedAuthenticationCredentialsException {
142
143             WebSession session = user.getAuthenticatedSession();
144             if (session == null) session = sessionManagementMethod.createEmptyWebSession();
145
146             // type check
147             if (!(credentials instanceof UsernamePasswordAuthenticationCredentials)) {
148                 user.getAuthenticationState()
149                     .setLastAuthFailure(
150                         "Credentials not UsernamePasswordAuthenticationCredentials");
151                 throw new UnsupportedAuthenticationCredentialsException(
152                     "Form based authentication method only supports "
153                     + UsernamePasswordAuthenticationCredentials.class.getSimpleName());
154             }
155             UsernamePasswordAuthenticationCredentials userCredentials =
156                 (UsernamePasswordAuthenticationCredentials) credentials;
157
158             AuthScope stateAuthScope = null;
159             NTCredentials stateCredentials = null;
160             try {
161                 stateAuthScope =
162                     new AuthScope(
163                         this.hostname,
164                         this.port,
165                         (this.realm == null || this.realm.isEmpty())
166                             ? AuthScope.ANY_REALM
167                             : this.realm);
168                 stateCredentials =
169                     new NTCredentials(
170                         userCredentials.getUsername(),
171                         userCredentials.getPassword(),
172                         InetAddress.getLocalHost().getCanonicalHostName(),
173                         this.realm);
174                 session.getHttpState().setCredentials(stateAuthScope, stateCredentials);
175             } catch (UnknownHostException e1) {
176                 user.getAuthenticationState().setLastAuthFailure(e1.getMessage());
177                 log.error(e1.getMessage(), e1);
178             }
179             user.getAuthenticationState().setLastAuthFailure("");
180             return session;
181         }
182

```

Web session จะ set authenticated สำหรับ User ที่ใช้งาน
(Support tactic : Identity actors)

```

/**
 * Modifies a message so its Request Header/Body matches the web session corresponding to this
 * user.
 *
 * @param message the message
 */
public void processMessageToMatchUser(HttpMessage message) {
    // If the user is not yet authenticated, authenticate now
    // Make sure there are no simultaneous authentications for the same user
    synchronized (this) {
        if (this.requiresAuthentication()) {
            this.authenticate();
            if (this.requiresAuthentication()) {
                log.info("Authentication failed for user: " + name);
                return;
            }
        }
    }
    processMessageToMatchAuthenticatedSession(message);
}

/**
 * Modifies a message so its Request Header/Body matches the web session corresponding to this
 * user.
 *
 * @param message the message
 */
public void processMessageToMatchAuthenticatedSession(HttpMessage message) {
    getContext()
        .getSessionManagementMethod()
        .processMessageToMatchSession(message, authenticatedSession);
}

```

เมื่อจะมีการเรียกใช้ Web Session User จะทำการเตรียม Http Message เพื่อบอกระบบให้รู้ว่าเป็น User ที่ได้รับสิทธิ์การใช้งาน

(Support tactic : Identity actors)

- Reason ผู้พัฒนาจะได้จำกัดสิทธิ์ User ที่ใช้งานในการใช้ Tool ของตัวเองกับ Web Session ที่ใช้งาน

Ref:

<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/zaproxy/zap/authentication>

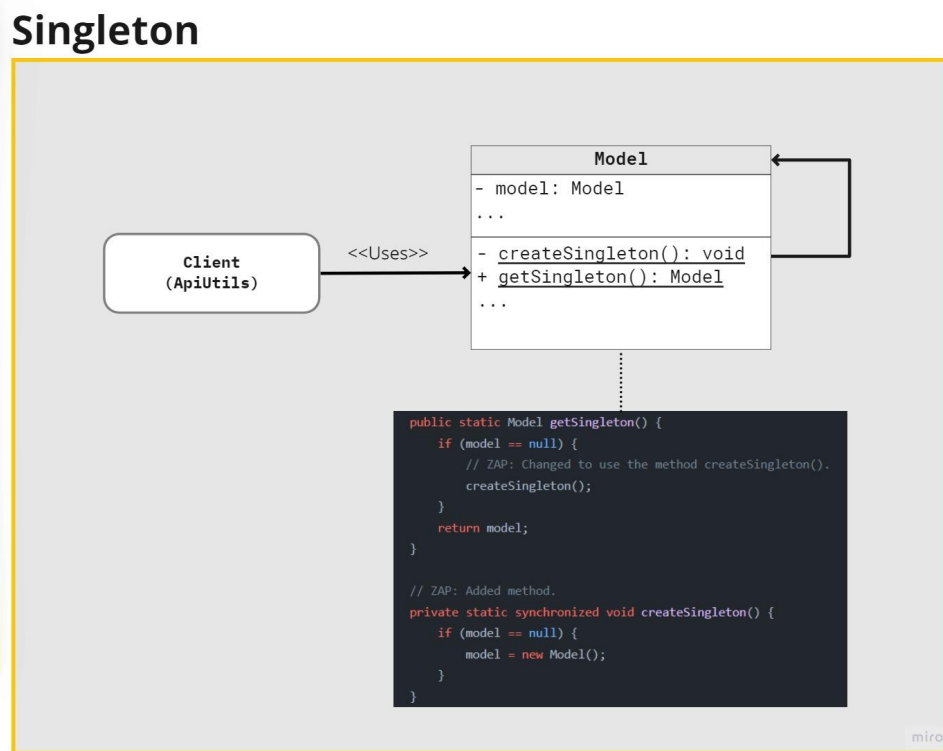
Design Patterns

Identification of at least 4 software design patterns used in the project

1. Singleton Pattern

- Creational Patterns
- Path: zaproxy/zap/src/main/java/org/parosproxy/paros/model/Model.java
- Line: 213 - 238
<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/parosproxy/paros/model/Model.java#L213-L238>
- Usage: zaproxy/zap/src/main/java/org/zaproxy/zap/utils/ApiUtils.java
<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/zaproxy/zap/utils/ApiUtils.java>

UML:



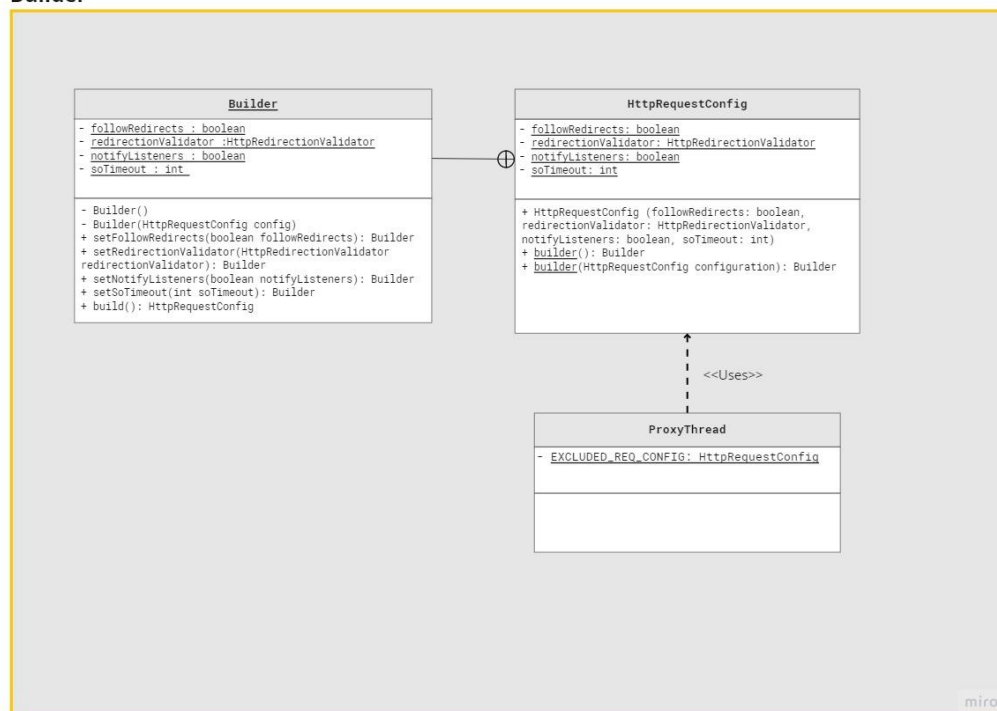
Singleton จะพบในคลาส Model Model จะมีการสร้าง object model และกำหนดให้เป็น private เพื่อไม่ให้คนอื่นเข้ามาแก้ไขได้ โดยคลาสนี้จะมี method ที่ชื่อว่า `getSingleton()` ซึ่งมี return type เป็น Model method นี้เป็น method ที่ใช้สร้าง object ของคลาส Model และทำหน้าที่จำกัด object ของ Model ให้มีเพียงตัวเดียว และมีการกำหนดการเข้าถึงให้เป็น static method เพื่อให้คลาสอื่น ๆ สามารถเข้าถึง object Model ได้

2. Builder Pattern

- Creational Patterns
- Path: zaproxy/zap/src/main/java/org/zaproxy/zap/network/HttpRequestConfig.java
- Line: 129 - 226
- <https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/zaproxy/zap/network/HttpRequestConfig.java#L128-L226>
- Usage: zaproxy/zap/src/main/java/org/parosproxy/paros/core/proxy/ProxyThread.java
<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/parosproxy/paros/core/proxy/ProxyThread.java>

UML:

Builder



Builder พบในคลาส HttpRequestConfig โดยคลาส HttpRequestConfig เป็นคลาสที่ใช้สำหรับ configure วิธีการทำงานของ Http Request โดยในคลาสนี้จะมี Inner class ที่ชื่อว่า Builder โดย Builder จะมี Attribute ต่าง ๆ ที่เกี่ยวข้องกับ HttpRequestConfig และมี Constructor ของ Builder จะใช้สร้าง Object ของ Builder ขึ้นมา

ใน Builder ประกอบด้วย method ต่าง ๆ ที่ใช้สำหรับการ set configure ต่าง ๆ ให้กับ HttpRequest และสุดท้ายจะมี method ที่ชื่อว่า build() โดยจะมีการ return ค่าเป็น HttpRequestConfig

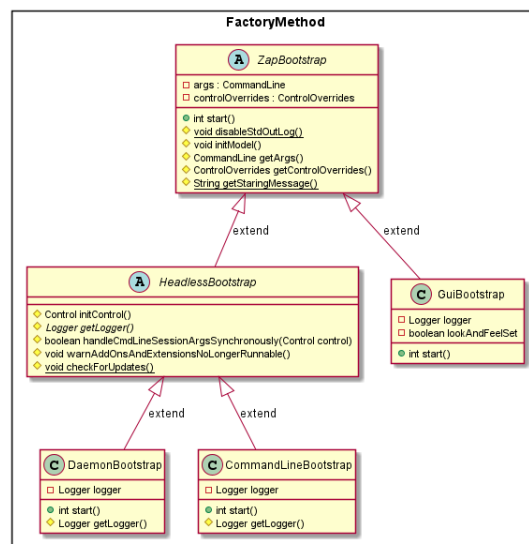
เนื่องจาก HttpRequestConfig มีขั้นตอนการสร้าง และการ config ค่าต่าง ๆ เยอะ จึงทำให้ต้องใช้ Builder แยกออกมา เพื่อให้สร้าง object ได้สะดวก

3. Factory Method

- Creational Patterns
- Path : zap/src/main/java/org/zaproxy/zap/ZapBootstrap.java

- Link :
<https://github.com/zaproxy/zaproxy/blob/ba55831c0ee1667586275307cbc14be1528442ca/zap/src/main/java/org/zaproxy/zap/ZapBootstrap.java>
- Sub Class:
 - zap/src/main/java/org/zaproxy/zap/HeadlessBootstrap.java
 - zap/src/main/java/org/zaproxy/zap/CommandLineBootstrap.java
 - zap/src/main/java/org/zaproxy/zap/DaemonBootstrap.java
 - zap/src/main/java/org/zaproxy/zap/GuiBootstrap.java
- Usage:
 - zap/src/main/java/org/zaproxy/zap/ZAP.java
 - Line : 159-172
 - [zaproxy/ZAP.java at ba55831c0ee1667586275307cbc14be1528442ca · zaproxy/zaproxy \(github.com\)](https://github.com/zaproxy/zaproxy/blob/ba55831c0ee1667586275307cbc14be1528442ca/zap/src/main/java/org/zaproxy/zap/ZAP.java)

UML



Zap proxy สามารถทำงานได้หลายรูปแบบ ทั้งการทำงานแบบ Desktop (Stand-alone,Gui) , Daemon (Background Process,No Gui) และ Cmdline(Inline process) เป็นต้น การใช้ Factory Method จึงเหมาะสมเพื่อให้การทำงานต่างๆ ไม่ Coupling กัน และ การใช้ Object ที่สร้าง Subclass จะเป็นตัวกำหนดการทำงานในแต่ละรูปแบบ ในการพัฒนาต่อให้ทำงานรูปแบบใหม่อย่างเช่น Zaas(Zap as a Service) ที่ผู้พัฒนากำลังจะทำอยู่ก็เพียงสร้าง Subclass ใหม่ และ ทำการกำหนดการทำงานต่างๆ หากในอนาคตมีการทำงานหลากหลายรูปแบบมากขึ้นก็จะทำให้เกิดความซับซ้อนของ Coding เนื่องจากแต่ละรูปแบบจะต้องสร้าง Class ขึ้นมา

4. Adapter Pattern

1. Adapter : MessageLocationProducerFocusListenerAdapter.java

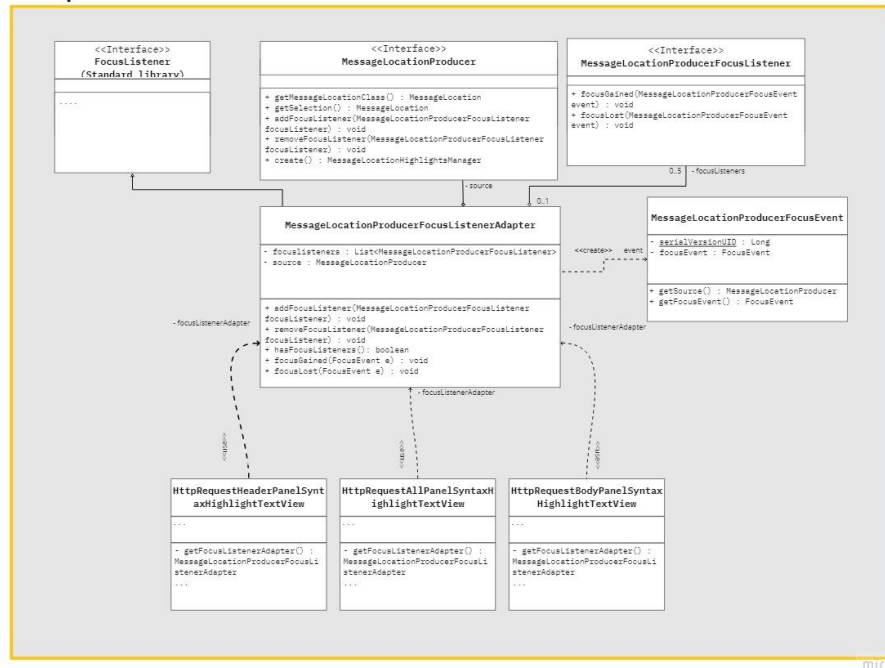
[zaproxy/MessageLocationProducerFocusListenerAdapter.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zaproxy/zaproxy \(github.com\)](https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zaproxy/MessageLocationProducerFocusListenerAdapter.java)

2. Usage

- ตัวอย่างที่นำ Data มาจาก Adapter [zaproxy/HttpRequestHeaderPanelSyntaxHighlightTextView.java at b9271e71e5749bb72d03c37a74863574c3329168 · zaproxy/zaproxy \(github.com\)](https://b9271e71e5749bb72d03c37a74863574c3329168-zaproxy/zaproxy.github.com)
- มีที่เรียกใช้ทั้งหมด 3 ตัว
[Search · MessageLocationProducerFocusListenerAdapter \(github.com\)](#)

UML:

Adapter



ทำงานผ่าน Adapter หรือ Interface ที่เป็นต้นแบบ(FocusListener) ซึ่ง Adapter ก็จะทำงานกับ Class ที่เราต้องการทำงานด้วย 2 Class คือ MessageLocationProducer(source) และ MessageLocationProducerFocusListener (focusListeners : Array[5]) โดยที่จะแปลงไปเป็น focusListener หลายๆตัว ที่จะทำการเก็บ Event ของ FocusEvent ซึ่งแต่ละ Event ก็จะถูกกำหนดผ่าน Format ของ source ในรูปแบบ MessageLocationProducerFocusEvent เป็น EventObject ทำให้ **Zap Proxy** สามารถปรับเปลี่ยนการทำงานได้ผ่าน Adapter เป็น library ตัวอื่นหรือการพัฒนาได้สะดวกขึ้น

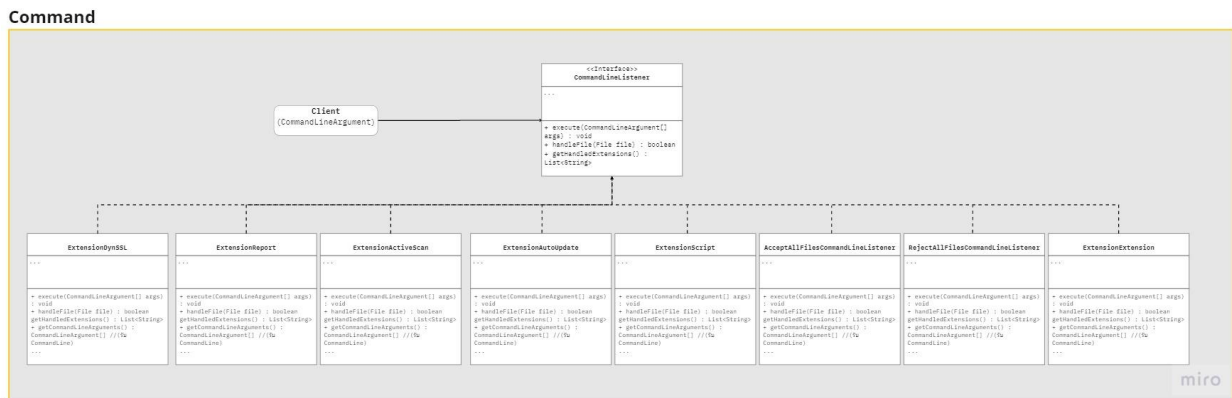
5. Command Pattern

CommandLineListener : [zaproxy/CommandLineListener.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zaproxy/zaproxy \(github.com\)](#)

- [zapproxy/ExtensionReport.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)
- [zapproxy/ExtensionActiveScan.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)
- [zapproxy/ExtensionAutoUpdate.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)
- [zapproxy/ExtensionDynSSL.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)
- [zapproxy/ExtensionExtension.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)

- f. [zapproxy/ExtensionScript.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)
- g. [zapproxy/CommandLineUnitTest.java at 8ac4b10d3d9d816957985694c2b7cf127d1364d3 · zapproxy/zapproxy \(github.com\)](#)
- (2 Class)

UML:



ZAP Proxy เป็น Source Code ขนาดใหญ่ มี Extension หลายตัวดังนั้นคำสั่งต่างๆก็มากขึ้นตามไปด้วย ทำให้การใช้ Command จะช่วยให้รองรับและเรียกใช้คำสั่งต่างๆได้ง่ายขึ้นโดยผ่าน Command object ซึ่งจะทำการส่งต่อไปให้ Receiver ตามที่ Command object ได้รับมา ถึงแม้ว่าจะทำให้การ Coding มีความซับซ้อน