**Arrakis XSS Option Guide**

1. 설정 옵션 설명

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| <config>  최상위 사용 | 필터링에 사용할 필터 정보들의 집합. 여러개의 <filter>를 정의할 수 있다.  **Element**   |  |  | | --- | --- | | <filters> | 필터의 고유 이름. 다른곳에서 filter 정보를 참조할때 쓰인다. | | <appliers> | 필터링을 위한 조건 정의 정보들의 집합. 하위에 여러개의 applier를 정의할 수 있다. | |

|  |  |  |  |
| --- | --- | --- | --- |
| <filters> | 필터의 고유 이름. 다른곳에서 filter 정보를 참조할때 쓰인다.  **Element**   |  |  | | --- | --- | | <filter> | 필터링에 사용할 필터의 정보.  기본적으로 DefaultFilter를 제공한다 (정규식을 이용한 단순 replace) | |
| <appliers> | 필터링을 위한 조건 정의 정보들의 집합. 하위에 여러개의 applier를 정의할 수 있다.  **Element**   |  |  | | --- | --- | | <applier> | 필터링 대상 조건을 판단하는 설정 정의 | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <filter> | 필터링에 사용할 필터의 정보.  기본적으로 DefaultFilter를 제공한다 (정규식을 이용한 단순 replace)  **Element**   |  |  | | --- | --- | | <name> | 필터의 고유 이름. 다른곳에서 filter 정보를 참조할때 쓰인다. | | <classPath> | 해당 필터를 구현한 class 의 위치. package 형태로 기술한다. | | <configPath> | 해당 필터의 설정 정보 파일 | | <config> | 해당 필터가 처리할 변환 rule 을 정의한다.  위 configPath 설정을 통해 별도 파일로 분리하고 내용을 기술할 수 있다. |   **Attribute**   |  |  | | --- | --- | | default | 아래 inspect 설정에서 사용할 filter를 지정하지 않았을때 기본적으로 적용되는 filter목록을 선언한다. | |

|  |  |  |  |
| --- | --- | --- | --- |
| <config>  flter 아래 사용 | 필터링을 수행하는 설정 정의  하위 Element는 Filter에서 추가적으로 옵션을 받아 처리하려면 정의해서 사용한다. (사용자 정의 태그)  메뉴얼에서는 기본적으로 제공하는 DefaultFilter에 대한 옵션을 설명하였다.  **Element**   |  |  | | --- | --- | | <rules> | 필터링 Rule을 정의한다 | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| <applier> | 필터링을 수행하는 설정 정의  **Element**   |  |  | | --- | --- | | <conditions> | 필터링 대상 판단의 조건들을 정의한다. | | <inspect> | 필터링 대상일때 실제 처리를 위한 설정 |   **Attribute**   |  |  | | --- | --- | | break | 해당 applier를 수행한 후 이후 applier를 수행하지 않으려면 해당 옵션을 넣는다. 보통 특정 예외조건의 선언시 상부에서 처리할때 사용한다.  ex)  break=”true” | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| <conditions> | 컨디션 매칭시 실제 실행할 조건 집합. 여러개의 condition을 가질수 있으며 순서대로 매칭한다. 먼저 매칭되는건이 있으면 아래는 무시한다. (각 contidion은 OR조건)  **Attribute**   |  |  | | --- | --- | | ref-type | 해당 match 조건에서 비교할 대상의 값/값들을 선택한다.  기본적으로 아래와 같은 옵션을 제공한다.  ref-type=”SimpleKeyMatch”  > 필터링 대상의 키값으로 매칭을 시도한다. (파라메터 키)  ref-type=”SimpleValueMatch”  > 필터링 대상의 실제 값으로 매칭을 시도한다. (파라메터 값)  해당 type은 커스터마이징 가능하며 웹 XSS 처리를 위해 아래 기능을 추가로 제공한다.  ref-type=”RequestMap”  > request의 파라메터 키와 실제 값의 정보를 가져와 비교할때 사용.  ref-type=”RequestMetaMap”  > request의 메타정보를 가져와 비교할때 사용.  현재는 URI를 가져와 비교 가능하다. |   **Element**   |  |  | | --- | --- | | <condition> | 필터링 매칭 조건을 정의한다. | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <inspact> | 컨디션 매칭시 실제 실행할 조건. 속성에 기본 필터를 선언하여 해당 필터로 동작하게 한다.  혹은 위 조건일때 필터를 동작하게 무시할 수 있다.  **Attribute**   |  |  | | --- | --- | | exclude | 컨디션 조건 해당시 별도 처리하지 않게 예외처리를 한다.  ex)  exclude=”false” (기본/기술하지 않음)  exclude=”true” (제외 적용) | | filter | 특정 필터를 기본적으로 사용하도록 선언한다. 여러개 정의를 할때는 콤마(,) 를 이용하여 정의하며 순서대로 필터가 처리된다.  ex)  filter=”DefaultFilter,XSSFilter” | | name | 해당 inspect의 고유명. 로그기록 등에 이용한다. 필수는 아님. |   **Element**   |  |  | | --- | --- | | rule | 필터링 후 변환 조건을 기술한다. | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <condition> | 컨디션 매칭시 실제 실행할 조건 정의.  각 condtion은 순서대로 비교하며 먼저 매칭된 condition이 있으면 아래 condition은 무시한다 (OR조건)  **Attribute**   |  |  | | --- | --- | | ref-type | 해당 match 조건에서 비교할 대상의 값/값들을 선택한다.  기본적으로 아래와 같은 옵션을 제공한다.  ref-type=”SimpleKeyMatch”  > 필터링 대상의 키값으로 매칭을 시도한다. (파라메터 키)  ref-type=”SimpleValueMatch”  > 필터링 대상의 실제 값으로 매칭을 시도한다. (파라메터 값)  해당 type은 커스터마이징 가능하며 웹 XSS 처리를 위해 아래 기능을 추가로 제공한다.  ref-type=”RequestMap”  > request의 파라메터 키와 실제 값의 정보를 가져와 비교할때 사용.  ref-type=”RequestMetaMap”  > request의 메타정보를 가져와 비교할때 사용.  현재는 URI를 가져와 비교 가능하다. | | operator | 하위 match 조건들에 대해 or / and 조건을 기술한다.  기본적으로 “and” 조건이다. (match의 모든 조건이 충족되어야함)  ex) operator=”or” | | name | 해당 condition의 고유명. 로그기록 등에 이용한다. 필수는 아님. |   **Element**   |  |  | | --- | --- | | <match> | 매칭 조건을 기술한다. | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <rule> | 특정 키에 대해 별도 처리를 하려고 할때 사용한다. 정의하지 않으면 기본적으로 모든 값에 대해 적용한다.  **Attribute**   |  |  | | --- | --- | | key-name | 별도 filter 처리를 할 키 이름을 입력한다.  ex)  exclude=”false” (기본/기술하지 않음)  exclude=”true” (제외 적용) | | match-type | 설정한 키를 어떤 룰에 맞게 매칭할지 판단한다.  속성은 아래와 같다  match-type=”exactly”  > 키값과 비교값이 정확히 일치할때 처리한다.  match-type=”startWith”  > 키값이 비교값의 시작부터 부분일치할때 처리한다.  match-type=”endWith”  > 키값이 비교값의 뒷부분이 부분일치할때 처리한다.  match-type=”contains”  > 키값이 비교값의 일부분일때 처리한다.  match-type=”reges”  > 정규식을 이용하여 키값이 비교값에 매칭될떄 처리한다. | | filter | 설정한 키값에 대해 특정 필터를 별도로 적용하려고 할때 사용함.  이때 <inspect> 태그에서 설정한 default filter는 무시된다. | | exclude | 설정한 키값에 대한 처리는 무시하려고 할때 사용함  ex) exclude=”true” | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <match> | 특정 키에 대해 별도 처리를 하려고 할때 사용한다. 정의하지 않으면 기본적으로 모든 값에 대해 적용한다.  **Attribute**   |  |  | | --- | --- | | ref-type | 해당 match 조건에서 비교할 대상의 값/값들을 선택한다.  기본적으로 아래와 같은 옵션을 제공한다.  ref-type=”SimpleKeyMatch”  > 필터링 대상의 키값으로 매칭을 시도한다. (파라메터 키)  ref-type=”SimpleValueMatch”  > 필터링 대상의 실제 값으로 매칭을 시도한다. (파라메터 값)  해당 type은 커스터마이징 가능하며 웹 XSS 처리를 위해 아래 기능을 추가로 제공한다.  ref-type=”RequestMap”  > request의 파라메터 키와 실제 값의 정보를 가져와 비교할때 사용.  ref-type=”RequestMetaMap”  > request의 메타정보를 가져와 비교할때 사용.  현재는 URI를 가져와 비교 가능하다. | | key-name | 별도 filter 처리를 할 키 이름을 입력한다.  ex)  exclude=”false” (기본/기술하지 않음)  exclude=”true” (제외 적용) | | match-type | 설정한 키를 어떤 룰에 맞게 매칭할지 판단한다.  속성은 아래와 같다  match-type=”exactly”  > 키값과 비교값이 정확히 일치할때 처리한다.  match-type=”startWith”  > 키값이 비교값의 시작부터 부분일치할때 처리한다.  match-type=”endWith”  > 키값이 비교값의 뒷부분이 부분일치할때 처리한다.  match-type=”contains”  > 키값이 비교값의 일부분일때 처리한다.  match-type=”reges”  > 정규식을 이용하여 키값이 비교값에 매칭될떄 처리한다. | | operator | 연산자를 정의한다. 해당 match 태그에서는 “not” 만 지원한다.  operator=”not”  > 해당 match 속성에 해당하는것이 아닌것을 적용한다. | | name | 해당 match의 고유명. 로그기록 등에 이용한다. 필수는 아님. | |

1. 기본 제공 filter 설명

현재 기본적으로 제공된 filter의 종류는 아래와 같다.

|  |  |
| --- | --- |
| net.ngom.arrakis.filter.DefaultFilter | 설정파일의 Rule설정에 따라 단순 정규식에 따른 Replace를 수행하는 Filter. 내부적으로 java의 replaceAll 함수를 이용한다. |
| kr.co.ensof.xss.filter.OWASPFilter | OWASP라는 비영리 오픈소스 웹 애플리케이션 보안 프로젝트에서 배포하는 보안취약점 개선 라이브러리를 사용한다.  <https://www.owasp.org/index.php/OWASP_Java_Encoder_Project>  참고. |
| kr.co.ensof.xss.filter.SQLInjectionFilter | SQL Injection 공격시 자주 사용하는 특수문자 패턴을 제거하는 Filter이다. |