RIPSTECH

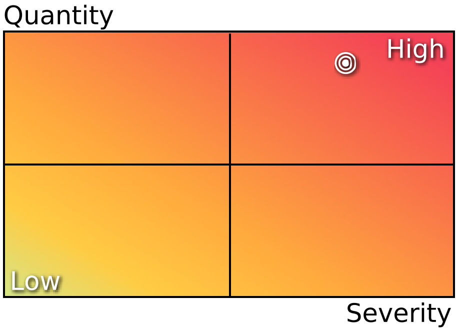
SECURITY ANALYSIS REPORT

Popup Builder v3.60 - RIPS Check Report
Date: 2020-01-21

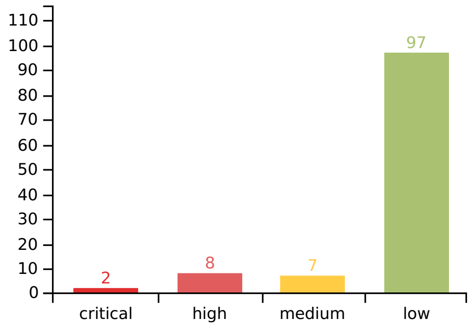
1. Executive Summary

Project Name:	Popup Builder v3.60 - RIPS Check Report	Analyzed Files:	78
Analysis Start Date:	2020-01-21, 09:36	Analyzed LOC:	23,566
Analysis End Date:	2020-01-21, 09:40	Analyzed Issue Types:	209
Analysis Time:	3m 50s	Detected Issues:	114
Engine Version(s):	php-preparser 3.3.1 php-engine 3.3.4 php-patchgen 3.3.0	Max Issues per Type:	100 / 500

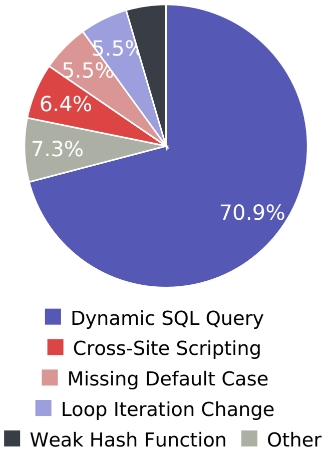
Risk Matrix



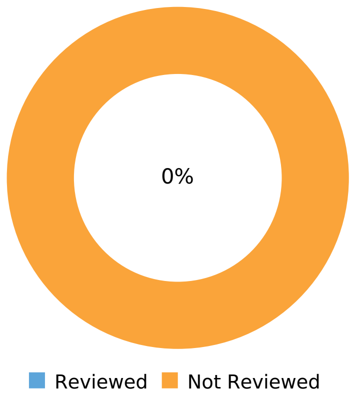
Vulnerabilities by Risk



Top Vulnerability Types



Review Status



2. Issue Breakdown

The detected security issues in this project are categorized as follows.

Severity	Vulnerability Type	CWE	OWASP	SANS 25	PCI DSS	ASVS	Issues
		[?]	Top 10 [?]	[?]	[?]	[?]	
Critical	Remote File Inclusion	98	A5	Rank 13	6.5.8	5.3.9	1
Critical	SQL Injection	89	A1	Rank 1	6.5.1	5.3.4	1
High	Local File Inclusion	97	A5	Rank 13	6.5.8	5.3.9	3
High	Path Traversal	22	A5	Rank 13	6.5.8	8.1.2	1
High	Object Instantiation	470	A5	Rank 10	6.5.8	5.2.4	3
High	Phar Deserialization	915	A8	Rank 16	6.5.8		1
Medium	Cross-Site Scripting	79	A7	Rank 4	6.5.7	5.3.3	7
Low	Information Leakage	209	A6	Not Ranked	6.5.5	7.4.1	1
Low	Dynamic SQL Query	89		Not Ranked			78
Low	Missing Error Handling	390		Not Ranked			1
Low	Missing Default Case	478		Not Ranked			6
Low	Weak Hash Function	328		Not Ranked			5
Low	Loop Iteration Change	834		Not Ranked			6

3. Issue Details

In the following, all security issues detected in the analyzed project are presented in detail. The issues are grouped by vulnerability type and by the detected markup context. A *markup context* represents the position of user-supplied data (*source*) used in a sensitive operation (*sink*). Depending on the markup context, an attacker can alter the operation and different security mechanisms must be applied in order to patch the security issue thoroughly.

3.1. Remote File Inclusion

ASVS:	4.0.1: 5.3.9
OWASP Top 10:	2017: A5
CWE:	98
SANS 25:	Rank 13
PCI DSS:	6.5.8
Severity:	Critical

This vulnerability is categorized as a remote file inclusion (RFI) vulnerability because no path name is prefixed to the injection point. An attacker can inject a protocol handler, such as http:// or ftp://, to include remote files from an attacker server and to execute system commands. Further, an attacker is able to include arbitrary files from the file system as program code. This can lead to the disclosure of sensitive files or to the execution of code that has been placed by the attacker on the file system, for example by injecting a payload into a log file.

A remote file inclusion vulnerability occurs when user input is used at the beginning of a path supplied to an inclusion function such as include(), require(), or require_once(). In this context, an attacker could include remote files by supplying a URL. To prevent abuse, it is advised to prefix the input with a constant part of the desired folder. This prevents inclusion of remote files. However, arbitrary inclusion of local files still needs to be prevented by hindering path traversal with the PHP built-in function basename(). If the names of includable files is given, it is best to use a whitelist approach.

Issue #2303 - popup-builder/com/classes/RegisterPostType.php: 249

Path:	popup-builder/com/classes/RegisterPostType.php
Line:	249
Sink:	require_once
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

The user-supplied data is concatenated into path markup in line 249 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::createPopupObj().

The user-supplied data is then used unsanitized in the sensitive operation require_once() in line 249 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::createPopupObj(). Please refer to the context and description for further information.

popup-builder/com/classes/RegisterPostType.php

```
class RegisterPostType
{
⋮
public function postTypeSupportForPopupTypes($supports)
{
$popupType = $this->getPopupTypeName();
⋮
}
⋮
private function createPopupObjFromPopupType()
{
⋮
$this->setPopupType($popupType);
⋮
$this->createPopupObj();
}
⋮
public function createPopupObj()
{
⋮
$popupType = $this->getPopupType();

⋮
$popupClassName = $this->getPopupClassNameFromPopupType($popupType);
⋮
require_once($typePath.$popupClassName.'.php');
⋮
}
⋮
}
```

Path Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['sg_popup_options']['sgpb-type']
```

3.2. SQL Injection

ASVS:	4.0.1: 5.3.4
OWASP Top 10:	2017: A1
CWE:	89
SANS 25:	Rank 1
PCI DSS:	6.5.1
Severity:	Critical

A SQL injection vulnerability occurs when unsanitized user input is embedded into a SQL query. An attacker can modify the SQL syntax and alter the query's target or result. This can lead to the retrieval of sensitive information from the database or to an attack against the underlying web server by using SQL file operations. An attacker can also elevate privileges if the SQL query is used for authentication.

SQL injection vulnerabilities occur when not sufficiently sanitized user input is used in the construction of a SQL query sent to the database. Malicious input could alter the semantics of the query and lead to not intended behavior. To prevent abuse, it is necessary to make sure that the data inserted in the SQL query only gets treated as data and not as SQL commands. This is best achieved by making use of prepared statements. In case the dynamic part of the SQL query does not present data in the executed query (e.g., table or column name) and thus, cannot be bound with prepared statements, validating against a whitelist is advised.

3.2.1. SQL Injection (unquoted)

ASVS:	4.0.1: 5.3.4
OWASP Top 10:	
CWE:	89
SANS 25:	Rank 1
PCI DSS:	6.5.1
Severity:	Critical

No quotes are used around the detected injection point in the SQL query. Thus, all applied operations to escape the data are insufficient because no quotes have to be broken in order to inject SQL syntax.

In this case of SQL injection, the injection occurs in a context with no quotes. To prevent abuse, it is necessary to make sure that the data inserted in the SQL query only gets treated as data and not as SQL commands. This is best achieved by making use of prepared statements. If the input is intended to be used as an integer, an explicit typecast can be applied. In case the dynamic part of the SQL query does not present data in the executed query (e.g., table or column name) and thus, cannot be bound with prepared statements, validating against a whitelist is advised.

Issue #2351 - popup-builder/com/classes/Actions.php: 1202

Path:	popup-builder/com/classes/Actions.php
Line:	1202
Sink:	get_results
Source:	_GET
Taint:	HTTP

Code Summary

The GET parameter 'orderby' is received in line 1189 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::getSubscribersCsvFile().

The user-supplied data is concatenated into sql markup in line 1189 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::getSubscribersCsvFile().

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 1202 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::getSubscribersCsvFile(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7     class Actions
8     {
⋮
1183  public function getSubscribersCsvFile()
1184  {
⋮
1186  $query = AdminHelper::subscribersRelatedQuery();
⋮
1189  $query .= ' ORDER BY '.esc_sql($_GET['orderby']).' '.esc_sql($_GET['order']);
⋮
1202  $subscribers = $wpdb->get_results($query, ARRAY_A);
⋮
1226  }
⋮
1258  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT firstName, lastName, email, cDate, SGPB_POSTS_TABLE_NAME.post_title AS subscriptionTitle FROM
SGPB_SUBSCRIBERS_TABLE_NAME LEFT JOIN SGPB_POSTS_TABLE_NAME ON
SGPB_POSTS_TABLE_NAME.ID=SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' %' ORDER BY
$_GET['orderby']$_GET['order']
```

Patch

SQL Injection No Quotes

popup-builder/com/classes/Actions.php

```
if (isset($_GET['order']) && !empty($_GET['order'])) {
$identifier = "`" . str_replace("`", "``", esc_sql($_GET['orderby'])) . "`";
$query .= ' ORDER BY ' . $identifier . ' ' . esc_sql($_GET['order']);
}
```

3.3. Local File Inclusion

ASVS:	4.0.1: 5.3.9
OWASP Top 10:	2017: A5
CWE:	97
SANS 25:	Rank 13
PCI DSS:	6.5.8
Severity:	High

This vulnerability is categorized as a local file inclusion (LFI) vulnerability because a path name is prefixed to the injection point. An attacker can use path traversal character sequences (../) to access and include arbitrary files from the file system as code. This can lead to the disclosure of sensitive files or to the execution of program code that was placed by the attacker on the file system, for example by injecting a payload into a log file.

A Local File Inclusion vulnerability occurs when not sufficiently sanitized user input is used to construct the path of a file included by a PHP built-in function such as include(), require(), or require_once(). This could allow an attacker to make use of directory traversal and include files containing sensitive information or malicious code. To prevent this, the input needs to be sanitized to prevent directory traversal. This can be achieved with PHP built-in function basename(). It is also recommended to use a whitelist approach in order to verify the value before usage.

3.3.1. Local File Inclusion (limited)

ASVS:	4.0.1: 5.3.9
OWASP Top 10:	2017: A5
CWE:	626
SANS 25:	Rank 13
PCI DSS:	6.5.8
Severity:	High

This vulnerability is categorized as a local file inclusion (LFI) vulnerability because a path name is prefixed to the injection point. An attacker can use path traversal character sequences (../) to access and include arbitrary files from the file system as code. This can lead to the disclosure of sensitive files or to the execution of program code that was placed by the attacker on the file system, for example by injecting a payload into a log file. The suffix in the file path can be truncated by an attacker with a null byte injection in PHP <= 5.3.4.

This vulnerability was reported because not sufficiently sanitized user input is used to construct a file path for a dynamic inclusion. The dynamic part of the file path is suffixed with a constant string which limits the possibilities of an attacker. However, in older PHP versions (<= 5.3.*) an attacker could make use of null byte injections to truncate the path at the desired location. To prevent this, the input needs to be sanitized to prevent directory traversal. This can be achieved with PHP built-in function basename(). It is also recommended to use a whitelist approach in order to verify the value before usage.

Issue #2287 - popup-builder/com/classes/popups/SGPopup.php: 292

Path:	popup-builder/com/classes/popups/SGPopup.php
Line:	292
Sink:	require_once
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 284 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::find().

The user-supplied data is concatenated into path markup in line 292 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::find().

The user-supplied data is then used unsanitized in the sensitive operation require_once() in line 292 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::find(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10    abstract class SGPopup
11    {
⋮
223   public static function find($popup, $args = array())
224   {
⋮
271   $savedData = array();
⋮
274   $savedData = PopupData::getPopupDataById($popupId, $saveMode);
⋮
276   $savedData = apply_filters('sgpbPopupSavedData', $savedData);
⋮
282   $type = 'html';
⋮
284   $type = $savedData['sgpb-type'];
⋮
287   $popupClassName = self::getPopupClassNameFormType($type);
⋮
292   require_once($typePath.$popupClassName.'.php');
⋮
318   }
⋮
1689  }
```

Path Context

The following snippet(s) do not represent actual code but the tainted context.

```
SG_POPUP_CLASSES_POPUPS_PATHArrayPopup.php
```

Patch

Whitelisting Dynamic Includes

popup-builder/com/classes/popups/SGPopup.php

```
//TODO: Fill in the array below with paths to files for which you want to allow inclusion.
if (!in_array($typePath . $popupClassName . '.php', ['path/file1.php', 'path/file2.php'], true)) {
throw new \Exception('Trying to include a not-whitelisted file.');
}
require_once $typePath . $popupClassName . '.php';
```

Issue #2302 - popup-builder/com/classes/RegisterPostType.php: 159

Path:	popup-builder/com/classes/RegisterPostType.php
Line:	159
Sink:	require_once
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

The user-supplied data is concatenated into path markup in line 159 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

The user-supplied data is then used unsanitized in the sensitive operation require_once() in line 159 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes(). Please refer to the context and description for further information.

popup-builder/com/classes/RegisterPostType.php

```
5    class RegisterPostType
6    {
 :
149  public function postTypeSupportForPopupTypes($supports)
150  {
151  $popupType = $this->getPopupTypeName();
 :
153  $popupClassName = $this->getPopupClassNameFromPopupType($popupType);
 :
159  require_once($typePath.$popupClassName.'.php');
 :
168  }
 :
487  }
```

Path Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['sg_popup_options']['sgpb-type'] Popup.php
```

Patch

Whitelisting Dynamic Includes

popup-builder/com/classes/RegisterPostType.php

```
159  //TODO: Fill in the array below with paths to files for which you want to allow inclusion.
160  if (!in_array($typePath . $popupClassName . '.php', ['path/file1.php', 'path/file2.php'], true)) {
161  throw new \Exception('Trying to include a not-whitelisted file.');
162  }
163  require_once $typePath . $popupClassName . '.php';
```

Issue #2380 - popup-builder/com/classes/popups/SGPopup.php: 1433

Path:	popup-builder/com/classes/popups/SGPopup.php
Line:	1433
Sink:	require_once
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

popup-builder/com/classes/RegisterPostType.php

```
5    class RegisterPostType
6    {
:
149  public function postTypeSupportForPopupTypes($supports)
150  {
151  $popupType = $this->getPopupTypeName();
:
168  }
:
487  }
```

The user-supplied data is concatenated into path markup in line 1433 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::createPopupTypeObjById().

The user-supplied data is then used unsanitized in the sensitive operation require_once() in line 1433 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::createPopupTypeObjById(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10    abstract class SGPopup
11    {
:
1414  public static function createPopupTypeObjById($popupId)
1415  {
1416  global $SGPB_POPUP_TYPES;
1417  $typePath = '';
1418  $popupOptionsData = SGPopup::getPopupOptionsById($popupId);
:
1422  $popupType = $popupOptionsData['sgpb-type'];
1423  $popupName = ucfirst(strtolower($popupType));
1424  $popupClassName = $popupName.'Popup';
:
1427  $typePath = $SGPB_POPUP_TYPES['typePath'][$popupType];
:
1433  require_once($typePath.$popupClassName.'.php');
⋮
}
⋮
}
```

Path Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['sg_popup_options']['sgpb-type'] Popup.php
```

Patch

Whitelisting Dynamic Includes

popup-builder/com/classes/popups/SGPopup.php

```
//TODO: Fill in the array below with paths to files for which you want to allow inclusion.
if (!in_array($typePath . $popupClassName . '.php', ['path/file1.php', 'path/file2.php'], true)) {
throw new \Exception('Trying to include a not-whitelisted file.');
}
require_once $typePath . $popupClassName . '.php';
```

3.4. Path Traversal

ASVS:	4.0.1: 8.1.2
OWASP Top 10:	2017: A5
CWE:	22
SANS 25:	Rank 13
PCI DSS:	6.5.8
Severity:	High

A path traversal vulnerability occurs when user input is used unsanitized in a file path for reading. An attacker can access arbitrary files on the file system by using path traversal character sequences (../). This can lead to the disclosure of code, configuration files, and login credentials.

The reported file path traversal vulnerability occurs because not sufficiently sanitized user input is used in the path provided to a file reading function. This can result in disclosure of sensitive information. To prevent abuse, it is necessary to sanitize the input to prohibit injection of path traversing characters like “../”. This can be achieved by using the PHP built-in function basename(). Whenever possible, it is advised to use a whitelist approach to filter out inputs that are not allowed.

Issue #2361 - popup-builder/com/classes/Ajax.php: 313

Path:	popup-builder/com/classes/Ajax.php
Line:	313
Sink:	file
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'importListURL' is received in line 309 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation file() in line 313 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
305  public function saveImportedSubscribers()
306  {
⋮
309  $fileURL = sanitize_text_field($_POST['importListURL']);
⋮
313  $csvFileArray = array_map('str_getcsv', file($fileURL));
⋮
338  }
⋮
683  }
```

Path Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['importListURL']
```

Patch

Whitelisting File Path

popup-builder/com/classes/Ajax.php

```
313  $fileName = basename($fileURL);
314  // TODO: Adjust whitelist. If whitelisting is not possible, use a restrictive preg_match()
315  if (!in_array($fileName, ['POSSIBLE_FILE1.ext', 'POSSIBLE_FILE2.ext'], true)) {
316  new \Exception('Invalid file path supplied.');
317  }
318  // TODO: Set a fixed directory if possible
319  $fixedDir = '/folder1/folder2/';
320  $filePath = $fixedDir . $fileName;
321  $csvFileArray = array_map('str_getcsv', file($filePath));
```

3.5. Object Instantiation

ASVS:	4.0.1: 5.2.4
OWASP Top 10:	2017: A5
CWE:	470
SANS 25:	Rank 10
PCI DSS:	6.5.8
Severity:	High

An object instantiation vulnerability occurs when unsanitized user input is used as class name in an object constructor. An attacker can instantiate an object of an arbitrary class which can lead to unexpected control flow of the application. For example, it can be abused to bypass authentication or access control checks.

An object instantiation vulnerability occurs when unsanitized user input is used as class name in an object constructor. An attacker can instantiate an object of an arbitrary class which can lead to unexpected control flow of the application. To prevent abuse in case such a construct is necessary, the class name should be checked against a whitelist.

Issue #2288 - popup-builder/com/classes/popups/SGPopup.php: 295

Path:	popup-builder/com/classes/popups/SGPopup.php
Line:	295
Sink:	
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 284 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::find().

The user-supplied data is concatenated into classname markup in line 293 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::find().

The user-supplied data is then used unsanitized in line 295 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::find(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10    abstract class SGPopup
11    {
⋮
223   public static function find($popup, $args = array())
224   {
⋮
271   $savedData = array();
⋮
274   $savedData = PopupData::getPopupDataById($popupId, $saveMode);
⋮
276   $savedData = apply_filters('sgpbPopupSavedData', $savedData);
⋮
282   $type = 'html';
⋮
284   $type = $savedData['sgpb-type'];
⋮
287   $popupClassName = self::getPopupClassNameFormType($type);
⋮
293   $popupClassName = __NAMESPACE__.'\\'.$popupClassName;
⋮
295   $obj = new $popupClassName();
⋮
318   }
⋮
1689  }
```

Classname Context

The following snippet(s) do not represent actual code but the tainted context.

```
sgpb\ $_POST['sg_popup_options' . $saveMode]['sgpb-type'] Popup
```

Patch

Whitelist Possible Classes before Instantiation

popup-builder/com/classes/popups/SGPopup.php

```
// TODO: Fill in the array below with classes which you want to allow instantiation
if (!in_array($popupClassName, ['PossibleClass1', 'PossibleClass2'], true)) {
throw new \Exception('Trying to instantiate a not-whitelisted class.');
}
$obj = new $popupClassName();
```

Issue #2304 - popup-builder/com/classes/RegisterPostType.php: 251

Path:	popup-builder/com/classes/RegisterPostType.php
Line:	251
Sink:	
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

The user-supplied data is concatenated into classname markup in line 250 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::createPopupObj().

The user-supplied data is then used unsanitized in line 251 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::createPopupObj(). Please refer to the context and description for further information.

popup-builder/com/classes/RegisterPostType.php

```
5     class RegisterPostType
6     {
⋮
149   public function postTypeSupportForPopupTypes($supports)
150   {
151   $popupType = $this->getPopupTypeName();
⋮
168   }
⋮
182   private function createPopupObjFromPopupType()
183   {
⋮
191   $this->setPopupType($popupType);
⋮
194   $this->createPopupObj();

195   }
⋮
239   public function createPopupObj()
240   {
⋮
242   $popupType = $this->getPopupType();
⋮
244   $popupClassName = $this->getPopupClassNameFromPopupType($popupType);
⋮
250   $popupClassName = __NAMESPACE__.'\\'.$popupClassName;
251   $popupTypeObj = new $popupClassName();
⋮
268   }
⋮
487   }
```

Classname Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['sg_popup_options']['sgpb-type']
```

Patch

Whitelist Possible Classes before Instantiation

popup-builder/com/classes/RegisterPostType.php

```
// TODO: Fill in the array below with classes which you want to allow instantiation
if (!in_array($popupClassName, ['PossibleClass1', 'PossibleClass2'], true)) {
throw new \Exception('Trying to instantiate a not-whitelisted class.');
}
$popupTypeObj = new $popupClassName();
```

Issue #2381 - popup-builder/com/classes/popups/SGPopup.php: 1436

Path:	popup-builder/com/classes/popups/SGPopup.php
Line:	1436
Sink:	
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

popup-builder/com/classes/RegisterPostType.php

```
5    class RegisterPostType
6    {
⋮
149  public function postTypeSupportForPopupTypes($supports)
150  {
151  $popupType = $this->getPopupTypeName();
⋮
168  }
⋮

487  }
```

The user-supplied data is concatenated into classname markup in line 1435 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::createPopupTypeObjById().

The user-supplied data is then used unsanitized in line 1436 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::createPopupTypeObjById(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10    abstract class SGPopup
11    {
```

```
⋮
public static function createPopupTypeObjById($popupId)
{
global $SGPB_POPUP_TYPES;
$typePath = '';
$popupOptionsData = SGPopup::getPopupOptionsById($popupId);
⋮
$popupType = $popupOptionsData['sgpb-type'];
$popupName = ucfirst(strtolower($popupType));
$popupClassName = $popupName.'Popup';
⋮
$typePath = $SGPB_POPUP_TYPES['typePath'][$popupType];
⋮
require_once($typePath.$popupClassName.'.php');
⋮
$popupClassName = __NAMESPACE__.'\\'.$popupClassName;
$popupTypeObj = new $popupClassName();
⋮
}
⋮
}
```

Classname Context

The following snippet(s) do not represent actual code but the tainted context.

```
sgpb\ $_POST['sg_popup_options']['sgpb-type'] Popup
```

Patch

Whitelist Possible Classes before Instantiation

popup-builder/com/classes/popups/SGPopup.php

```
// TODO: Fill in the array below with classes which you want to allow instantiation
if (!in_array($popupClassName, ['PossibleClass1', 'PossibleClass2'], true)) {
throw new \Exception('Trying to instantiate a not-whitelisted class.');
}
$popupTypeObj = new $popupClassName();
```

3.6. Phar Deserialization

ASVS:	
OWASP Top 10:	2017: A8
CWE:	915
SANS 25:	Rank 16
PCI DSS:	6.5.8
Severity:	High

All file operations in PHP allow to use URL-style wrappers when accessing file paths. An attacker can misuse these wrappers when user input fully controls the file path. In this case, the phar:// wrapper can be injected which allows to load a given file as PHP Archive file. The meta data of Phar files is stored serialized and it is unserialized when the file is accessed via the phar:// wrapper. Thus, if an attacker is able to upload a Phar file to the web server, the attacker can use this vulnerability to unserialize its meta data and to inject arbitrary PHP objects to the application. This can lead to further vulnerabilities such as remote code execution. Note that a Phar file can also be hidden within a JPG image such that a regular picture upload is sufficient.

To prevent Phar Deserialization vulnerabilities you should not allow users to control the path of a file operation. If this is not possible you should check the path and do not perform the file operation if the path contains phar://.

Issue #2379 - popup-builder/com/classes/popups/SGPopup.php: 1430

Path: popup-builder/com/classes/popups/SGPopup.php
Line: 1430
Sink: file_exists
Source: _POST
Taint: HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

popup-builder/com/classes/RegisterPostType.php

```
5     class RegisterPostType
6     {
⋮
149   public function postTypeSupportForPopupTypes($supports)
150   {
151   $popupType = $this->getPopupTypeName();
⋮
168   }
⋮
487   }
```

The user-supplied data is concatenated into file path markup in line 1430 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::createPopupTypeObjById().

The user-supplied data is then used unsanitized in the sensitive operation file_exists() in line 1430 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::createPopupTypeObjById(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10     abstract class SGPopup
11     {
⋮
1414   public static function createPopupTypeObjById($popupId)
1415   {
1416   global $SGPB_POPUP_TYPES;
1417   $typePath = '';
1418   $popupOptionsData = SGPopup::getPopupOptionsById($popupId);
⋮
1422   $popupType = $popupOptionsData['sgpb-type'];
1423   $popupName = ucfirst(strtolower($popupType));
1424   $popupClassName = $popupName.'Popup';
⋮
1427   $typePath = $SGPB_POPUP_TYPES['typePath'][$popupType];
⋮
1430   if (!file_exists($typePath.$popupClassName.'.php')) {
⋮
1440   }
⋮
1689   }
```

File Path Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['sg_popup_options']['sgpb-type'] Popup.php
```

Patch

Prohibiting Usage of phar:// Stream Wrapper

popup-builder/com/classes/popups/SGPopup.php

```
if (stripos($typePath . $popupClassName . '.php', 'phar://') !== false) {
throw new \Exception('Potential Phar PHP Object Injection detected.');
}
if (!file_exists($typePath . $popupClassName . '.php')) {
wp_die(__('Popup class does not exist', SG_POPUP_TEXT_DOMAIN));
}
```

3.7. Cross-Site Scripting

ASVS:	4.0.1: 5.3.3
OWASP Top 10:	2017: A7
CWE:	79
SANS 25:	Rank 4
PCI DSS:	6.5.7
Severity:	Medium

A reflected cross-site scripting (XSS) vulnerability occurs when unsanitized user input is embedded into the HTML response page of the web application. It allows an attacker to inject arbitrary HTML or JavaScript code into the response page of a tampered request. Usually, this attack is performed by crafting a malicious link that is sent to a victim. When opened, the attacker's JavaScript payload within the link is reflected by the application and executed in the victim's browser in the context of the web application's domain. This enables the attacker to perform phishing attacks, to steal cookies associated with the domain, or to cause the victim’s browser to execute arbitrary actions on the victim’s behalf and without the victim’s knowledge.

To prevent cross-site scripting vulnerabilities, special characters that are interpreted by the browser to execute not intended actions need to be escaped or filtered out of user input before usage. Which characters are considered harmful and need to be sanitized depends on the context the injection happens in (e.g., attribute context, URL context, JavaScript context, ...).

3.7.1. Cross-Site Scripting (normal tag)

ASVS:	4.0.1: 5.3.3
OWASP Top 10:	2017: A7
CWE:	80
SANS 25:	Rank 4
PCI DSS:	6.5.7
Severity:	Medium

The detected injection occurs between two HTML elements. An attacker can inject a new HTML element, such as the <script> element, to invoke the JavaScript interpreter and execute arbitrary JavaScript code.

The detected cross-site scripting vulnerability occurs in the main HTML context of the page. An attacker could inject HTML tags to execute arbitrary JavaScript or alter the appearance of the page. To prevent this from happening, special characters that can introduce such tags need to be escaped or filtered out from user-controlled input prior to usage. For this, the PHP built-in function `htmlentities()` can be used. This function transforms special characters, such as `<`, to their HTML encoded representation (`&lt;` in the case of `<`). Browsers would still render the characters as they are, but no longer interpret them as being part of the HTML structure.

Issue #2289 - popup-builder/public/views/customEditor.php: 46

Path:	popup-builder/public/views/customEditor.php
Line:	46
Sink:	echo
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_scripts[js][sgpb-]' is received in line 46 of the file popup-builder/public/views/customEditor.php.

The user-supplied data is concatenated into html markup in line 46 of the file popup-builder/public/views/customEditor.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 46 of the file popup-builder/public/views/customEditor.php. Please refer to the context and description for further information.

popup-builder/public/views/customEditor.php

```
10   $savedData = get_post_meta($popupId , 'sg_popup_scripts', true);
:
46   echo $savedData['js']['sgpb-'.$key];
```

HTML Context

The following snippet(s) do not represent actual code but the tainted context.

```
<textarea class="wp-editor-area editor-content" data-attr-event="" placeholder=" #... type your code" mode=""
name="sgpb-"> $_POST['sg_popup_scripts']['js']['sgpb-' . $key] "'>
```

Patch

HTML htmlentities() Encoding

popup-builder/public/views/customEditor.php

```
46   echo htmlentities($savedData['js']['sgpb-' . $key]);
```

Issue #2290 - popup-builder/public/views/customEditor.php: 77

Path:	popup-builder/public/views/customEditor.php
Line:	77
Sink:	echo
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_scripts[css]' is received in line 77 of the file popup-builder/public/views/customEditor.php.

The user-supplied data is concatenated into html markup in line 77 of the file popup-builder/public/views/customEditor.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 77 of the file popup-builder/public/views/customEditor.php. Please refer to the context and description for further information.

popup-builder/public/views/customEditor.php

```
10  $savedData = get_post_meta($popupId , 'sg_popup_scripts', true);
:
77  echo $savedData['css'];
```

HTML Context

The following snippet(s) do not represent actual code but the tainted context.

```
<textarea class="wp-editor-area editor-content editor-content-css" placeholder=" #... type your code" mode=""
name="sgpb-css-editor" > $_POST['sg_popup_scripts']['css'] "'>
```

Patch

HTML htmlentities() Encoding

popup-builder/public/views/customEditor.php

```
77  echo htmlentities($savedData['css']);
```

Issue #2305 - popup-builder/com/classes/Updates.php: 253

Path:	popup-builder/com/classes/Updates.php
Line:	253
Sink:	echo
Source:	_GET
Taint:	HTTP

Code Summary

The GET parameter 'message' is received in line 250 of the file popup-builder/com/classes/Updates.php in the method sgpbUpdates::sgpbAdminNotices().

The user-supplied data is concatenated into html markup in line 253 of the file popup-builder/com/classes/Updates.php in the method sgpbUpdates::sgpbAdminNotices().

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 253 of the file popup-builder/com/classes/Updates.php in the method sgpbUpdates::sgpbAdminNotices(). Please refer to the context and description for further information.

popup-builder/com/classes/Updates.php

```
9    class Updates
10   {
:
245  public function sgpbAdminNotices()
246  {
:
250  $message = urldecode($_GET['message']);
:
253  <h3><?php echo $message; ?></h3>
⋮
261  }
262  }
```

HTML Context

The following snippet(s) do not represent actual code but the tainted context.

```
<h3> $_GET['message'] "'>
```

Patch

HTML htmlentities() Encoding

popup-builder/com/classes/Updates.php

```
253  <h3><?php echo htmlentities($message); ?></h3>
```

Issue #2349 - popup-builder/com/classes/Actions.php: 920

Path:	popup-builder/com/classes/Actions.php
Line:	920
Sink:	echo
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'sg_popup_options[sgpb-type]' is received in line 151 of the file popup-builder/com/classes/RegisterPostType.php in the method sgpbRegisterPostType::postTypeSupportForPopupTypes().

popup-builder/com/classes/RegisterPostType.php

```
5    class RegisterPostType
6    {
⋮
149  public function postTypeSupportForPopupTypes($supports)
150  {
151  $popupType = $this->getPopupTypeName();
⋮
168  }
⋮
487  }
```

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 920 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::popupsTableColumnsValues(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7    class Actions
8    {
⋮
889  public function popupsTableColumnsValues($column, $postId)
890  {
⋮
915  global $SGPB_POPUP_TYPES;
916  $type = $popup->getType();
⋮
918   $type = $SGPB_POPUP_TYPES['typeLabels'][$type];
⋮
920   echo $type;
⋮
933   }
⋮
1258  }
```

HTML Context

The following snippet(s) do not represent actual code but the tainted context.

Patch

HTML htmlentities() Encoding

popup-builder/com/classes/Actions.php

```
920  echo htmlentities($type);
```

3.7.2. Cross-Site Scripting (style tag)

ASVS:	4.0.1: 5.3.3
OWASP Top 10:	2017: A7
CWE:	79
SANS 25:	Rank 4
PCI DSS:	6.5.7
Severity:	Medium

The detected injection occurs within a <style> element. An attacker can inject arbitrary CSS code that modifies the page’s appearance or even leads to the execution of JavaScript code in older browsers.

The detected cross-site scripting vulnerability occurs within a CSS context of the output. Escaping special HTML characters that may allow an attacker to transition out of the CSS context (e.g, by injecting `</style>`), is not sufficient here. The reason being that an attacker could still inject arbitrary CSS code into the context. Besides to being able to arbitrarily alter the appearance of the page, an attacker could achieve execution of JavaScript code from within the CSS context in older browsers. If the intention is to dynamically alter certain property values within the CSS code, it is recommended to validate the format of every input on its own.

Issue #2310 - popup-builder/com/classes/ScriptsLoader.php: 148

Path:	popup-builder/com/classes/ScriptsLoader.php
Line:	148
Sink:	echo
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter '_wpb_shortcodes_custom_css' is received in line 139 of the file popup-builder/com/classes/ScriptsLoader.php in the method sgpbScriptsLoader::loadToAdmin().

The user-supplied data is concatenated into css markup in line 142 of the file popup-builder/com/classes/ScriptsLoader.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 148 of the file popup-builder/com/classes/ScriptsLoader.php. Please refer to the context and description for further information.

popup-builder/com/classes/ScriptsLoader.php

```
6    class ScriptsLoader
7    {
⋮
126  public function loadToAdmin()
127  {
⋮
130  foreach ($popups as $popup) {
131  $popupId = $popup->getId();
⋮
133  $events = array();
⋮
135  $events = json_encode($events);
⋮
137  $popupOptions = $this->getEncodedOptionsFromPopup($popup);
⋮
139  $popupContent = apply_filters('sgpbPopupContentLoadToPage', $popup->getPopupTypeContent(), $popupId);
⋮
141  add_action('admin_footer', function() use ($popupId, $events, $popupOptions, $popupContent) {
142  $footerPopupContent = '<div style="position:absolute;top: -999999999999999999999px;">
143  <div class="sg-popup-builder-content" id="sg-popup-content-wrapper-'.$popupId.'" data-id="'.esc_attr($popupId).'" data-events="'.esc_attr($events).'" data-options="'.esc_attr($popupOptions).'">
144  <div class="sgpb-popup-builder-content-'.esc_attr($popupId).' sgpb-popup-builder-content-html">'.$popupContent.'</div>
145  </div>
146  </div>';
⋮
148  echo $footerPopupContent;
149  });
150  }
⋮
154  }
⋮
339  }
```

CSS Context

The following snippet(s) do not represent actual code but the tainted context.

```
<style> $_POST['_wpb_shortcodes_custom_css'] </style> </div> </div> </div> </div>
```

3.7.3. Cross-Site Scripting (double-quoted attribute)

ASVS:	4.0.1: 5.3.3
OWASP Top 10:	2017: A7
CWE:	79
SANS 25:	Rank 4
PCI DSS:	6.5.7
Severity:	Medium

The detected injection occurs within a double-quoted HTML attribute. An attacker can break out of this attribute by injecting a double quote (\"). This allows to terminate the current attribute and to append another attribute to the HTML element. For example, an eventhandler attribute can be appended that allows to execute arbitrary JavaScript code.

The detected cross-site scripting vulnerability occurs within the context of an attribute surrounded by double quotes. To prevent abuse, it is necessary to prohibit the potentially malicious input from breaking out of this context, and inject an event handler or start a new HTML tag. The PHP built-in function htmlentities() can be used for this matter. While escaping of single quotes is not necessary at this point, it is still recommended to do so by adding the ENT_QUOTES flag to the call to htmlentities().

Issue #2291 - popup-builder/public/views/mainActionButtons.php: 12

Path:	popup-builder/public/views/mainActionButtons.php
Line:	12
Sink:	echo
Source:	_SERVER
Taint:	HTTP

Code Summary

The URI that contains partially unencoded special characters in certain browsers is received in line 12 of the file popup-builder/public/views/mainActionButtons.php.

The user-supplied data is concatenated into html markup in line 12 of the file popup-builder/public/views/mainActionButtons.php.

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 12 of the file popup-builder/public/views/mainActionButtons.php. Please refer to the context and description for further information.

popup-builder/public/views/mainActionButtons.php

```
12   <a href="<?php echo AdminHelper::getSettingsURL(array('sgpbImport' => 1)); ?>" class="page-title-action">
```

HTML Context

The following snippet(s) do not represent actual code but the tainted context.

```
<a href=" $_SERVER['REQUEST_URI'] ?.*=https://target/wp-admin//edit.php?
post_type=SG_POPUP_POST_TYPE&page=SG_POPUP_SETTINGS_PAGE"'>
```

Patch

HTML htmlentities() Encoding In Attribute Context

popup-builder/public/views/mainActionButtons.php

```
12   <a href="<?php echo htmlentities(AdminHelper::getSettingsURL(array('sgpbImport' => 1)), ENT_QUOTES); ?>" class="page-title-action">
```

Issue #2375 - popup-builder/com/classes/ConditionCreator.php: 121

Path:	popup-builder/com/classes/ConditionCreator.php
Line:	121
Sink:	echo
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'conditionName' is received in line 607 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addConditionRuleRow().

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
602  public function addConditionRuleRow()
603  {
⋮
606  global $SGPB_DATA_CONFIG_ARRAY;
607  $targetType = sanitize_text_field($_POST['conditionName']);
608  $builderObj = new ConditionBuilder();
⋮
618  $data .= ConditionCreator::createConditionRuleRow($builderObj);
⋮
622  }
⋮
683  }
```

The user-supplied data is concatenated into html markup in line 121 of the file popup-builder/com/classes/ConditionCreator.php in the method sgpbConditionCreator::createConditionRuleRow().

The user-supplied data is then used unsanitized in the sensitive operation echo() in line 121 of the file popup-builder/com/classes/ConditionCreator.php in the method sgpbConditionCreator::createConditionRuleRow(). Please refer to the context and description for further information.

popup-builder/com/classes/ConditionCreator.php

```
3    class ConditionCreator
4    {
⋮
80   public static function createConditionRuleRow($conditionDataObj)
81   {
⋮
93   <?php $idHiddenDiv = $conditionDataObj->getConditionName().'_'.$conditionDataObj->getGroupId().'_'.$conditionDataObj->getRuleId();?>
⋮
118  <div class="sg-hide-condition-row"><div id="<?php echo $idHiddenDiv;?>"><?php echo $hiddenContent; ?></div></div>
⋮
121  <?php echo self::createConditionOperators($conditionDataObj, $idHiddenDiv); ?>
⋮
129  }
⋮
656  }
```

HTML Context

The following snippet(s) do not represent actual code but the tainted context.

```
<a href="javascript:void(0)" class="sg-rules-G_13_-rule btn btn-primary btn-xs" data-id=" $_POST['conditionName'] __"><span>1</span></a></div>
```

Patch

HTML htmlentities() Encoding In Attribute Context

popup-builder/com/classes/ConditionCreator.php

```
121  <?php echo htmlentities(self::createConditionOperators($conditionDataObj, $idHiddenDiv), ENT_QUOTES); ?>
```

3.8. Information Leakage

ASVS:	4.0.1: 7.4.1
OWASP Top 10:	2017: A6
CWE:	209
PCI DSS:	6.5.5
Severity:	Low

An information leakage vulnerability occurs when confidential information about the web server's setup or the application’s inner workings is leaked to the application's user. Although the issue might not be exploitable, it can help an attacker to prepare other attacks.

The affected code might be leftover debug code. In such a case, it should be removed before running the code in production.

3.8.1. Information Leakage (system)

ASVS:	4.0.1: 7.4.1
OWASP Top 10:	2017: A6
CWE:	214
PCI DSS:	6.5.5
Severity:	Low

The affected code leaks information about the system that allows an attacker to learn about used software versions or installation paths.

The affected code might be leftover debug code. In such a case, it should be removed before running the code in production.

Issue #2293 - popup-builder/public/views/settingsOptions.php: 81

Path:	popup-builder/public/views/settingsOptions.php
Line:	81
Sink:	echo
Taint:	HTTP

Code Summary

User-supplied data is concatenated into info markup in line 81 of the file popup-builder/public/views/settingsOptions.php.

The operation echo() leaks sensitive system information. It is located in line 81 of the file popup-builder/public/views/settingsOptions.php. Please refer to the context and description for further information.

popup-builder/public/views/settingsOptions.php

```
10  $systemInfo = AdminHelper::getSystemInfoText();
 ⋮
81  <textarea onclick="this.select();" rows="10" class="form-control" readonly><?php echo $systemInfo ;?></textarea>
```

Info Context

The following snippet(s) do not represent actual code but the tainted context.

```
PHP version
```

3.9. Dynamic SQL Query

CWE: 89
Severity: Low

A SQL query is constructed dynamically by concatenation. This can lead to SQL injection attacks.

It is recommended to use prepared statements for all SQL queries. The prepared statement itself should only use placeholders for data and never concatenate data directly into the query.

Issue #2296 - popup-builder/com/classes/Installer.php: 252

Path: popup-builder/com/classes/Installer.php
Line: 252
Sink: execute
Taint: HTTP

Code Summary

A code quality issue was detected in line 252 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::deleteCustomTables(). Please refer to the context and description for further information.

popup-builder/com/classes/Installer.php

```
5    class Installer
6    {
⋮
239  private static function deleteCustomTables($blogId = '')
240  {
241  $allTableNames = self::getAllTableNames();
⋮
248  foreach ($allTableNames as $tableName) {
249  $deleteTable = $wpdb->prefix.$blogId.$tableName;
250  $deleteTableSql = 'DROP TABLE '.$deleteTable;
⋮
252  $wpdb->query($deleteTableSql);
253  }
⋮
256  }
⋮
329  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DROP TABLE Array
```

Issue #2301 - popup-builder/com/helpers/AdminHelper.php: 883

Path: popup-builder/com/helpers/AdminHelper.php
Line: 883
Sink: execute
Source: _GET

Taint: HTTP

Code Summary

A code quality issue was detected in line 883 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteUserFromSubscribers(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
864   public static function deleteUserFromSubscribers($params = array())
865   {
⋮
883   $prepareSql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE
      email = %s && subscriptionType = %s', $email, $popup);
⋮
898   }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = $_GET['email'] && subscriptionType =
```

Issue #2314 - popup-builder/com/classes/Installer.php: 18

Path: popup-builder/com/classes/Installer.php
Line: 18
Sink: execute
Taint: HTTP

Code Summary

A code quality issue was detected in line 18 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::createTables(). Please refer to the context and description for further information.

popup-builder/com/classes/Installer.php

```
5     class Installer
6     {
7     public static function createTables($tables, $blogId = '')
8     {
⋮
14    foreach ($tables as $table) {
15    $createTable = 'CREATE TABLE IF NOT EXISTS ';
16    $createTable .= $wpdb->prefix.$blogId;
17    $createTable .= $table;
18    $wpdb->query($createTable);
19    }
⋮
22    }
⋮
329   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
CREATE TABLE IF NOT EXISTS
```

Issue #2317 - popup-builder/com/helpers/AdminHelper.php: 959

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	959
Sink:	execute
Taint:	HTTP

Code Summary

A code quality issue was detected in line 959 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::addUnsubscribeColumn(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9      class AdminHelper
10     {
 ⋮
954    public static function addUnsubscribeColumn()
955    {
956    global $wpdb;
 ⋮
958    $sql = 'ALTER TABLE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' ADD COLUMN unsubscribed INT NOT
       NULL DEFAULT 0 ';
959    $wpdb->query($sql);
960    }
 ⋮
2167   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
ALTER TABLE SGPB_SUBSCRIBERS_TABLE_NAME ADD COLUMN unsubscribed INT NOT NULL DEFAULT 0
```

Issue #2350 - popup-builder/com/classes/Actions.php: 1202

Path:	popup-builder/com/classes/Actions.php
Line:	1202
Sink:	get_results
Taint:	HTTP

Code Summary

A code quality issue was detected in line 1202 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::getSubscribersCsvFile(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7      class Actions
```

```
8      {
⋮
1183  public function getSubscribersCsvFile()
1184  {
⋮
1186  $query = AdminHelper::subscribersRelatedQuery();
⋮
1189  $query .= ' ORDER BY '.esc_sql($_GET['orderby']).' '.esc_sql($_GET['order']);
⋮
1202  $subscribers = $wpdb->get_results($query, ARRAY_A);
⋮
1226  }
⋮
1258  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT firstName, lastName, email, cDate, SGPB_POSTS_TABLE_NAME.post_title AS subscriptionTitle FROM
SGPB_SUBSCRIBERS_TABLE_NAME LEFT JOIN SGPB_POSTS_TABLE_NAME ON SGPB_POSTS_TABLE_NAME.ID=
SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' %' ORDER BY
```

Issue #2356 - popup-builder/com/classes/Ajax.php: 267

Path:	popup-builder/com/classes/Ajax.php
Line:	267
Sink:	execute
Source:	_POST
Taint:	HTTP

Code Summary

A code quality issue was detected in line 267 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5     class Ajax
6     {
⋮
254  public function addSubscribers()
255  {

⋮
262  $email = sanitize_text_field($_POST['email']);
⋮
266  foreach ($subscriptionPopupsId as $subscriptionPopupId) {
267  $selectSql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE email = %s AND subscriptionType = %d', $email, $subscriptionPopupId);
268  $res = $wpdb->get_row($selectSql, ARRAY_A);
269  // add new subscriber
270  if (empty($res)) {
271  $sql = $wpdb->prepare('INSERT INTO '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $subscriptionPopupId);
272  $res = $wpdb->query($sql);
273  }
274  // edit existing
275  else {
```

```
276 $sql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d, unsubscribered = 0 WHERE id = %d', $firstName, $lastName, $email, $date, $subscriptionPopupId, $res['id']);
277 $wpdb->query($sql);
278 $res = 1;
279 }
 ⋮
281 if ($res) {
282 $status = SGPB_AJAX_STATUS_TRUE;
283 }
284 }
 ⋮
288 }
 ⋮
683 }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = $_POST['email'] AND subscriptionType = 1
```

Issue #2358 - popup-builder/com/classes/Ajax.php: 271

Path:	popup-builder/com/classes/Ajax.php
Line:	271
Sink:	execute
Source:	_POST
Taint:	HTTP

Code Summary

A code quality issue was detected in line 271 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5   class Ajax
6   {
 ⋮
254 public function addSubscribers()
255 {
 ⋮
260 $firstName = sanitize_text_field($_POST['firstName']);
 ⋮
262 $email = sanitize_text_field($_POST['email']);
 ⋮
264 $subscriptionPopupsId = array_map('sanitize_text_field', $_POST['popups']);
 ⋮
266 foreach ($subscriptionPopupsId as $subscriptionPopupId) {
267 $selectSql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE email = %s AND subscriptionType = %d', $email, $subscriptionPopupId);
268 $res = $wpdb->get_row($selectSql, ARRAY_A);
269 // add new subscriber
270 if (empty($res)) {
271 $sql = $wpdb->prepare('INSERT INTO '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $subscriptionPopupId);
272 $res = $wpdb->query($sql);
273  }
274  // edit existing
275  else {
276  $sql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d, unsubscribered = 0 WHERE id = %d', $firstName, $lastName, $email, $date, $subscriptionPopupId, $res['id']);
277  $wpdb->query($sql);
278  $res = 1;
279  }
⋮
281  if ($res) {
282  $status = SGPB_AJAX_STATUS_TRUE;
283  }
284  }
⋮
288  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (firstName, lastName, email, cDate, subscriptionType) VALUES (
$_POST['firstName'] , , , , 1)
```

Issue #2360 - popup-builder/com/classes/Ajax.php: 276

Path:	popup-builder/com/classes/Ajax.php
Line:	276
Sink:	execute
Source:	_POST
Taint:	HTTP

Code Summary

A code quality issue was detected in line 276 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
254  public function addSubscribers()
255  {
⋮
260  $firstName = sanitize_text_field($_POST['firstName']);
⋮
262  $email = sanitize_text_field($_POST['email']);
⋮
264  $subscriptionPopupsId = array_map('sanitize_text_field', $_POST['popups']);
⋮
266  foreach ($subscriptionPopupsId as $subscriptionPopupId) {
267  $selectSql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE email = %s AND subscriptionType = %d', $email, $subscriptionPopupId);
268  $res = $wpdb->get_row($selectSql, ARRAY_A);
269  // add new subscriber
270  if (empty($res)) {
```

```
271  $sql = $wpdb->prepare('INSERT INTO '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $subscriptionPopupId);
272  $res = $wpdb->query($sql);
273  }
274  // edit existing
275  else {
276  $sql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d, unsubscribered = 0 WHERE id = %d', $firstName, $lastName, $email, $date, $subscriptionPopupId, $res['id']);
277  $wpdb->query($sql);
278  $res = 1;
279  }
⋮
281  if ($res) {
282  $status = SGPB_AJAX_STATUS_TRUE;
283  }
284  }
⋮
288  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET firstName = $_POST['firstName'] , lastName = , email = , cDate = ,
subscriptionType = 1, unsubscribered = 0 WHERE id = 1
```

Issue #2369 - popup-builder/com/classes/Ajax.php: 456

Path:	popup-builder/com/classes/Ajax.php
Line:	456
Sink:	execute
Source:	_POST
Taint:	HTTP

Code Summary

A code quality issue was detected in line 456 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
425  public function subscriptionSubmission()
426  {
⋮
428  $this->setPostData($_POST);
429  $submissionData = $this->getValueFromPost('formData');
⋮
432  parse_str($submissionData, $formData);
⋮
446  global $wpdb;
⋮
450  $email = sanitize_email($formData['sgpb-subs-email']);
⋮
454  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
 ⋮
456  $getSubscriberQuery = $wpdb->prepare('SELECT id FROM '.$subscribersTableName.' WHERE email = %s AND s
     ubscriptionType = %d', $email, $popupPostId);
 ⋮
475  }
 ⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = $_POST['formData']['sgpb-subs-email'] AND
subscriptionType = 1
```

Issue #2371 - popup-builder/com/classes/Ajax.php: 461

Path:	popup-builder/com/classes/Ajax.php
Line:	461
Sink:	execute
Source:	_POST
Taint:	HTTP

Code Summary

A code quality issue was detected in line 461 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
 ⋮
425  public function subscriptionSubmission()
426  {
 ⋮
428  $this->setPostData($_POST);

429  $submissionData = $this->getValueFromPost('formData');
430  $popupPostId = (int)$this->getValueFromPost('popupPostId');
 ⋮
432  parse_str($submissionData, $formData);
 ⋮
446  global $wpdb;
 ⋮

450  $email = sanitize_email($formData['sgpb-subs-email']);
451  $firstName = sanitize_text_field($formData['sgpb-subs-first-name']);
 ⋮
454  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
 ⋮
456  $getSubscriberQuery = $wpdb->prepare('SELECT id FROM '.$subscribersTableName.' WHERE email = %s AND s
     ubscriptionType = %d', $email, $popupPostId);
 ⋮
461  $sql = $wpdb->prepare('INSERT INTO '.$subscribersTableName.' (firstName, lastName, email, cDate, subscripti
     onType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $popupPostId);
 ⋮
475  }
 ⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (firstName, lastName, email, cDate, subscriptionType) VALUES (
$_POST['formData']['sgpb-subs-first-name'] , , 1, , 1)
```

Issue #2373 - popup-builder/com/classes/Ajax.php: 465

Path:	popup-builder/com/classes/Ajax.php
Line:	465
Sink:	execute
Source:	_POST
Taint:	HTTP

Code Summary

A code quality issue was detected in line 465 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5     class Ajax
6     {
⋮
425   public function subscriptionSubmission()
426   {
⋮
428   $this->setPostData($_POST);
429   $submissionData = $this->getValueFromPost('formData');
430   $popupPostId = (int)$this->getValueFromPost('popupPostId');
⋮
432   parse_str($submissionData, $formData);
⋮
446   global $wpdb;
⋮
450   $email = sanitize_email($formData['sgpb-subs-email']);
451   $firstName = sanitize_text_field($formData['sgpb-subs-first-name']);
⋮
454   $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
⋮
456   $getSubscriberQuery = $wpdb->prepare('SELECT id FROM '.$subscribersTableName.' WHERE email = %s AND s
      ubscriptionType = %d', $email, $popupPostId);
⋮
461   $sql = $wpdb->prepare('INSERT INTO '.$subscribersTableName.' (firstName, lastName, email, cDate, subscripti
      onType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $popupPostId);
⋮
      $sql = $wpdb->prepare('UPDATE '.$subscribersTableName.' SET firstName = %s, lastName = %s, email = %s, c
465   Date = %s, subscriptionType = %d WHERE id = %d', $firstName, $lastName, $email, $date, $popupPostId, $list['
      id']);
⋮
475   }
⋮
683   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET firstName = $_POST['formData']['sgpb-subs-first-name'] , lastName =
, email = 1, cDate = , subscriptionType = 1 WHERE id = 1
```

Issue #2377 - popup-builder/com/classes/Installer.php: 192

Path:	popup-builder/com/classes/Installer.php
Line:	192
Sink:	get_results
Taint:	HTTP

Code Summary

A code quality issue was detected in line 192 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::deleteCustomTerms(). Please refer to the context and description for further information.

popup-builder/com/classes/Installer.php

```
5    class Installer
6    {
⋮
182  public static function deleteCustomTerms($taxonomy)
183  {
184  global $wpdb;
⋮
186  $customTermsQuery = 'SELECT t.name, t.term_id
187  FROM '.$wpdb->terms . ' AS t
188  INNER JOIN ' . $wpdb->term_taxonomy . ' AS tt
189  ON t.term_id = tt.term_id
190  WHERE tt.taxonomy = "'.$taxonomy.'"';
⋮
192  $terms = $wpdb->get_results($customTermsQuery);
⋮
202  }
⋮
329  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT t.name, t.term_id FROM AS t INNER JOIN AS tt ON t.term_id = tt.term_id WHERE tt.taxonomy = " "
```

Issue #2382 - popup-builder/com/classes/popups/SGPopup.php: 1641

Path:	popup-builder/com/classes/popups/SGPopup.php
Line:	1641
Sink:	get_var
Taint:	HTTP

Code Summary

A code quality issue was detected in line 1641 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::getPopupOpeningCountById(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10    abstract class SGPopup
11    {
⋮
1633  public function getPopupOpeningCountById($popupId)
1634  {
1635  global $wpdb;
⋮
1640  $tableName = $wpdb->prefix.'sgpb_analytics';
1641  if ($wpdb->get_var("SHOW TABLES LIKE '$tableName'") == $tableName) {
⋮
1650  }
⋮
1689  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SHOW TABLES LIKE ' sgpb_analytics'
```

Issue #2384 - popup-builder/com/helpers/Functions.php: 177

Path:	popup-builder/com/helpers/Functions.php
Line:	177
Sink:	get_results
Taint:	HTTP

Code Summary

A code quality issue was detected in line 177 of the file popup-builder/com/helpers/Functions.php in the method sgpbFunctions::getDatabaseEngine(). Please refer to the context and description for further information.

popup-builder/com/helpers/Functions.php

```
4    class Functions
5    {
⋮
171  public static function getDatabaseEngine()
172  {
173  global $wpdb;
174  $dbName = $wpdb->dbname;
⋮
176  $engineCheckSql = "SELECT ENGINE FROM information_schema.TABLES WHERE TABLE_SCHEMA = '$dbName'";
177  $result = $wpdb->get_results($engineCheckSql, ARRAY_A);
⋮
187  }
⋮
261  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT ENGINE FROM information_schema.TABLES WHERE TABLE_SCHEMA = ' '
```

Issue #2385 - popup-builder/com/helpers/Functions.php: 180

Path:	popup-builder/com/helpers/Functions.php

Line: 180
Sink: get_results
Taint: HTTP

Code Summary

A code quality issue was detected in line 180 of the file popup-builder/com/helpers/Functions.php in the method sgpbFunctions::getDatabaseEngine(). Please refer to the context and description for further information.

popup-builder/com/helpers/Functions.php

```
4     class Functions
5     {
⋮
171   public static function getDatabaseEngine()
172   {
⋮
179   $engineCheckSql = "SHOW TABLE STATUS WHERE Name = '".$wpdb->prefix."users' AND Engine = 'MyISAM'";
180   $result = $wpdb->get_results($engineCheckSql, ARRAY_A);
⋮
187   }
⋮
261   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SHOW TABLE STATUS WHERE Name = ' users' AND Engine = 'MyISAM'
```

Issue #2389 - popup-builder/com/helpers/AdminHelper.php: 1722

Path: popup-builder/com/helpers/AdminHelper.php
Line: 1722
Sink: get_row
Taint: HTTP

Code Summary

A code quality issue was detected in line 1722 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::getSubscriberDataById(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
1719  public static function getSubscriberDataById($id)
1720  {
1721  global $wpdb;
1722  $result = $wpdb->get_row('SELECT * FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE id='.$id, ARRAY_A);
⋮
1725  }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT * FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE id=
```

Issue #2394 - popup-builder/com/libs/Table.php: 100

Path:	popup-builder/com/libs/Table.php
Line:	100
Sink:	get_results
Taint:	HTTP

Code Summary

A code quality issue was detected in line 100 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items(). Please refer to the context and description for further information.

popup-builder/com/libs/Table.php

```
9    class SGPBTable extends SGPBListTable
10   {
⋮
92   public function prepare_items()
93   {
94   global $wpdb;
⋮
98   $this->customizeQuery($query);
⋮
100  $totalItems = count($wpdb->get_results($query)); //return the total number of affected rows
⋮
150  }
⋮
231  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT firstName, lastName, email, cDate, SGPB_POSTS_TABLE_NAME.post_title AS subscriptionTitle FROM
SGPB_SUBSCRIBERS_TABLE_NAME LEFT JOIN SGPB_POSTS_TABLE_NAME ON SGPB_POSTS_TABLE_NAME.ID=
SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' %'
```

Issue #2395 - popup-builder/com/libs/Table.php: 100

Path:	popup-builder/com/libs/Table.php
Line:	100
Sink:	get_results
Source:	_GET
Taint:	HTTP

Code Summary

A code quality issue was detected in line 100 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items(). Please refer to the context and description for further information.

popup-builder/com/libs/Table.php

```
9    class SGPBTable extends SGPBListTable
10   {
⋮
92   public function prepare_items()
93   {
94   global $wpdb;
⋮
98   $this->customizeQuery($query);
⋮
100  $totalItems = count($wpdb->get_results($query)); //return the total number of affected rows
⋮
150  }
⋮

231  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT firstName, lastName, email, cDate, SGPB_POSTS_TABLE_NAME.post_title AS subscriptionTitle FROM
SGPB_SUBSCRIBERS_TABLE_NAME LEFT JOIN SGPB_POSTS_TABLE_NAME ON SGPB_POSTS_TABLE_NAME.ID=
SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' $_GET['sgpb-subscribers-date'] %'
```

Issue #2396 - popup-builder/com/libs/Table.php: 145

Path:	popup-builder/com/libs/Table.php
Line:	145
Sink:	get_results
Taint:	HTTP

Code Summary

A code quality issue was detected in line 145 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items(). Please refer to the context and description for further information.

popup-builder/com/libs/Table.php

```
9    class SGPBTable extends SGPBListTable
10   {
⋮

92   public function prepare_items()
93   {
⋮
98   $this->customizeQuery($query);
⋮
120  $query .= ' ORDER BY '.$orderby.' '.$order;
⋮
132  $query .= ' LIMIT '.(int)$offset.','.(int)$perPage;
⋮
145  $items = $wpdb->get_results($query, ARRAY_N);
⋮
150  }
⋮
231  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT firstName, lastName, email, cDate, SGPB_POSTS_TABLE_NAME.post_title AS subscriptionTitle FROM
SGPB_SUBSCRIBERS_TABLE_NAME LEFT JOIN SGPB_POSTS_TABLE_NAME ON SGPB_POSTS_TABLE_NAME.ID=
SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' %' ORDER BY LIMIT 1,1
```

Issue #2397 - popup-builder/com/libs/Table.php: 145

Path:	popup-builder/com/libs/Table.php
Line:	145
Sink:	get_results
Source:	_GET
Taint:	HTTP

Code Summary

A code quality issue was detected in line 145 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items(). Please refer to the context and description for further information.

popup-builder/com/libs/Table.php

```
9     class SGPBTable extends SGPBListTable
10    {
⋮
92    public function prepare_items()
93    {
⋮
98    $this->customizeQuery($query);
⋮
120   $query .= ' ORDER BY '.$orderby.' '.$order;
⋮
132   $query .= ' LIMIT '.(int)$offset.','.(int)$perPage;
⋮
145   $items = $wpdb->get_results($query, ARRAY_N);
⋮
150   }
⋮
231   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT firstName, lastName, email, cDate, SGPB_POSTS_TABLE_NAME.post_title AS subscriptionTitle FROM
SGPB_SUBSCRIBERS_TABLE_NAME LEFT JOIN SGPB_POSTS_TABLE_NAME ON SGPB_POSTS_TABLE_NAME.ID=
SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' $_GET['sgpb-subscribers-date'] %' ORDER BY
LIMIT 1,1
```

3.9.1. Dynamic SQL Query (Table)

CWE:	89
Severity:	Low

A SQL query is constructed with a dynamically concatenated table specification. This can lead to SQL injection attacks.

It is not possible to use prepared statements to secure dynamic table names. It is highly recommended to use a whitelist for all possible table names.

Issue #2294 - popup-builder/com/classes/popups/SubscriptionPopup.php: 637

Path:	popup-builder/com/classes/popups/SubscriptionPopup.php
Line:	637
Sink:	get_var
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 637 of the file popup-builder/com/classes/popups/SubscriptionPopup.php in the method sgpbSubscriptionPopup::getSubscribersCount().

The user-supplied data is then used unsanitized in the sensitive operation get_var() in line 637 of the file popup-builder/com/classes/popups/SubscriptionPopup.php in the method sgpbSubscriptionPopup::getSubscribersCount(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SubscriptionPopup.php

```
6     class SubscriptionPopup extends SGPopup
7     {
⋮
634   public static function getSubscribersCount()
635   {
636   global $wpdb;
637   $count = $wpdb->get_var('SELECT COUNT(*) FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME);
⋮
640   }
⋮
685   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT COUNT(*) FROM SGPB_SUBSCRIBERS_TABLE_NAME
```

Issue #2295 - popup-builder/com/classes/Installer.php: 252

Path:	popup-builder/com/classes/Installer.php
Line:	252
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 250 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::deleteCustomTables().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 252 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::deleteCustomTables(). Please refer to the context and description for further information.

popup-builder/com/classes/Installer.php

```
5     class Installer
```

```
6    {
⋮
239  private static function deleteCustomTables($blogId = '')
240  {
241  $allTableNames = self::getAllTableNames();
⋮
248  foreach ($allTableNames as $tableName) {
249  $deleteTable = $wpdb->prefix.$blogId.$tableName;
250  $deleteTableSql = 'DROP TABLE '.$deleteTable;
⋮
252  $wpdb->query($deleteTableSql);
253  }
⋮
256  }
⋮
329  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DROP TABLE Array
```

Issue #2297 - popup-builder/com/helpers/AdminHelper.php: 883

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	883
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 883 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteUserFromSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 883 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteUserFromSubscribers(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9      class AdminHelper
10     {
⋮
864    public static function deleteUserFromSubscribers($params = array())
865    {
⋮
883    $prepareSql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE
       email = %s && subscriptionType = %s', $email, $popup);
⋮
898    }
⋮
2167   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = %s && subscriptionType = %s
```

Issue #2299 - popup-builder/com/helpers/AdminHelper.php: 925

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	925
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 925 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriber().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 925 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriber(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
914   public static function deleteSubscriber($params = array())
915   {
⋮
925   $prepareSql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET unsubscribed = 1 WHERE id = %s ', $params['subscriberId']);
⋮
930   }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET unsubscribed = 1 WHERE id = %s
```

Issue #2300 - popup-builder/com/helpers/AdminHelper.php: 926

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	926
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 925 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriber().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 926 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriber(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
914   public static function deleteSubscriber($params = array())
915   {
⋮
925   $prepareSql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET unsubscribed = 1 WHERE id = %s ', $params['subscriberId']);
926   $wpdb->query($prepareSql);
⋮
930   }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET unsubscribed = 1 WHERE id = %s
```

Issue #2318 - popup-builder/com/classes/ConvertToNewVersion.php: 224

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	224
Sink:	get_results
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 223 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::getAllSavedPopups().

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 224 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::getAllSavedPopups(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
219   private function getAllSavedPopups()
220   {
221   global $wpdb;
⋮
223   $query = 'SELECT `id`, `type`, `title`, `options` from '.$wpdb->prefix.'sg_popup ORDER BY id';
224   $popups = $wpdb->get_results($query, ARRAY_A);
⋮
227   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `id`, `type`, `title`, `options` from sg_popup ORDER BY id
```

Issue #2319 - popup-builder/com/classes/ConvertToNewVersion.php: 169

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	169
Sink:	get_row
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 169 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::convertSettings().

The user-supplied data is then used unsanitized in the sensitive operation get_row() in line 169 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::convertSettings(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
:
166    private function convertSettings()
167    {
168    global $wpdb;
169    $settings = $wpdb->get_row('SELECT options FROM '.$wpdb->prefix .'sg_popup_settings WHERE id = 1', ARRAY_A);
:
198    }
:
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT options FROM sg_popup_settings WHERE id = 1
```

Issue #2320 - popup-builder/com/classes/ConvertToNewVersion.php: 728

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	728
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 728 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 728 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
710   private function popupObjectFromArray($arr)
711   {
⋮
728   $query = $wpdb->prepare('SELECT `url` FROM '.$wpdb->prefix.'sg_image_popup WHERE id = %d', $arr['id']);
⋮
890   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `url` FROM sg_image_popup WHERE id = %d
```

Issue #2321 - popup-builder/com/classes/ConvertToNewVersion.php: 736

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	736
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 736 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 736 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
710   private function popupObjectFromArray($arr)
711   {
⋮
736   $query = $wpdb->prepare('SELECT `content` FROM '.$wpdb->prefix.'sg_html_popup WHERE id = %d', $arr['id'
      ]);
⋮
890   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content` FROM sg_html_popup WHERE id = %d
```

Issue #2322 - popup-builder/com/classes/ConvertToNewVersion.php: 744

Path: popup-builder/com/classes/ConvertToNewVersion.php
Line: 744
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 744 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 744 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
⋮
710    private function popupObjectFromArray($arr)
711    {
⋮
744    $query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_fblike_popup WHERE id =
       %d', $arr['id']);
⋮
890    }
⋮
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `options` FROM sg_fblike_popup WHERE id = %d
```

Issue #2323 - popup-builder/com/classes/ConvertToNewVersion.php: 758

Path: popup-builder/com/classes/ConvertToNewVersion.php
Line: 758
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 758 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 758 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
```

```
7      {
⋮
710    private function popupObjectFromArray($arr)
711    {
⋮
758    $query = $wpdb->prepare('SELECT `url` FROM '.$wpdb->prefix.'sg_shortCode_popup WHERE id = %d', $arr['id']);
⋮
890    }
⋮
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `url` FROM sg_shortCode_popup WHERE id = %d
```

Issue #2324 - popup-builder/com/classes/ConvertToNewVersion.php: 766

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	766
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 766 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 766 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
⋮
710    private function popupObjectFromArray($arr)
711    {
⋮
766    $query = $wpdb->prepare('SELECT `url` FROM '.$wpdb->prefix.'sg_iframe_popup WHERE id = %d', $arr['id']);
⋮
890    }
⋮
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `url` FROM sg_iframe_popup WHERE id = %d
```

Issue #2325 - popup-builder/com/classes/ConvertToNewVersion.php: 773

Path:	popup-builder/com/classes/ConvertToNewVersion.php

Line:	773
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 773 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 773 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
⋮
710    private function popupObjectFromArray($arr)
711    {
⋮
773    $query = $wpdb->prepare('SELECT `url`, `options` FROM '.$wpdb->prefix.'sg_video_popup WHERE id = %d', $
       arr['id']);
⋮
890    }
⋮
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `url`, `options` FROM sg_video_popup WHERE id = %d
```

Issue #2326 - popup-builder/com/classes/ConvertToNewVersion.php: 787

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	787
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 787 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 787 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
```

```
⋮
710   private function popupObjectFromArray($arr)
711   {
⋮
787   $query = $wpdb->prepare('SELECT `content`, `yesButton` as `yesButtonLabel`, `noButton` as `noButtonLabel`, `url` as `restrictionUrl` FROM '.$wpdb->prefix.'sg_age_restriction_popup WHERE id = %d', $arr['id']);
⋮
890   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `yesButton` as `yesButtonLabel`, `noButton` as `noButtonLabel`, `url` as `restrictionUrl` FROM sg_age_restriction_popup WHERE id = %d
```

Issue #2327 - popup-builder/com/classes/ConvertToNewVersion.php: 798

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	798
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 798 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 798 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
710   private function popupObjectFromArray($arr)
711   {
⋮
798   $query = $wpdb->prepare('SELECT `socialContent`, `buttons`, `socialOptions` FROM '.$wpdb->prefix.'sg_social_popup WHERE id = %d', $arr['id']);
⋮
890   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `socialContent`, `buttons`, `socialOptions` FROM sg_social_popup WHERE id = %d
```

Issue #2328 - popup-builder/com/classes/ConvertToNewVersion.php: 813

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	813
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 813 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 813 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6       class ConvertToNewVersion
7       {
⋮
710     private function popupObjectFromArray($arr)
711     {
⋮
813     $query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_subscription_popup WHERE id = %d', $arr['id']);
⋮
890     }
⋮
1263    }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `options` FROM sg_subscription_popup WHERE id = %d
```

Issue #2329 - popup-builder/com/classes/ConvertToNewVersion.php: 828

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	828
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 828 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 828 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6       class ConvertToNewVersion
7       {
```

```
⋮
710   private function popupObjectFromArray($arr)
711   {
⋮
828   $query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_countdown_popup WHERE i
      d = %d', $arr['id']);
⋮
890   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `options` FROM sg_countdown_popup WHERE id = %d
```

Issue #2330 - popup-builder/com/classes/ConvertToNewVersion.php: 842

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	842
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 842 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 842 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
710   private function popupObjectFromArray($arr)
711   {
⋮
842   $query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_contact_form_popup WHER
      E id = %d', $arr['id']);
⋮
890   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `options` FROM sg_contact_form_popup WHERE id = %d
```

Issue #2331 - popup-builder/com/classes/ConvertToNewVersion.php: 856

Path: popup-builder/com/classes/ConvertToNewVersion.php
Line: 856
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 856 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 856 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
⋮
710    private function popupObjectFromArray($arr)
711    {
⋮
856    $query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_popup_mailchimp WHERE id = %d', $arr['id']);
⋮
890    }
⋮
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `options` FROM sg_popup_mailchimp WHERE id = %d
```

Issue #2332 - popup-builder/com/classes/ConvertToNewVersion.php: 871

Path: popup-builder/com/classes/ConvertToNewVersion.php
Line: 871
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 871 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 871 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
```

```
7       {
⋮
710     private function popupObjectFromArray($arr)
711     {
⋮
871     $query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_popup_aweber WHERE id = %d', $arr['id']);
⋮
890     }
⋮
1263    }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `content`, `options` FROM sg_popup_aweber WHERE id = %d
```

Issue #2334 - popup-builder/com/classes/ConvertToNewVersion.php: 519

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	519
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 518 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::getAddonsEventFromPopup().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 519 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::getAddonsEventFromPopup(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6       class ConvertToNewVersion
7       {
⋮
510     private function getAddonsEventFromPopup($popup)
511     {
⋮
516     global $wpdb;
⋮
518     $addonsOptionSqlString = 'SELECT options FROM '.$wpdb->prefix.'sg_popup_addons_connection WHERE popupId = %d and extensionType = "option"';
519     $addonsSql = $wpdb->prepare($addonsOptionSqlString, $popupId);
⋮
539     }
⋮
1263    }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT options FROM sg_popup_addons_connection WHERE popupId = %d and extensionType = "option"
```

Issue #2336 - popup-builder/com/classes/ConvertToNewVersion.php: 133

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	133
Sink:	get_results
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 132 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::convertSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 133 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::convertSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
⋮
129    public function convertSubscribers()
130    {
131    global $wpdb;
132    $subscribersSql = 'SELECT `id`, `firstName`, `lastName`, `email`, `subscriptionType`, `status` from '.$wpdb->
       prefix.'sg_subscribers';
133    $subscribers = $wpdb->get_results($subscribersSql, ARRAY_A);
⋮
146    }
⋮
1263   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT `id`, `firstName`, `lastName`, `email`, `subscriptionType`, `status` from sg_subscribers
```

Issue #2337 - popup-builder/com/classes/ConvertToNewVersion.php: 151

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	151
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 151 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::getPostByTitle().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 151 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::getPostByTitle(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
148   private function getPostByTitle($pageTitle, $output = OBJECT)
149   {
150   global $wpdb;
151   $post = $wpdb->get_var($wpdb->prepare("SELECT ID FROM $wpdb->posts WHERE post_title = %s AND post_type='popupbuilder'", $pageTitle));
⋮
157   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT ID FROM WHERE post_title = %s AND post_type='popupbuilder'
```

Issue #2338 - popup-builder/com/classes/ConvertToNewVersion.php: 143

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	143
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 143 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::convertSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 143 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::convertSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6     class ConvertToNewVersion
7     {
⋮
129   public function convertSubscribers()
130   {
⋮
139   foreach ($subscribers as $subscriber) {
140   $subscriber['subscriptionType'] = $this->getPostByTitle($subscriber['subscriptionType']);
⋮
142   $date = date('Y-m-d');
143   $sql = $wpdb->prepare('INSERT INTO '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' (`firstName`, `lastName`, `email`, `cDate`, `subscriptionType`, `unsubscribed`) VALUES (%s, %s, %s, %s, %d, %d) ', $subscriber['firstName'], $subscriber['lastName'], $subscriber['email'], $date, $subscriber['subscriptionType'], 0);
144   $wpdb->query($sql);
145   }
146   }
⋮
1263  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (`firstName`, `lastName`, `email`, `cDate`, `subscriptionType`,
`unsubscribed`) VALUES (%s, %s, %s, %s, %d, %d)
```

Issue #2341 - popup-builder/com/helpers/AdminHelper.php: 328

Path: popup-builder/com/helpers/AdminHelper.php
Line: 328
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 328 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriptionPopupSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 328 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriptionPopupSubscribers(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
324   public static function deleteSubscriptionPopupSubscribers($popupId)
325   {
326   global $wpdb;
⋮
328   $prepareSql = $wpdb->prepare('DELETE FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE su
      bscriptionType = %s', $popupId);
⋮
330   }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DELETE FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE subscriptionType = %s
```

Issue #2342 - popup-builder/com/helpers/AdminHelper.php: 329

Path: popup-builder/com/helpers/AdminHelper.php
Line: 329
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 328 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriptionPopupSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 329 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::deleteSubscriptionPopupSubscribers(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
324   public static function deleteSubscriptionPopupSubscribers($popupId)
325   {
326   global $wpdb;
⋮
328   $prepareSql = $wpdb->prepare('DELETE FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE subscriptionType = %s', $popupId);
329   $wpdb->query($prepareSql);
330   }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DELETE FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE subscriptionType = %s
```

Issue #2343 - popup-builder/com/classes/Actions.php: 617

Path:	popup-builder/com/classes/Actions.php
Line:	617
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 617 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 617 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7     class Actions
8     {
⋮
574   public function newsletterSendEmail()
575   {
⋮
617   $sql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE status = 0 and unsubscribed = 0 and subscriptionType = %d limit 1', $subscriptionFormId);
⋮
706   }
⋮
1258  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE status = 0 and unsubscribed = 0 and subscriptionType =
%d limit 1
```

Issue #2344 - popup-builder/com/classes/Actions.php: 620

Path:	popup-builder/com/classes/Actions.php
Line:	620
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 620 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 620 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7      class Actions
8      {
⋮
574    public function newsletterSendEmail()
575    {
⋮
620    $getTotalSql = $wpdb->prepare('SELECT count(*) FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.'
       WHERE unsubscribed = 0 and subscriptionType = %d', $subscriptionFormId);
⋮
706    }
⋮
1258   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT count(*) FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE unsubscribed = 0 and subscriptionType = %d
```

Issue #2345 - popup-builder/com/classes/Actions.php: 645

Path:	popup-builder/com/classes/Actions.php
Line:	645
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 645 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 645 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7     class Actions
8     {
⋮
574   public function newsletterSendEmail()
575   {
⋮
645   $getAllDataSql = $wpdb->prepare('SELECT id, firstName, lastName, email FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE unsubscribed = 0 and id >= %d and subscriptionType = %s limit %d', $currentStateEmailId, $subscriptionFormId, $emailsInFlow);
⋮
706   }
⋮
1258  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id, firstName, lastName, email FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE unsubscribed = 0 and id >=
%d and subscriptionType = %s limit %d
```

Issue #2347 - popup-builder/com/classes/Actions.php: 689

Path:	popup-builder/com/classes/Actions.php
Line:	689
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 689 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 689 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7     class Actions
8     {
⋮
574   public function newsletterSendEmail()
575   {
⋮
689   $errorLogSql = $wpdb->prepare('INSERT INTO '. $wpdb->prefix .SGPB_SUBSCRIBERS_ERROR_TABLE_NAME.' (`popupType`, `email`, `date`) VALUES (%s, %s, %s)', $subscriptionFormId, $subscriber['email'], date('Y-m-d H:i'));
⋮
706   }
⋮
1258  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_ERROR_TABLE_NAME (`popupType`, `email`, `date`) VALUES (%s, %s, %s)
```

Issue #2348 - popup-builder/com/classes/Actions.php: 704

Path:	popup-builder/com/classes/Actions.php
Line:	704
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 704 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 704 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7       class Actions
8       {
:
574     public function newsletterSendEmail()
575     {
:
704     $updateStatusQuery = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET status = 1 where id >= %d and subscriptionType = %d limit %d', $currentStateEmailId, $subscriptionFormId, $emailsInFlow);
:
706     }
:
1258    }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET status = 1 where id >= %d and subscriptionType = %d limit %d
```

Issue #2352 - popup-builder/com/classes/Ajax.php: 166

Path:	popup-builder/com/classes/Ajax.php
Line:	166
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 166 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::resetPopupOpeningCount().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 166 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::resetPopupOpeningCount(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
151  public function resetPopupOpeningCount()
152  {
⋮
166  $stmt = $wpdb->prepare(' DELETE FROM '.$wpdb->prefix.'sgpb_analytics WHERE target_id = %d AND event_id
     NOT IN (7, 12, 13)', $popupId);
⋮
171  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DELETE FROM sgpb_analytics WHERE target_id = %d AND event_id NOT IN (7, 12, 13)
```

Issue #2353 - popup-builder/com/classes/Ajax.php: 249

Path:	popup-builder/com/classes/Ajax.php
Line:	249
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 249 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::deleteSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 249 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::deleteSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
240  public function deleteSubscribers()
241  {
⋮
249  $prepareSql = $wpdb->prepare('DELETE FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE id
     = %d', $subscriberId);
⋮
252  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DELETE FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE id = %d
```

Issue #2354 - popup-builder/com/classes/Ajax.php: 250

Path:	popup-builder/com/classes/Ajax.php
Line:	250
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 249 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::deleteSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 250 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::deleteSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5     class Ajax
6     {
⋮
240   public function deleteSubscribers()
241   {
⋮
249   $prepareSql = $wpdb->prepare('DELETE FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE id
      = %d', $subscriberId);
250   $wpdb->query($prepareSql);
⋮
252   }
⋮
683   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
DELETE FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE id = %d
```

Issue #2355 - popup-builder/com/classes/Ajax.php: 267

Path:	popup-builder/com/classes/Ajax.php
Line:	267
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 267 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 267 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5     class Ajax
6     {
⋮
254  public function addSubscribers()
255  {
⋮
266  foreach ($subscriptionPopupsId as $subscriptionPopupId) {
267  $selectSql = $wpdb->prepare('SELECT id FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE email = %s AND subscriptionType = %d', $email, $subscriptionPopupId);
268  $res = $wpdb->get_row($selectSql, ARRAY_A);
269  // add new subscriber
270  if (empty($res)) {
271  $sql = $wpdb->prepare('INSERT INTO '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $subscriptionPopupId);
272  $res = $wpdb->query($sql);
273  }
274  // edit existing
275  else {
276  $sql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d, unsubscribered = 0 WHERE id = %d', $firstName, $lastName, $email, $date, $subscriptionPopupId, $res['id']);
277  $wpdb->query($sql);
278  $res = 1;
279  }
⋮
281  if ($res) {
282  $status = SGPB_AJAX_STATUS_TRUE;
283  }
284  }
⋮
288  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = %s AND subscriptionType = %d
```

Issue #2357 - popup-builder/com/classes/Ajax.php: 271

Path:	popup-builder/com/classes/Ajax.php
Line:	271
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 271 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 271 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
254  public function addSubscribers()
255  {
 ⋮
271  $sql = $wpdb->prepare('INSERT INTO '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $subscriptionPopupId);
 ⋮
288  }
 ⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d)
```

Issue #2359 - popup-builder/com/classes/Ajax.php: 276

Path:	popup-builder/com/classes/Ajax.php
Line:	276
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 276 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 276 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::addSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
 ⋮
254  public function addSubscribers()
255  {
 ⋮
276  $sql = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d, unsubscribered = 0 WHERE id = %d', $firstName, $lastName, $email, $date, $subscriptionPopupId, $res['id']);
 ⋮
288  }
 ⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d, unsubscribered = 0 WHERE id = %d
```

Issue #2362 - popup-builder/com/classes/Ajax.php: 324

Path:	popup-builder/com/classes/Ajax.php

Line: 324
Sink: prepare
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 324 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation prepare() in line 324 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
:
305  public function saveImportedSubscribers()
306  {
:
322  global $wpdb;
323  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
324  $sql = $wpdb->prepare('SELECT submittedData FROM '.$subscribersTableName);
:
338  }
:
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT submittedData FROM SGPB_SUBSCRIBERS_TABLE_NAME
```

Issue #2363 - popup-builder/com/classes/Ajax.php: 326

Path: popup-builder/com/classes/Ajax.php
Line: 326
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 326 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 326 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {

:
305  public function saveImportedSubscribers()
306  {
⋮
322  global $wpdb;
323  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;

⋮
326  $sql = $wpdb->prepare('INSERT INTO '.$subscribersTableName.' (firstName, lastName, email, cDate, subscriptionType, status, unsubscribed) VALUES (%s, %s, %s, %s, %d, %d, %d) ', $csvData[$mapping['firstName']], $csvData[$mapping['lastName']], $csvData[$mapping['email']], $csvData[$mapping['date']], $formId, 0, 0);
⋮
338  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (firstName, lastName, email, cDate, subscriptionType, status, unsubscribed) VALUES (%s, %s, %s, %s, %d, %d, %d)
```

Issue #2364 - popup-builder/com/classes/Ajax.php: 329

Path:	popup-builder/com/classes/Ajax.php
Line:	329
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 329 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 329 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
305  public function saveImportedSubscribers()
306  {
⋮
322  global $wpdb;
323  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
⋮
329  $sql = $wpdb->prepare('INSERT INTO '.$subscribersTableName.' (firstName, lastName, email, cDate, subscriptionType, status, unsubscribed, submittedData) VALUES (%s, %s, %s, %s, %d, %d, %d, %s) ', $csvData[$mapping['firstName']], $csvData[$mapping['lastName']], $csvData[$mapping['email']], $csvData[$mapping['date']], $formId, 0, 0, '');
⋮
338  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (firstName, lastName, email, cDate, subscriptionType, status,
unsubscribed, submittedData) VALUES (%s, %s, %s, %s, %d, %d, %d, %s)
```

Issue #2365 - popup-builder/com/classes/Ajax.php: 332

Path:	popup-builder/com/classes/Ajax.php
Line:	332
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 324 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 332 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::saveImportedSubscribers(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
305  public function saveImportedSubscribers()
306  {
⋮
324  $sql = $wpdb->prepare('SELECT submittedData FROM '.$subscribersTableName);
⋮
332  $wpdb->query($sql);
⋮
338  }

⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT submittedData FROM SGPB_SUBSCRIBERS_TABLE_NAME
```

Issue #2366 - popup-builder/com/classes/Ajax.php: 348

Path:	popup-builder/com/classes/Ajax.php
Line:	348
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 348 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::sendNewsletter().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 348 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::sendNewsletter(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5     class Ajax
6     {
⋮
340   public function sendNewsletter()
341   {
⋮
343   global $wpdb;
⋮
348   $updateStatusQuery = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET stat
      us = 0 WHERE subscriptionType = %d', $subscriptionFormId);
⋮
356   }
⋮
683   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET status = 0 WHERE subscriptionType = %d
```

Issue #2367 - popup-builder/com/classes/Ajax.php: 349

Path:	popup-builder/com/classes/Ajax.php
Line:	349
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 348 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::sendNewsletter().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 349 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::sendNewsletter(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5     class Ajax
6     {
⋮
340   public function sendNewsletter()
341   {
⋮
343   global $wpdb;
⋮
348   $updateStatusQuery = $wpdb->prepare('UPDATE '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' SET stat
      us = 0 WHERE subscriptionType = %d', $subscriptionFormId);
349   $wpdb->query($updateStatusQuery);
⋮
356   }
⋮
683   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET status = 0 WHERE subscriptionType = %d
```

Issue #2368 - popup-builder/com/classes/Ajax.php: 456

Path:	popup-builder/com/classes/Ajax.php
Line:	456
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 456 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 456 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
:
425  public function subscriptionSubmission()
426  {
:
446  global $wpdb;
:
450  $email = sanitize_email($formData['sgpb-subs-email']);
:
454  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
:
456  $getSubscriberQuery = $wpdb->prepare('SELECT id FROM '.$subscribersTableName.' WHERE email = %s AND s
     ubscriptionType = %d', $email, $popupPostId);
:
475  }
:
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = %s AND subscriptionType = %d
```

Issue #2370 - popup-builder/com/classes/Ajax.php: 461

Path:	popup-builder/com/classes/Ajax.php
Line:	461
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 461 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 461 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
425  public function subscriptionSubmission()
426  {
⋮
446  global $wpdb;
⋮
454  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
⋮
461  $sql = $wpdb->prepare('INSERT INTO '.$subscribersTableName.' (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d) ', $firstName, $lastName, $email, $date, $popupPostId);
⋮
475  }
⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
INSERT INTO SGPB_SUBSCRIBERS_TABLE_NAME (firstName, lastName, email, cDate, subscriptionType) VALUES (%s, %s, %s, %s, %d)
```

Issue #2372 - popup-builder/com/classes/Ajax.php: 465

Path:	popup-builder/com/classes/Ajax.php
Line:	465
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 465 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 465 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::subscriptionSubmission(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
⋮
425  public function subscriptionSubmission()
426  {
⋮
446  global $wpdb;
⋮
454  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
⋮
465  $sql = $wpdb->prepare('UPDATE '.$subscribersTableName.' SET firstName = %s, lastName = %s, email = %s, cDate = %s, subscriptionType = %d WHERE id = %d', $firstName, $lastName, $email, $date, $popupPostId, $list['id']);
⋮
475  }
 ⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SGPB_SUBSCRIBERS_TABLE_NAME SET firstName = %s, lastName = %s, email = %s, cDate = %s,
subscriptionType = %d WHERE id = %d
```

Issue #2374 - popup-builder/com/classes/Ajax.php: 511

Path:	popup-builder/com/classes/Ajax.php
Line:	511
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 511 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::sendSuccessEmails().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 511 of the file popup-builder/com/classes/Ajax.php in the method sgpbAjax::sendSuccessEmails(). Please refer to the context and description for further information.

popup-builder/com/classes/Ajax.php

```
5    class Ajax
6    {
 ⋮
501  public function sendSuccessEmails($popupPostId, $subscriptionDetails)
502  {
 ⋮
509  $subscribersTableName = $wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME;
 ⋮
511  $getSubscriberCountQuery = $wpdb->prepare('SELECT COUNT(id) as countIds FROM '.$subscribersTableName.'
     WHERE subscriptionType = %d', $popupPostId);
 ⋮
535  }
 ⋮
683  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT COUNT(id) as countIds FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE subscriptionType = %d
```

Issue #2376 - popup-builder/com/classes/Installer.php: 192

Path:	popup-builder/com/classes/Installer.php
Line:	192
Sink:	get_results
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 186 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::deleteCustomTerms().

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 192 of the file popup-builder/com/classes/Installer.php in the method sgpbInstaller::deleteCustomTerms(). Please refer to the context and description for further information.

popup-builder/com/classes/Installer.php

```
5    class Installer
6    {
⋮
182  public static function deleteCustomTerms($taxonomy)
183  {
184  global $wpdb;
⋮
186  $customTermsQuery = 'SELECT t.name, t.term_id
187  FROM '.$wpdb->terms . ' AS t
188  INNER JOIN ' . $wpdb->term_taxonomy . ' AS tt

189  ON t.term_id = tt.term_id
190  WHERE tt.taxonomy = "'.$taxonomy.'"';
⋮
192  $terms = $wpdb->get_results($customTermsQuery);
⋮
202  }
⋮
329  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT t.name, t.term_id FROM AS t INNER JOIN AS tt ON t.term_id = tt.term_id WHERE tt.taxonomy = " "
```

Issue #2378 - popup-builder/com/classes/popups/SubscriptionPopup.php: 674

Path:	popup-builder/com/classes/popups/SubscriptionPopup.php
Line:	674
Sink:	get_results
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 674 of the file popup-builder/com/classes/popups/SubscriptionPopup.php in the method sgpbSubscriptionPopup::getAllSubscribersDate().

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 674 of the file popup-builder/com/classes/popups/SubscriptionPopup.php in the method sgpbSubscriptionPopup::getAllSubscribersDate(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SubscriptionPopup.php

```
6    class SubscriptionPopup extends SGPopup
7    {
⋮
670  public static function getAllSubscribersDate()
{
⋮
global $wpdb;
$subscriptionPopups = $wpdb->get_results('SELECT id, cDate FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME, ARRAY_A);
⋮
}
}
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT id, cDate FROM SGPB_SUBSCRIBERS_TABLE_NAME
```

Issue #2383 - popup-builder/com/classes/popups/SGPopup.php: 1656

Path:	popup-builder/com/classes/popups/SGPopup.php
Line:	1656
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 1656 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::getAnalyticsDataByPopupId().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 1656 of the file popup-builder/com/classes/popups/SGPopup.php in the method sgpbSGPopup::getAnalyticsDataByPopupId(). Please refer to the context and description for further information.

popup-builder/com/classes/popups/SGPopup.php

```
10    abstract class SGPopup
11    {
⋮
1652  public static function getAnalyticsDataByPopupId($popupId)
1653  {
1654  global $wpdb;
⋮
1656  $stmt = $wpdb->prepare('SELECT COUNT(*) FROM '.$wpdb->prefix.'sgpb_analytics WHERE target_id = %d AND event_id NOT IN (7, 12, 13)', $popupId);
⋮
1660  }
⋮
1689  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT COUNT(*) FROM sgpb_analytics WHERE target_id = %d AND event_id NOT IN (7, 12, 13)
```

Issue #2386 - popup-builder/com/helpers/AdminHelper.php: 1598

Path:	popup-builder/com/helpers/AdminHelper.php

Line: 1598

Sink: execute

Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 1598 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::findSubscribersByEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 1598 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::findSubscribersByEmail(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
1593  public static function findSubscribersByEmail($subscriberEmail = '', $list = 0)
1594  {
1595  global $wpdb;
⋮
1598  $prepareSql = $wpdb->prepare('SELECT * FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE email = %s AND subscriptionType = %d ', $subscriberEmail, $list);
⋮
1606  }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT * FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = %s AND subscriptionType = %d
```

Issue #2387 - popup-builder/com/helpers/AdminHelper.php: 1601

Path: popup-builder/com/helpers/AdminHelper.php

Line: 1601

Sink: execute

Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 1601 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::findSubscribersByEmail().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 1601 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::findSubscribersByEmail(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
```

```
⋮
1593  public static function findSubscribersByEmail($subscriberEmail = '', $list = 0)
1594  {
⋮
1601  $prepareSql = $wpdb->prepare('SELECT * FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE email = %s ', $subscriberEmail);
⋮
1606  }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT * FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE email = %s
```

Issue #2388 - popup-builder/com/helpers/AdminHelper.php: 1722

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	1722
Sink:	get_row
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 1722 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::getSubscriberDataById().

The user-supplied data is then used unsanitized in the sensitive operation get_row() in line 1722 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::getSubscriberDataById(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
1719  public static function getSubscriberDataById($id)
1720  {
1721  global $wpdb;
1722  $result = $wpdb->get_row('SELECT * FROM '.$wpdb->prefix.SGPB_SUBSCRIBERS_TABLE_NAME.' WHERE id='.$id, ARRAY_A);
⋮
1725  }
⋮
2167  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT * FROM SGPB_SUBSCRIBERS_TABLE_NAME WHERE id=
```

Issue #2390 - popup-builder/com/libs/Importer.php: 1105

Path:	popup-builder/com/libs/Importer.php

Line: 1105

Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 1105 of the file popup-builder/com/libs/Importer.php in the method sgpbWP_Import::backfill_attachment_urls().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 1105 of the file popup-builder/com/libs/Importer.php in the method sgpbWP_Import::backfill_attachment_urls(). Please refer to the context and description for further information.

popup-builder/com/libs/Importer.php

```
14    class WP_Import extends WP_Importer
15    {
⋮
1097  public function backfill_attachment_urls()
1098  {
⋮
1103  foreach ($this->url_remap as $from_url => $to_url) {
1104  // remap urls in post_content
1105  $wpdb->query($wpdb->prepare("UPDATE {$wpdb->posts} SET post_content = REPLACE(post_content, %s, %s
      )", $from_url, $to_url));
1106  // remap enclosure urls
1107  $result = $wpdb->query($wpdb->prepare("UPDATE {$wpdb->postmeta} SET meta_value = REPLACE(meta_va
      lue, %s, %s) WHERE meta_key='enclosure'", $from_url, $to_url));
1108  }
1109  }
⋮
1237  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SET post_content = REPLACE(post_content, %s, %s)
```

Issue #2391 - popup-builder/com/libs/Importer.php: 1107

Path: popup-builder/com/libs/Importer.php
Line: 1107
Sink: execute
Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 1107 of the file popup-builder/com/libs/Importer.php in the method sgpbWP_Import::backfill_attachment_urls().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 1107 of the file popup-builder/com/libs/Importer.php in the method sgpbWP_Import::backfill_attachment_urls(). Please refer to the context and description for further information.

popup-builder/com/libs/Importer.php

```
14    class WP_Import extends WP_Importer
15    {
⋮
1097  public function backfill_attachment_urls()
1098  {
⋮
1103  foreach ($this->url_remap as $from_url => $to_url) {
1104  // remap urls in post_content
1105  $wpdb->query($wpdb->prepare("UPDATE {$wpdb->posts} SET post_content = REPLACE(post_content, %s, %s
      )", $from_url, $to_url));
1106  // remap enclosure urls
1107  $result = $wpdb->query($wpdb->prepare("UPDATE {$wpdb->postmeta} SET meta_value = REPLACE(meta_va
      lue, %s, %s) WHERE meta_key='enclosure'", $from_url, $to_url));
1108  }
1109  }
⋮
1237  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
UPDATE SET meta_value = REPLACE(meta_value, %s, %s) WHERE meta_key='enclosure'
```

Issue #2393 - popup-builder/com/libs/ListTable.php: 532

Path:	popup-builder/com/libs/ListTable.php
Line:	532
Sink:	execute
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 532 of the file popup-builder/com/libs/ListTable.php in the method sgpbdatatableSGPBListTable::months_dropdown().

The user-supplied data is then used unsanitized in the sensitive operation execute() in line 532 of the file popup-builder/com/libs/ListTable.php in the method sgpbdatatableSGPBListTable::months_dropdown(). Please refer to the context and description for further information.

popup-builder/com/libs/ListTable.php

```
13    class SGPBListTable {
⋮
517   protected function months_dropdown( $post_type ) {
⋮
532   $months = $wpdb->get_results( $wpdb->prepare( "
533   SELECT DISTINCT YEAR( post_date ) AS year, MONTH( post_date ) AS month
534   FROM $wpdb->posts
535   WHERE post_type = %s
536   ORDER BY post_date DESC
537   ", $post_type ) );
⋮
577   }
⋮
1315  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT DISTINCT YEAR( post_date ) AS year, MONTH( post_date ) AS month FROM WHERE post_type = %s ORDER BY
post_date DESC
```

Issue #2398 - popup-builder/com/libs/Table.php: 100

Path:	popup-builder/com/libs/Table.php
Line:	100
Sink:	get_results
Taint:	HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 369 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::subscribersRelatedQuery().

popup-builder/com/helpers/AdminHelper.php

```
9      class AdminHelper
10     {
:
332    public static function subscribersRelatedQuery($query = '', $additionalColumn = '')
333    {
:
369    $query .= " WHERE $searchQuery";
:
373    }
:
2167   }
```

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 100 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items(). Please refer to the context and description for further information.

popup-builder/com/libs/Table.php

```
9     class SGPBTable extends SGPBListTable
10    {
:
92    public function prepare_items()
93    {
:
100   $totalItems = count($wpdb->get_results($query)); //return the total number of affected rows
:
150   }
:
231   }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT FROM LEFT JOIN SGPB_POSTS_TABLE_NAME ON
SGPB_POSTS_TABLE_NAME.ID=SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' %'
```

Issue #2399 - popup-builder/com/libs/Table.php: 145

Path:	popup-builder/com/libs/Table.php
Line:	145

Sink: get_results

Taint: HTTP

Code Summary

User-supplied data is concatenated into sql markup in line 132 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items().

The user-supplied data is then used unsanitized in the sensitive operation get_results() in line 145 of the file popup-builder/com/libs/Table.php in the method sgpbdatatableSGPBTable::prepare_items(). Please refer to the context and description for further information.

popup-builder/com/libs/Table.php

```
9    class SGPBTable extends SGPBListTable
10   {
⋮
92   public function prepare_items()
93   {
⋮
132  $query .= ' LIMIT '.(int)$offset.','.(int)$perPage;
⋮
145  $items = $wpdb->get_results($query, ARRAY_N);
⋮
150  }
⋮
231  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT FROM LEFT JOIN SGPB_POSTS_TABLE_NAME ON
SGPB_POSTS_TABLE_NAME.ID=SGPB_SUBSCRIBERS_TABLE_NAME.subscriptionType WHERE cDate LIKE ' %' ORDER BY
LIMIT 1,1
```

3.10. Missing Error Handling

CWE: 390
Severity: Low

The application checks for an error, but no error handling code is present.

All errors should be handled by the application to avoid undefined states, crashes, or exposure of sensitive information.

Issue #2292 - popup-builder/com/helpers/AdminHelper.php: 2138

Path: popup-builder/com/helpers/AdminHelper.php
Line: 2138
Sink: if
Taint: HTTP

Code Summary

A code quality issue was detected in line 2138 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::getBrowser(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
2080  public function getBrowser()
2081  {
⋮
2138  if (!preg_match_all($pattern, $uAgent, $matches)) {
2139  // we have no matching number just continue
2140  }
⋮
2166  }
2167  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Empty conditional block.
```

3.11. Missing Default Case

CWE:	478
Severity:	Low

The switch statement has no default case. This can lead to logical errors when the defined cases do not handle all possibilities. Thus, further code can lead to errors or vulnerabilities.

Each switch statement should have a default case to handle the situation where no case was matched.

Issue #2286 - popup-builder/com/classes/Notification.php: 56

Path:	popup-builder/com/classes/Notification.php
Line:	56
Sink:	switch
Taint:	HTTP

Code Summary

A code quality issue was detected in line 56 of the file popup-builder/com/classes/Notification.php in the method sgpbNotification::render(). Please refer to the context and description for further information.

popup-builder/com/classes/Notification.php

```
4   class Notification
5   {
⋮
51  public function render()
52  {
⋮
56  switch ($type) {
57  case 1:
58  $color = '#01B9FF !important';
break;
case 2:
$color = '#28a745 !important';
break;
case 3:
$color = '#dc3545 !important';
break;
}
⋮
}
⋮
}
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Missing default block in switch.
```

Issue #2306 - popup-builder/com/classes/Updates.php: 248

Path:	popup-builder/com/classes/Updates.php
Line:	248
Sink:	switch
Taint:	HTTP

Code Summary

A code quality issue was detected in line 248 of the file popup-builder/com/classes/Updates.php in the method sgpbUpdates::sgpbAdminNotices(). Please refer to the context and description for further information.

popup-builder/com/classes/Updates.php

```
9    class Updates
10   {
⋮
245  public function sgpbAdminNotices()
246  {
⋮
248  switch ($_GET['sl_activation']) {
249  case 'false':
250  $message = urldecode($_GET['message']);
251  ?>
252  <div class="error">
253  <h3><?php echo $message; ?></h3>
254  </div>
255  <?php
256  break;
257  case 'true':
258  break;

259  }
⋮
261  }
262  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Missing default block in switch.
```

Issue #2316 - popup-builder/com/helpers/AdminHelper.php: 1829

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	1829
Sink:	switch
Taint:	HTTP

Code Summary

A code quality issue was detected in line 1829 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::removeSelectedTypeOptions(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9     class AdminHelper
10    {
⋮
1827  public static function removeSelectedTypeOptions($type)
1828  {
1829  switch ($type) {
1830  case 'cron':
1831  $crons = _get_cron_array();
1832  foreach ($crons as $key => $value) {
1833  foreach ($value as $key => $body) {
1834  if (strstr($key, 'sgpb')) {
1835  wp_clear_scheduled_hook($key);
1836  }
1837  }
1838  }
1839  break;
1840  }
1841  }
⋮
2167  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Missing default block in switch.
```

Issue #2333 - popup-builder/com/classes/ConvertToNewVersion.php: 726

Path:	popup-builder/com/classes/ConvertToNewVersion.php
Line:	726
Sink:	switch
Taint:	HTTP

Code Summary

A code quality issue was detected in line 726 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::popupObjectFromArray(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
class ConvertToNewVersion
{
⋮
private function popupObjectFromArray($arr)
{
⋮
switch ($type) {
case 'image':
$query = $wpdb->prepare('SELECT `url` FROM '.$wpdb->prefix.'sg_image_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['url'])) {
$options['image-url'] = $result['url'];
}
break;
case 'html':
$query = $wpdb->prepare('SELECT `content` FROM '.$wpdb->prefix.'sg_html_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
break;
case 'fblike':
$query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_fblike_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
$customOptions = $result['options'];
$customOptions = json_decode($customOptions, true);
⋮
if (!empty($options)) {
$this->setCustomOptions($customOptions);
}
break;
case 'shortcode':
$query = $wpdb->prepare('SELECT `url` FROM '.$wpdb->prefix.'sg_shortCode_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['url'])) {
$this->setContent($result['url']);
}
break;
case 'iframe':
$query = $wpdb->prepare('SELECT `url` FROM '.$wpdb->prefix.'sg_iframe_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
if (!empty($result['url'])) {
$options['iframe-url'] = $result['url'];
}
break;
case 'video':
$query = $wpdb->prepare('SELECT `url`, `options` FROM '.$wpdb->prefix.'sg_video_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
if (!empty($result['url'])) {
$options['video-url'] = $result['url'];
}
⋮
$customOptions = $result['options'];
```

```
$customOptions = json_decode($customOptions, true);
⋮
if (!empty($customOptions)) {
$this->setCustomOptions($customOptions);
}
break;
case 'ageRestriction':
$query = $wpdb->prepare('SELECT `content`, `yesButton` as `yesButtonLabel`, `noButton` as `noButtonLabel`, `url` as `restrictionUrl` FROM '.$wpdb->prefix.'sg_age_restriction_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
unset($result['content']);
if (!empty($result)) {
$this->setCustomOptions($result);
}
break;
case 'social':
$query = $wpdb->prepare('SELECT `socialContent`, `buttons`, `socialOptions` FROM '.$wpdb->prefix.'sg_social_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['socialContent'])) {
$this->setContent($result['socialContent']);
}
⋮
$buttons = json_decode($result['buttons'], true);
$socialOptions = json_decode($result['socialOptions'], true);
⋮
$socialAllOptions = array_merge($buttons, $socialOptions);
⋮
$this->setCustomOptions($socialAllOptions);
break;
case 'subscription':
$query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_subscription_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
⋮
$subsOptions = $result['options'];
$subsOptions = json_decode($subsOptions, true);
⋮
if (!empty($subsOptions)) {
$this->setCustomOptions($subsOptions);
}
break;
case 'countdown':
$query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_countdown_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
$customOptions = $result['options'];
$customOptions = json_decode($customOptions, true);
⋮
if (!empty($options)) {
$this->setCustomOptions($customOptions);
}
break;
```

```
case 'contactForm':
$query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_contact_form_popup WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
$customOptions = $result['options'];
$customOptions = json_decode($customOptions, true);
⋮
if (!empty($options)) {
$this->setCustomOptions($customOptions);
}
break;
case 'mailchimp':
$query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_popup_mailchimp WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
⋮
$customOptions = $result['options'];
$customOptions = json_decode($customOptions, true);
⋮
if (!empty($options)) {
$this->setCustomOptions($customOptions);
}
break;
case 'aweber':
$query = $wpdb->prepare('SELECT `content`, `options` FROM '.$wpdb->prefix.'sg_popup_aweber WHERE id = %d', $arr['id']);
$result = $wpdb->get_row($query, ARRAY_A);
⋮
if (!empty($result['content'])) {
$this->setContent($result['content']);
}
⋮
$customOptions = $result['options'];
$customOptions = json_decode($customOptions, true);
⋮
if (!empty($options)) {
$this->setCustomOptions($customOptions);
}
break;
}
⋮
}
⋮
}
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Missing default block in switch.
```

Issue #2335 - popup-builder/com/classes/ConvertToNewVersion.php: 605

Path: popup-builder/com/classes/ConvertToNewVersion.php

Line:	605
Sink:	switch
Taint:	HTTP

Code Summary

A code quality issue was detected in line 605 of the file popup-builder/com/classes/ConvertToNewVersion.php in the method sgpbConvertToNewVersion::filterOptions(). Please refer to the context and description for further information.

popup-builder/com/classes/ConvertToNewVersion.php

```
6      class ConvertToNewVersion
7      {
:
548    private function filterOptions($options)
549    {
:
605    switch ($themeNumber) {
606    case 1:
607    $options['popup-content-padding'] += 7;
608    break;
609    case 4:
610    case 6:
611    $options['popup-content-padding'] += 12;
612    break;
613    case 2:
614    case 3:
615    $options['popup-content-padding'] += 0;
616    break;
617    case 5:
618    $options['popup-content-padding'] += 5;
619    break;
620    }
:
698    }
:
1263   }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Missing default block in switch.
```

Issue #2392 - popup-builder/com/libs/Importer.php: 53

Path:	popup-builder/com/libs/Importer.php
Line:	53
Sink:	switch
Taint:	HTTP

Code Summary

A code quality issue was detected in line 53 of the file popup-builder/com/libs/Importer.php in the method sgpbWP_Import::dispatch(). Please refer to the context and description for further information.

popup-builder/com/libs/Importer.php

```php
class WP_Import extends WP_Importer
{
⋮
public function dispatch()
{
⋮
switch ($step) {
case 0:
$this->greet();
break;
case 1:
check_admin_referer('import-upload');
if ($this->handle_upload()) {
$this->import_options();
}
break;
case 2:
check_admin_referer('import-wordpress');
$this->fetch_attachments = (!empty($_POST['fetch_attachments']) && $this->allow_fetch_attachments());
$this->id = (int) $_POST['import_id'];
$file = get_attached_file($this->id);
set_time_limit(0);
$this->import($file);
break;
}
⋮
}
⋮
}
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Missing default block in switch.
```

3.12. Weak Hash Function

OWASP Top 10:	
CWE:	328
Severity:	Low

The code uses a hash function that is cryptographically insecure. An attacker may be able to craft different values that produce the same hash, or to find preimages for some or all values in the output space of the hash function. This can be dangerous if the hash function is used in a security context, e.g., for authentication purposes.

A secure hash algorithm should be used. The availability of algorithms depends on the used PHP version. Secure hash algorithms that may be available include SHA-256, SHA-384, and SHA-512, all of which belong to the SHA-2 family of hash functions, as well as SHA3-224, SHA3-256, SHA3-384, and SHA3-512, which belong to the SHA-3 family of hash functions.

Issue #2298 - popup-builder/com/helpers/AdminHelper.php: 907

Path:	popup-builder/com/helpers/AdminHelper.php
Line:	907
Sink:	md5

Taint: HTTP

Code Summary

A weak hash function in the operation md5() is used in line 907 of the file popup-builder/com/helpers/AdminHelper.php in the method sgpbAdminHelper::subscriberExists(). Please refer to the context and description for further information.

popup-builder/com/helpers/AdminHelper.php

```
9      class AdminHelper
10     {
⋮
900    public static function subscriberExists($params = array())
901    {
⋮
907    $realToken = md5($params['subscriberId'].$params['email']);
⋮
912    }
⋮
2167   }
```

Algorithm Context

The following snippet(s) do not represent actual code but the tainted context.

```
weak hash algorithm: MD5
```

Patch

Replace Weak Hash with Strong Hash

popup-builder/com/helpers/AdminHelper.php

```
907  $realToken = hash('sha3-256', $params['subscriberId'] . $params['email']);
```

Issue #2307 - popup-builder/com/libs/EDD_SL_Plugin_Updater.php: 39

Path: popup-builder/com/libs/EDD_SL_Plugin_Updater.php
Line: 39
Sink: md5
Taint: HTTP

Code Summary

A weak hash function in the operation md5() is used in line 39 of the file popup-builder/com/libs/EDD_SL_Plugin_Updater.php in the method sgpbEDD_SL_Plugin_Updater::__construct(). Please refer to the context and description for further information.

popup-builder/com/libs/EDD_SL_Plugin_Updater.php

```
12   class EDD_SL_Plugin_Updater {
⋮
30   public function __construct( $_api_url, $_plugin_file, $_api_data = null ) {
⋮
39   $this->cache_key = md5( serialize( $this->slug . $this->api_data['license'] . $this->beta ) );
⋮
43   }
:
371  }
```

Algorithm Context

The following snippet(s) do not represent actual code but the tainted context.

```
weak hash algorithm: MD5
```

Patch

Replace Weak Hash with Strong Hash

popup-builder/com/libs/EDD_SL_Plugin_Updater.php

```
39  $this->cache_key = hash('sha3-256', serialize($this->slug . $this->api_data['license'] . $this->beta));
```

Issue #2308 - popup-builder/com/libs/EDD_SL_Plugin_Updater.php: 195

Path:	popup-builder/com/libs/EDD_SL_Plugin_Updater.php
Line:	195
Sink:	md5
Taint:	HTTP

Code Summary

A weak hash function in the operation md5() is used in line 195 of the file popup-builder/com/libs/EDD_SL_Plugin_Updater.php in the method sgpbEDD_SL_Plugin_Updater::plugins_api_filter(). Please refer to the context and description for further information.

popup-builder/com/libs/EDD_SL_Plugin_Updater.php

```
12   class EDD_SL_Plugin_Updater {
:
180  public function plugins_api_filter( $_data, $_action = '', $_args = null ) {
:
195  $cache_key = 'edd_api_request_' . md5( serialize( $this->slug . $this->api_data['license'] . $this->beta ) );
:
226  }
:
371  }
```

Algorithm Context

The following snippet(s) do not represent actual code but the tainted context.

```
weak hash algorithm: MD5
```

Patch

Replace Weak Hash with Strong Hash

popup-builder/com/libs/EDD_SL_Plugin_Updater.php

```
195  $cache_key = 'edd_api_request_' . hash('sha3-256', serialize($this->slug . $this->api_data['license'] . $this->beta));
```

Issue #2309 - popup-builder/com/libs/EDD_SL_Plugin_Updater.php: 308

Path: popup-builder/com/libs/EDD_SL_Plugin_Updater.php
Line: 308
Sink: md5
Taint: HTTP

Code Summary

A weak hash function in the operation md5() is used in line 308 of the file popup-builder/com/libs/EDD_SL_Plugin_Updater.php in the method sgpbEDD_SL_Plugin_Updater::show_changelog(). Please refer to the context and description for further information.

popup-builder/com/libs/EDD_SL_Plugin_Updater.php

```
12   class EDD_SL_Plugin_Updater {
 :
292  public function show_changelog() {
 :
308  $cache_key = md5( 'edd_plugin_' . sanitize_key( $_REQUEST['plugin'] ) . '_' . $beta . '_version_info' );
 :
341  }
 :
371  }
```

Algorithm Context

The following snippet(s) do not represent actual code but the tainted context.

```
weak hash algorithm: MD5
```

Patch

Replace Weak Hash with Strong Hash

popup-builder/com/libs/EDD_SL_Plugin_Updater.php

```
308  $cache_key = hash('sha3-256', 'edd_plugin_' . sanitize_key($_REQUEST['plugin']) . '_' . $beta . '_version_info');
```

Issue #2346 - popup-builder/com/classes/Actions.php: 666

Path: popup-builder/com/classes/Actions.php
Line: 666
Sink: md5
Taint: HTTP

Code Summary

A weak hash function in the operation md5() is used in line 666 of the file popup-builder/com/classes/Actions.php in the method sgpbActions::newsletterSendEmail(). Please refer to the context and description for further information.

popup-builder/com/classes/Actions.php

```
7      class Actions
8      {
 :
574    public function newsletterSendEmail()
575    {
 :
```

```
666   $replacementUnsubscribe .= '?sgpbUnsubscribe='.md5($replacementId.$replacementEmail);
⋮
706   }
⋮
1258  }
```

Algorithm Context

The following snippet(s) do not represent actual code but the tainted context.

```
weak hash algorithm: MD5
```

Patch

Replace Weak Hash with Strong Hash

popup-builder/com/classes/Actions.php

```
666  $replacementUnsubscribe .= '?sgpbUnsubscribe='.hash('sha3-256', $replacementId . $replacementEmail);
```

3.13. Loop Iteration Change

CWE:	834
Severity:	Low

The application performs a loop over an iteration variable. Within this loop, the iteration variable is changed. This can lead to insufficient limitation or even unlimited loop executions with excessive resource consumption or crashes.

If it is not intended to override the loop counter dynamically within the body, a new variable can be introduced and used.

Issue #2311 - popup-builder/com/classes/Filters.php: 262

Path:	popup-builder/com/classes/Filters.php
Line:	262
Sink:	
Taint:	HTTP

Code Summary

A code quality issue was detected in line 262 of the file popup-builder/com/classes/Filters.php in the method sgpbFilters::popupEvents(). Please refer to the context and description for further information.

popup-builder/com/classes/Filters.php

```
8    class Filters
9    {
⋮
256  public function popupEvents($events)
257  {
⋮
262  $events[] = array('param' => 'click');
⋮
276  }
⋮
608  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Change of iteration variable events
```

Issue #2312 - popup-builder/com/classes/Filters.php: 263

Path: popup-builder/com/classes/Filters.php
Line: 263
Sink:
Taint: HTTP

Code Summary

A code quality issue was detected in line 263 of the file popup-builder/com/classes/Filters.php in the method sgpbFilters::popupEvents(). Please refer to the context and description for further information.

popup-builder/com/classes/Filters.php

```
8     class Filters
9     {
⋮
256   public function popupEvents($events)
257   {
⋮
263   $events[] = array('param' => 'hover');
⋮
276   }
⋮
608   }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Change of iteration variable events
```

Issue #2313 - popup-builder/com/classes/Filters.php: 264

Path: popup-builder/com/classes/Filters.php
Line: 264
Sink:
Taint: HTTP

Code Summary

A code quality issue was detected in line 264 of the file popup-builder/com/classes/Filters.php in the method sgpbFilters::popupEvents(). Please refer to the context and description for further information.

popup-builder/com/classes/Filters.php

```
8     class Filters
9     {
⋮
256   public function popupEvents($events)
257  {
⋮
264  $events[] = array('param' => 'confirm');
⋮
276  }
⋮
608  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
Change of iteration variable events
```

Issue #2315 - popup-builder/com/classes/extension/SgpbPopupExtensionActivator.php: 74

Path:	popup-builder/com/classes/extension/SgpbPopupExtensionActivator.php
Line:	74
Sink:	
Taint:	HTTP

Code Summary

A code quality issue was detected in line 74 of the file popup-builder/com/classes/extension/SgpbPopupExtensionActivator.php in the method sgpbPopupExtensionActivator::moveExtensionToPluginsSection(). Please refer to the context and description for further information.

popup-builder/com/classes/extension/SgpbPopupExtensionActivator.php

```
4    class PopupExtensionActivator
5    {
⋮
72   private function moveExtensionToPluginsSection($extensionsInfo)
73   {
74   foreach ($extensionsInfo as $extensionFolder => $extensionsInfo) {
⋮
79   }
⋮
108  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
extensionsInfo
```

Issue #2339 - popup-builder/com/classes/PopupGroupFilter.php: 179

Path:	popup-builder/com/classes/PopupGroupFilter.php
Line:	179
Sink:	
Taint:	HTTP

Code Summary

A code quality issue was detected in line 179 of the file popup-builder/com/classes/PopupGroupFilter.php in the method sgpbPopupGroupFilter::extendPopups(). Please refer to the context and description for further information.

popup-builder/com/classes/PopupGroupFilter.php

```
4    class PopupGroupFilter
5    {
⋮
161  private function extendPopups()
162  {
⋮
179  $popups = array_merge($popups, $insidePopups);
⋮
192  }
193  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
popups
```

Issue #2340 - popup-builder/com/classes/PopupGroupFilter.php: 185

Path:	popup-builder/com/classes/PopupGroupFilter.php
Line:	185
Sink:	
Taint:	HTTP

Code Summary

A code quality issue was detected in line 185 of the file popup-builder/com/classes/PopupGroupFilter.php in the method sgpbPopupGroupFilter::extendPopups(). Please refer to the context and description for further information.

popup-builder/com/classes/PopupGroupFilter.php

```
4    class PopupGroupFilter
5    {
⋮
161  private function extendPopups()
162  {
⋮
185  $popups = array_merge($popups, $subPopups);
⋮
192  }
193  }
```

Code Context

The following snippet(s) do not represent actual code but the tainted context.

```
popups
```