

# **WEB APPLICATION VULNERABILITY ASSESSMENT**

## REPORT

SUBMITTED BY  
CYBER PULSE

# Flash cross-domain policy

## Summary

	Severity:	High
	Confidence:	Certain
	Host:	http://testphp.vulnweb.com
	Path:	/crossdomain.xml

## Issue detail

The application publishes a Flash cross-domain policy which allows access from any domain.

Allowing access from all domains means that any domain can perform two-way interaction with this application. Unless the application consists entirely of unprotected public content, this policy is likely to present a significant security risk.

## Issue background

The Flash cross-domain policy controls whether Flash client components running on other domains can perform two-way interaction with the domain that publishes the policy. If another domain is allowed by the policy, then that domain can potentially attack users of the application. If a user is logged in to the application, and visits a domain allowed by the policy, then any malicious content running on that domain can potentially gain full access to the application within the security context of the logged-in user.

Even if an allowed domain is not overtly malicious in itself, security vulnerabilities within that domain could potentially be leveraged by a third-party attacker to exploit the trust relationship and attack the application that allows access. Any domains that are allowed by the Flash cross-domain policy should be reviewed to determine whether it is appropriate for the application to fully trust both their intentions and security posture.

## Issue remediation

Any inappropriate entries in the Flash cross-domain policy file should be removed.

## Vulnerability classifications

- CWE-942: Overly Permissive Cross-domain Whitelist

# Request

```
GET /crossdomain.xml HTTP/1.1
Host: testphp.vulnweb.com
Connection: close
```


# Response

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 13 Mar 2021 04:21:46 GMT
Content-Type: text/xml
Content-Length: 224
Last-Modified: Tue, 11 Sep 2012 10:30:22 GMT
Connection: close
ETag: "504f12be-e0"
Accept-Ranges: bytes

<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*" to-ports="*" secure="false"/>
...[SNIP]...
```

# Unencrypted communications

## Summary

	Severity:	<b>Low</b>
	Confidence:	<b>Certain</b>
	Host:	<b><a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a></b>
	Path:	<b>/</b>

## Issue description

The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third- party websites. Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.

To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attacker situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure. Please note that using a mixture of encrypted and unencrypted communications is an ineffective defense against active attackers, because they can easily remove references to encrypted resources when these references are transmitted over an unencrypted connection.

## Issue remediation

Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

## References

- Marking HTTP as non-secure
- Configuring Server-Side SSL/TLS
- HTTP Strict Transport Security

## Vulnerability classifications

- CWE-326: Inadequate Encryption Strength

# Cross-domain Referrer leakage

## Summary

	Severity:	<b>Information</b>
	Confidence:	<b>Certain</b>
	Host:	<b><a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a></b>
	Path:	<b><a href="/listproducts.php">/listproducts.php</a></b>

## Issue detail

The page was loaded from a URL containing a query string:

- <http://testphp.vulnweb.com/listproducts.php>

The response contains the following links to other domains:

- <http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab>
- <http://www.acunetix.com/>
- <https://www.acunetix.com/>
- <https://www.acunetix.com/blog/articles/prevent-sql-injection-vulnerabilities-in-php-applications/>
- <https://www.acunetix.com/vulnerability-scanner/>
- <https://www.acunetix.com/vulnerability-scanner/php-security-scanner/>
- <http://www.electasy.com/Fractal-Explorer/index.html>

## Issue background

When a web browser makes a request for a resource, it typically adds an HTTP header, called the "Referer" header, indicating the URL of the resource from which the request originated. This occurs in numerous situations, for example when a web page loads an image or script, or when a user clicks on a link or submits a form.

If the resource being requested resides on a different domain, then the Referer header is still generally included in the cross-domain request. If the originating URL contains any sensitive information within its query string, such as a session token, then this information will be

transmitted to the other domain. If the other domain is not fully trusted by the application, then this may lead to a security compromise.

You should review the contents of the information being transmitted to other domains, and also determine whether those domains are fully trusted by the originating application. Today's browsers may withhold the Referer header in some situations (for example, when loading a non-HTTPS resource from a page that was loaded over HTTPS, or when a Refresh directive is issued), but this behavior should not be relied upon to protect the originating URL from disclosure. Note also that if users can author content within the application then an attacker may be able to inject links referring to a domain they control in order to capture data from URLs used within the application.

## Issue remediation

Applications should never transmit any sensitive information within the URL query string. In addition to being leaked in the Referer header, such information may be logged in various locations and may be visible on-screen to untrusted parties. If placing sensitive information in the URL is unavoidable, consider using the Referer-Policy HTTP header to reduce the chance of it being disclosed to third parties.

## References

- Referer Policy

## Vulnerability classifications

- CWE-200: Information Exposure

## Request

```
GET /listproducts.php?cat=1 HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://testphp.vulnweb.com/categories.php
Upgrade-Insecure-Requests: 1
```

# Response

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 13 Mar 2021 04:34:24 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Length: 7880

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" codeOutsideHTMIsLoc
...[SNIP]...
<p>
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash
/swflash.cab#version=6,0,29,0" width="107" height="66">
<param name="movie" value="Flash/add.swf">
...[SNIP]...
<div id="siteInfo"> <a href="http://www.acunetix.com">About Us</a>
...[SNIP]...
```