Important SSL Security Vulnerability – The Heartbleed Bug

On Monday, April 7th 2014, an OpenSSL vulnerability was disclosed which has been called one of the worst security holes in recent internet history. The bug, called the Heartbleed bug, was introduced in OpenSSL version 1.0.1. It has been in the wild since March of 2012 and is patched with OpenSSL version 1.0.1g released on April 7th 2014. The problem, tagged CVE-2014-0160, is described in detail here:

https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-0160

The bug allows any attacker to read the memory of a vulnerable host, which means that any keys that have been used on a host with a vulnerable version of OpenSSL should be considered compromised. Distributions have been updating their packages and pushing out updates, but users need to pull down the most recent packages and revoke any previous keys based on insecure versions.

For customers running Redhat Linux of CentOS releases, you can simply run “yum update openssl” as root or “sudo yum update openssl” as a sudoer, and then restart any services using OpenSSL to protect any at-risk instances.

Please note that several of the prominent Linux operating systems have released fixed packages that still bear the OpenSSL 1.0.1e name. Even though the OpenSSL project released 1.0.1g as their newest software, downstream Linux providers have in some cases elected to include just the fix for CVE-2014-0160 in their packages in order to provide a small update quickly. Updates to 1.0.1g are likely to come later.

The following instructions show you how to update your systems with a secure version of OpenSSL, revoke any insecure SSL certificates, and test whether you are vulnerable or not.

Update your System

All Linux software repositories used by iomart have been updated to include the newest versions of the OpenSSL packages as they are made available by distribution packagers. The easiest way to update your packages is to update your entire system.

On **Ubuntu and Debian**, you can update by typing:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

On **CentOS and Fedora**, you can type:

```
yum update
```

At the time of this writing, **Fedora 19** does not have the latest versions in the stable repositories yet.

These commands will pull in all of the recent updates, including OpenSSL. You should see openssl or some variant in your list of upgraded packages.

After you are finished, you should reboot your machine to make sure your server is not using the old version loaded in memory. You can do that by issuing:

`sudo shutdown -r` now or `shutdown -r` now as root.

Checking your Version Numbers

You should check your version of OpenSSL after you have updated your system.

While OpenSSL version 1.0.1g is the official fix of this problem, the version that fixes this for different distributions and releases may vary. Some releases and distributions patched their older versions to fix the problem, rather than releasing an entirely new version into an older, stable ecosystem.

Because of this reason, it is best to check through your distribution's packaging system, since the openssl version command might not reflect the information we need.

Debian and Ubuntu Releases and Fix Versions

For Debian and Ubuntu systems, you get the current version of your OpenSSL package by typing:

```
dpkg -l | grep "openssl"
```

You should receive output like this:

```
ii  openssl     1.0.1e-2+deb7u6              amd64        Secure Socket Layer (SSL) binary and
related cryptographic tools
```

For Debian users, the release of Debian that you are running will determine the correct version for the fix. If your version of OpenSSL is at least as recent as the version listed here for your distribution, you should be protected:

- Debian 6 (Squeeze): Unaffected (Shipped with older version prior to vulnerability)
- Debian 7 (Wheezy): 1.0.1e-2+deb7u6
- Debian testing (Jessie): 1.0.1g-1
- Debian unstable (Sid): 1.0.1g-1

For Ubuntu users, the correct, patched version is also release-dependent. Use this list to see the minimum secure version for your release:

- Ubuntu 10.04: Unaffected (Shipped with older version prior to vulnerability)
- Ubuntu 12.04: 1.0.1-4ubuntu5.12
- Ubuntu 12.10: 1.0.1c-3ubuntu2.7
- Ubuntu 13.04: SUPPORT END OF LIFE REACHED, SHOULD UPGRADE
- Ubuntu 13.10: 1.0.1e-3ubuntu1.2

If you are on one of the supported distros, ensure that your OpenSSL version is up-to-date. If your distribution is not supported anymore (Ubuntu 13.04), it is highly recommended that you transition to a supported operating system due to this bug's severity.

CentOS, and Fedora Releases and Fix Versions

For CentOS and Fedora systems, you can query the version of the OpenSSL package installed on your system by typing:

```
rpm -q -a | grep "openssl"
```

You should receive output that looks like this:

```
openssl-1.0.1e-16.el6_5.7.x86_64
```

For CentOS, here are the releases and the minimum versions of OpenSSL that must be applied to protect future SSL interactions. We will take the architecture off the end in our list:

- CentOS 5: Unaffected (Shipped with older version prior to vulnerability)
- CentOS 6: openssl-1.0.1e-16.el6.5.7

For Fedora users, you can check that your package version is at least as recent as the ones listed below. Again, I have removed the architecture below because this applies to both 32-bit and 64-bit releases:

- Fedora 17: Unaffected (Shipped with older version prior to vulnerability)
- Fedora 19: openssl-1.0.1e-37.fc19.1

Note: Pay close attention to the Fedora version number. The trailing ".1" tells you if it is patched or not. If your package does not have the ".1" at the end, you are still vulnerable!

Additional Considerations from a Client's Perspective

Because of the widespread nature of this bug, there are other considerations that you should take into account as well. As a consumer of web services and sites, you should also react quickly to try to minimize the potential damage to your accounts and information.

You should consider any communication that you secured by SSL previously to have been compromised by this bug. This means any kind of interaction with secure websites were open to snooping.

A good first step is to change your password on every site that you use, after you have verified that they have updated their OpenSSL versions to patch this vulnerability. If you change your password prior to the remote site patching their SSL version, your new password is just as vulnerable as your old one.

One consideration that is of high importance is to secure any VPN instances that you have set up. There are a few different ways that VPN connections are implemented, but SSL is one of the most popular. For instance, OpenVPN uses SSL. Any certificates required to connect to your server should be regenerated to ensure that they are secured.

Another good measure is to remove all session keys and cookies. This means regenerate API keys, clear cookies that are stored in your browser, etc. This may be a massive inconvenience, but the cost of not going through these pains now is that your accounts are basically wide open and communication with remote servers that have not updated their OpenSSL should be considered no more secure than plain-text.

Conclusion

This is an incredibly large bug that users and administrators cannot afford to ignore or put off. You should update any servers that you have control over immediately and inform your users of the vulnerabilities so that they can do damage control from their end.

By patching your system and updating your SSL certificates, you should be protected against this vulnerability, as a host, for future communications.