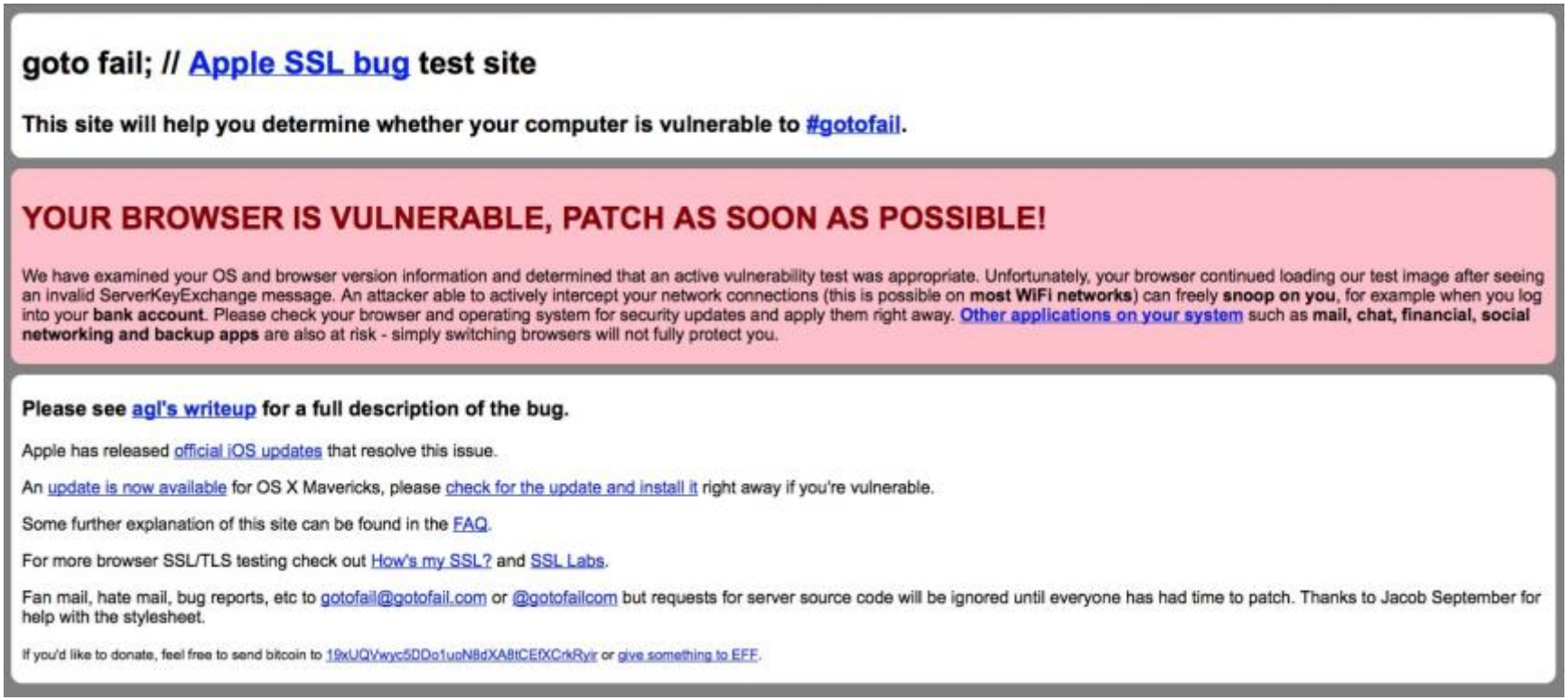
Apple SSL bug

Why it shouldn’t have happened

Your Internet is broken



goto fail; // Apple SSL bug test site

This site will help you determine whether your computer is vulnerable to #gotofail.

YOUR BROWSER IS VULNERABLE, PATCH AS SOON AS POSSIBLE!

We have examined your OS and browser version information and determined that an active vulnerability test was appropriate. Unfortunately, your browser continued loading our test image after seeing an invalid ServerKeyExchange message. An attacker able to actively intercept your network connections (this is possible on **most WiFi networks**) can freely **snoop on you**, for example when you log into your **bank account**. Please check your browser and operating system for security updates and apply them right away. **Other applications on your system** such as **mail, chat, financial, social networking and backup apps** are also at risk - simply switching browsers will not fully protect you.

Please see agl's writeup for a full description of the bug.

Apple has released official iOS updates that resolve this issue.

An update is now available for OS X Mavericks, please check for the update and install it right away if you're vulnerable.

Some further explanation of this site can be found in the FAQ.

For more browser SSL/TLS testing check out How's my SSL? and SSL Labs.

Fan mail, hate mail, bug reports, etc to gotofail@gotofail.com or @gotofailcom but requests for server source code will be ignored until everyone has had time to patch. Thanks to Jacob September for help with the stylesheet.

If you'd like to donate, feel free to send bitcoin to 19xUQVwyc5DDq1uoN8dXA8tCEfXCrkRyir or give something to EFF.

Disclosure failure

iOS 7.0.6

- Data Security

Available for: iPhone 4 and later, iPod touch (5th generation), iPad 2 and later

Impact: An attacker with a privileged network position may capture or modify data in sessions protected by SSL/TLS

Description: Secure Transport failed to validate the authenticity of the connection. This issue was addressed by restoring missing validation steps.

CVE-ID

CVE-2014-1266

Last Modified: Feb 21, 2014

Single line error

```
    if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

        err = sslRawVerify(ctx,
                           ctx->peerPubKey,
                           dataToSign,                          /* plaintext */
                           dataToSignLen,                  /* plaintext length */
                           signature,
                           signatureLen);
        if(err) {
                sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                    "returned %d\n", (int)err);
                goto fail;
        }

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
```

Simpler example

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
        if (argc > 1)
                goto err;
                goto err;
        printf("%s\n", argv[0]);
        return 0;
 err:
        return argc - 1;
}
```

Correctly use compilers

```
~ % gcc bug.c -o bug
~ % clang bug.c -o bug
~ % gcc -Wall -Wextra bug.c -o bug
~ % clang -Wall -Wextra bug.c -o bug
~ %
~ % clang -Wall -Wextra -Werror -Wunreachable-code bug.c -o bug
bug.c:8:2: error: will never be executed [-Werror,-Wunreachable-code]
        printf("%s\n", argv[0]);
        ^~~~~~
1 error generated.
~ %
```

Re-indent automagically

```
~ % cp bug.c bug.c.old
'bug.c' -> 'bug.c.old'
~ % indent -linux bug.c
~ % colordiff -u bug.c.old bug.c
--- bug.c.old
+++ bug.c
@@ -4,7 +4,7 @@
 {
         if (argc > 1)
                 goto err;
-                goto err;
+        goto err;
         printf("%s\n", argv[0]);
         return 0;
   err:
~ %
```

Testing: unit test

```
~ % cat bug.c
#include <stdio.h>

int main(int argc, char *argv[])
{
        if (argc > 1)
                goto err;
        goto err;
        printf("%s\n", argv[0]);
        return 0;
 err:
        return argc - 1;
}
~ % gcc -fprofile-arcs -ftest-coverage bug.c -o bug
~ % ./bug 1; echo $?
1
~ % ./bug; echo $?
0
~ %
```

Testing: coverage

```
~ % gcov bug &> /dev/null
~ % cat bug.c.gcov
        -:    0:Source:bug.c
        -:    0:Graph:bug.gcno
        -:    0:Data:bug.gcda
        -:    0:Runs:2
        -:    0:Programs:1
        -:    1:#include <stdio.h>
        -:    2:
        2:    3:int main(int argc, char *argv[])
        -:    4:{
        2:    5:        if (argc > 1)
        1:    6:                goto err;
        1:    7:        goto err;
        -:    8:        printf("%s\n", argv[0]);
        -:    9:        return 0;
        -:   10: err:
        2:   11:        return argc - 1;
        -:   12:}
~ %
```

One week after

Advisories

Tag	Other identifiers	Severity	Information
GNUTLS-SA-2014-2	CVE-2014-0092	Certificate verification issue	A vulnerability was discovered that affects the certificate verification functions of all gnutls versions. A specially crafted certificate could bypass certificate validation checks. The vulnerability was discovered during an audit of GnuTLS for Red Hat. **Who is affected by this attack?** • Anyone using certificate authentication in any version of GnuTLS. **How are past sessions affected?** • The vulnerability to be exploited it requires an active man-in-the-middle attacker. Past sessions are not affected unless they were under such an attack. **How to mitigate the attack?** • Upgrade to the latest GnuTLS version (3.2.12 or 3.1.22), or apply the patch for GnuTLS 2.12.x.

www.cern.ch