Exposure to “Lucky Thirteen” SSL Vulnerability

David Holmes, 2013-04-02

This week, two researchers, Kenny Paterson and Nadhem Alfardan, officially release a paper detailing a new set of timing attacks against SSL, which they call “Lucky Thirteen”. Unlike Rizzo & Duong’s SSL attacks (BEAST and CRIME), the Lucky Thirteen attacks don’t require execution of code in the browser. But, like BEAST and CRIME, they center on characteristics of block ciphers (such as AES). Actually these attacks aren’t ***really*** new; they just use sampling to exploit gaps of weakness that were previously thought to be too small to be measured. Before we get into the some of the nitty-gritty details, let me do an unofficial vendor statement. There is no public tool (yet) to test whether or not a particular SSL implementation is vulnerable to these attacks. So, here we are making some guesses as to the exposure for F5 products.

Lucky Thirteen - F5 Projected Threat Level - Low

1. In general, we think the data planes of F5 hardware appliances and blades are not vulnerable
2. The virtual editions are likely vulnerable. Follow the mitigation recommendations in the official statement.
3. The management ports are likely vulnerable. You should not have these hooked to the Internet anyway.

When the OpenSSL group issues a patch, we will integrate it and issue a hotfix for #2 and #3.

So why do we think the threat level is low? For TLS (not UDP DTLS), actual applicability of the vulnerability in the real world may be low, as admitted by the authors themselves.

> “Given its complexity, the full plaintext recovery attack on TLS should, for now, be considered more of a theoretical threat.”

The vast majority of SSL traffic passed by BIG-IPs is protected by hardware acceleration. We had some back and forth with the authors about whether or not our cryptographic offload would protect us. They were uncertain but we both agreed that until we actually test it there’s no way to know for sure.

> “We also expect that all implementations will be vulnerable to simple variants of our attacks, unless the implementers have taken great care to ensure that the decryption processing time is uniform, or nearly so. Our experiences in investigating open-source implementations suggests this is unlikely.”

So we at F5 think that cryptographic offload provides some measure of protection because it processes in near-constant time. Another factor is the asynchronous nature of that offload. Our traffic management microkernel (tmm) pushes the SSL records to the accelerator, then goes and does something else. After a time, it schedules a visit back to the accelerator to pick up the result. As one of our architects said, after reading the paper, “scheduling jitter is larger than operation time by over an order of magnitude.” Paterson and Alfardan think that with enough sampling they still might be able to tease out information, but I’m not convinced yet.

At risk of getting this wrong (after all, I just read the paper and associated sources patches, I didn’t write them), here is my understanding of these Lucky Thirteen vulnerabilities.

At a cryptographic level, the nature of the problem is that the most TLS processing code behaves slightly differently when decrypting different blocks of ‘bad’ ciphertext. This difference in behavior is tiny, but through sampling and noise reduction, and stars aligning (or at least MAC byte boundaries and block sizes aligning), one can in theory break a TLS session and recover the plaintext if enough messages are seen. The “datagram” version of TLS (called DTLS) is more vulnerable because data can be more readily tampered with in transit.

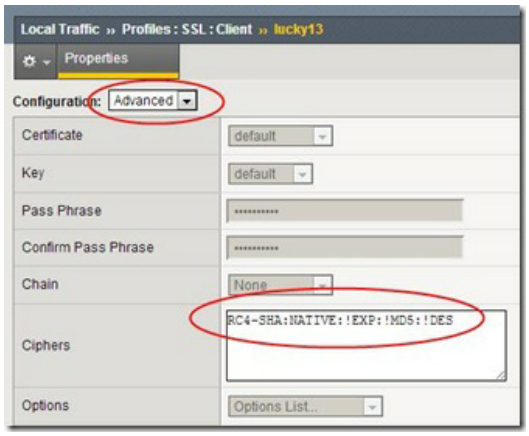
As time passes, we’ll be able to get a sense of what the crypto community thinks of these attacks. I know that Adam Langley over at Google is taking these very seriously, but the Google servers rely on non-constant time software SSL stacks, against which Paterson and Alfardan have demonstrated their exploit.

If I were to be bold and make some predictions, I'd say that "Lucky Thirteen" will be placed next to BEAST and CRIME in the cryptographic museum of exploits. They get lots of attention, but few (if any) actual attacks are seen in the wild. As BEAST and CRIME are speeding the adoption of TLS 1.2, maybe "Lucky Thirteen" will speed the adoption of the AES-GCM ciphers (which aren't vulnerable).

All of this might seem messy, but this is the way that security evolves when it is done properly.

Recommendation:

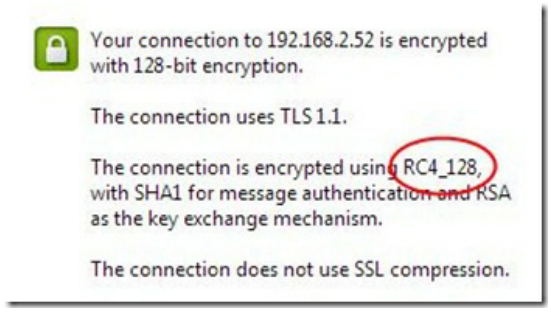
For any virtual server that is not hardware-accelerated (or if you are unsure), consider switching your TLS ciphers to prefer RC4.



There are probably better cipher strings out there, but I tested this one on Chrome and IE and it looks to give the right result.

> **RC4-SHA:NATIVE:!EXP:!DES:!MD5**

MD5 has nothing to do with this, but it is force of habit for me to exclude it.



Connect with David: Connect with F5:

Related blogs & articles:

Lucky Thirteen Vulnerability Paper
SSL News Roundup – BEAST, CRIME and Pulse
Getting Good Grades on your SSL
SSL: On the Failure of False Start
A Different Approach to Cross-VM Side Channel Exploitation

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com