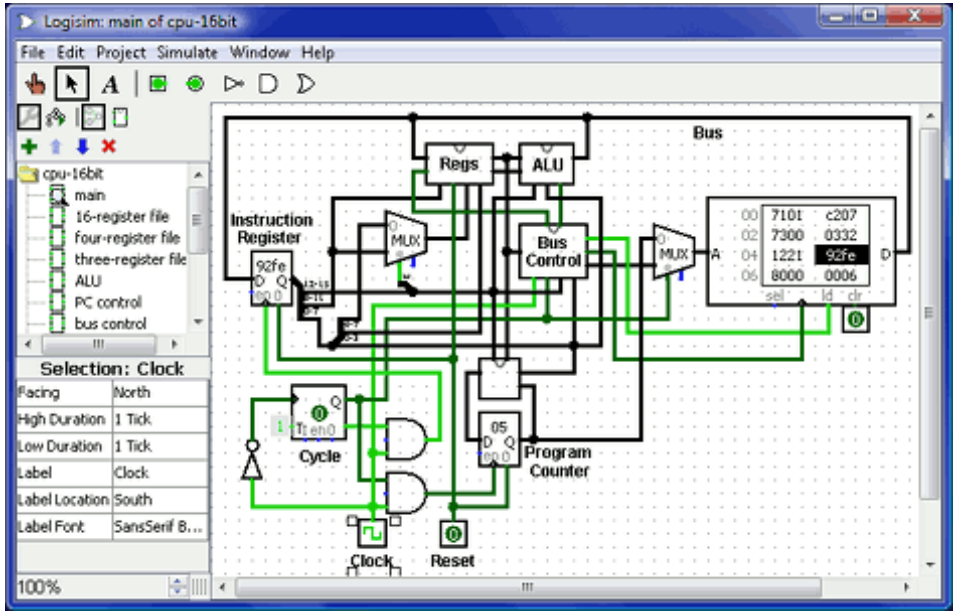
The Matter of Heartbleed

Authors: Durumeric, Kasten, Adrian, Halderman, Bailey, Li, Weaver, Amann, Beekman, Payer, Paxson

Presenter: Ben

Miss your CS31 Days?

- Be a CS31 Grader!
- Talk to Kevin



About the Authors

- University of Michigan
  - Ph.D. Students: Zakir Durumeric, James Kasten, David Adrian
  - Advisers: Alex Halderman, Michael Bailey
- Berkley
  - Ph.D. Students: Frank Li, Mathias Payer
  - Professors: Nicholas Weaver, Vern Paxson
- International Computer Science Institute
  - "ICSI is a leading independent, nonprofit center for research in computer science."

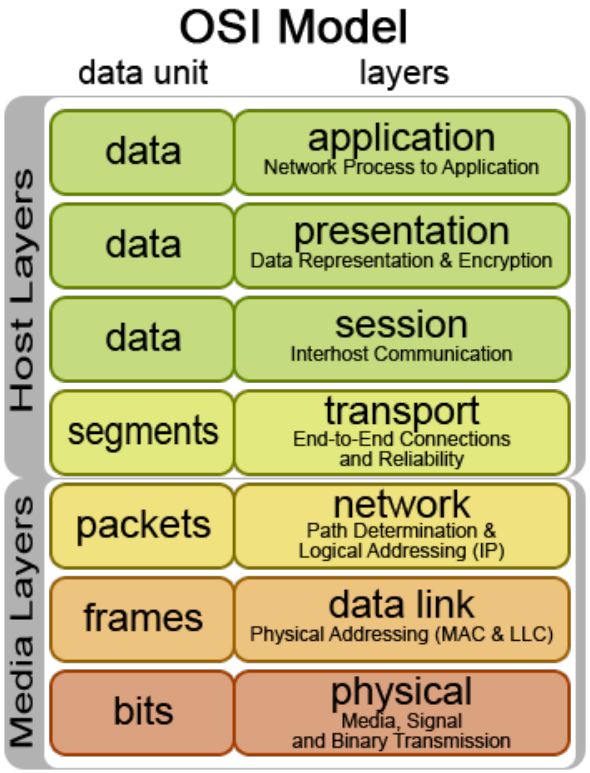
About the Conference -- IMC

- Papers … contribute to the current understanding of how to collect or analyze Internet measurements, or give insight into how the Internet behaves.
  - Response to difficulty … finding appropriate publication/ presentation venues for high-quality Internet measurement research
  - Frustration with the … treatment of measurement submissions
- "Matter of Heartbleed" won best paper

Goal: Characterize Heartbleed

- Who was vulnerable?
- Were fixes applied? How?
- How does notification help?
- Broader Community Takeaways.

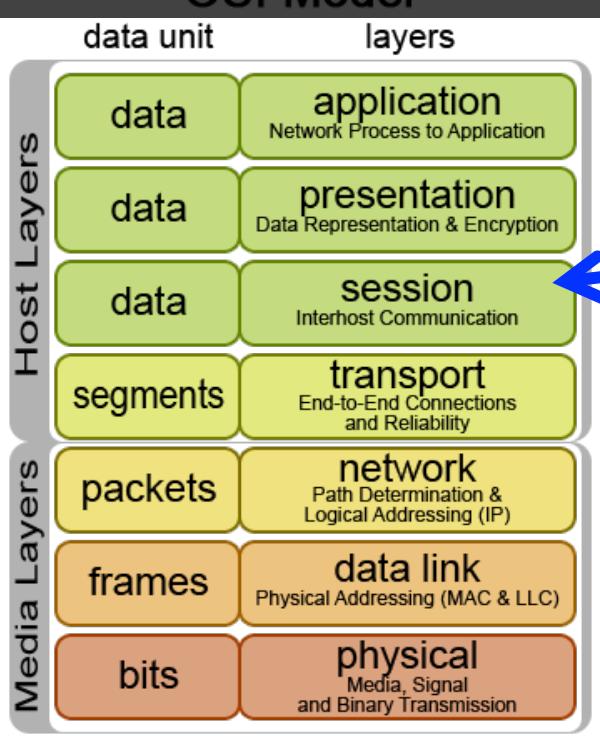
What's SSL / TLS / OpenSSL?



The internet does not take care of securing your data for you!

Responsibility of the end host

What's OpenSSL?



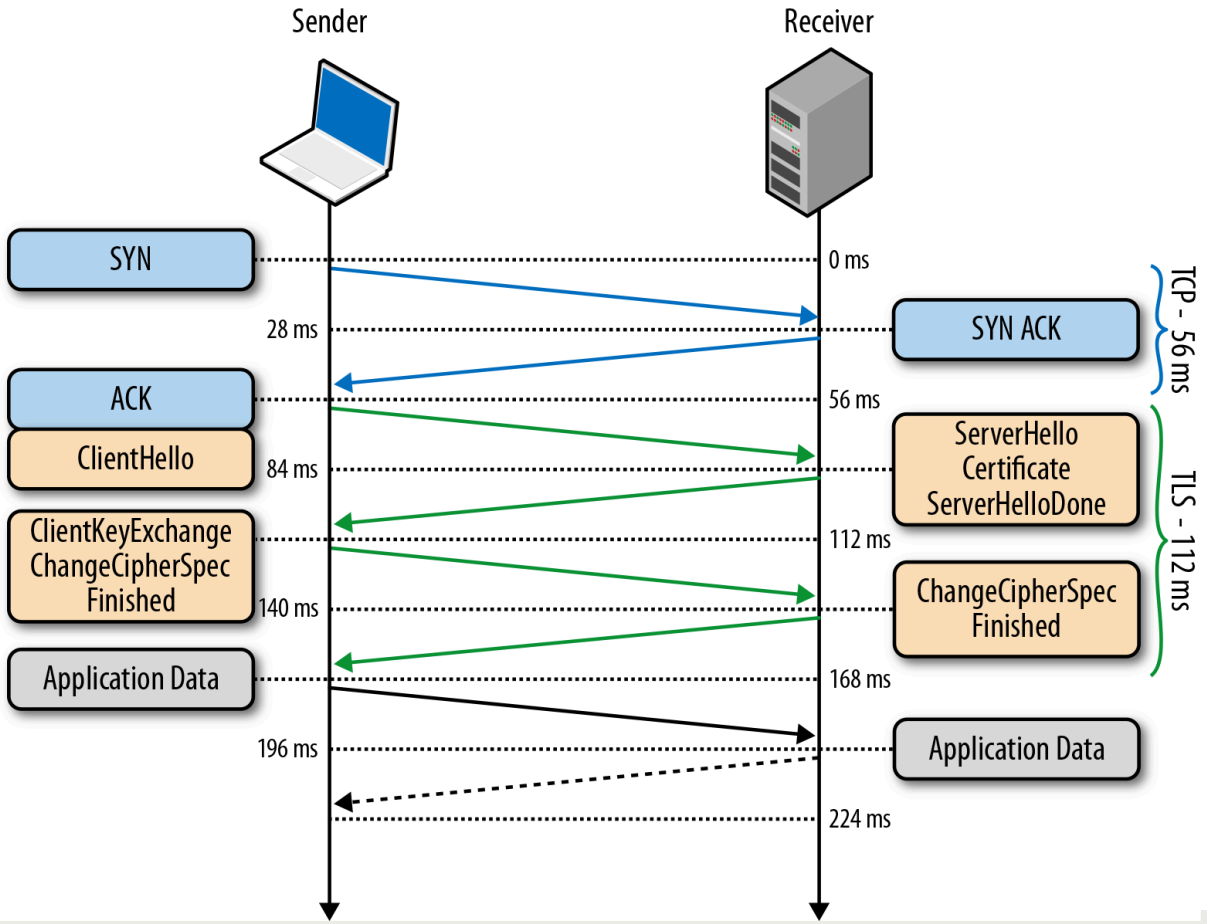
What's TLS?

Transport Layer Security

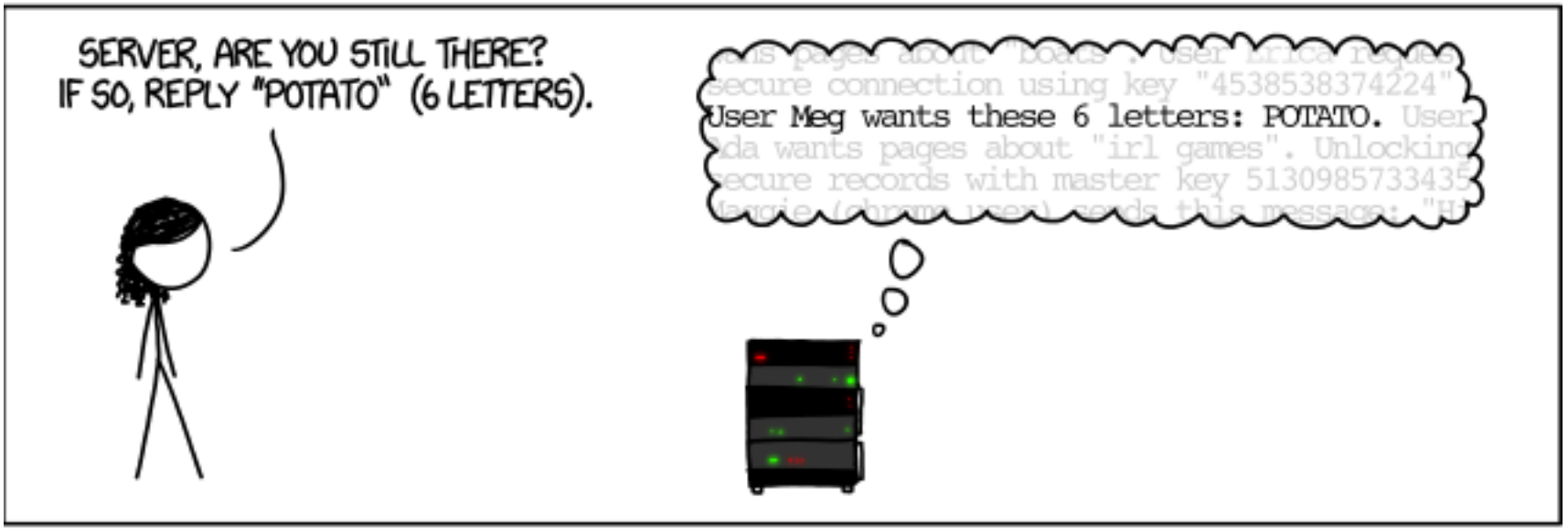
From Wikipedia, the free encyclopedia

Transport Layer Security (**TLS**) and its predecessor, **Secure Sockets Layer** (**SSL**), both of which are frequently referred to as 'SSL', are cryptographic protocols designed to provide communications security over a computer network.[1] Several versions of the protocols are in widespread use in applications such as web browsing, email, Internet faxing, instant messaging, and voice-over-IP (VoIP). Major web sites (including Google, YouTube, Facebook and many others) use TLS to secure all communications between their servers and web browsers.

TLS Connection Setup

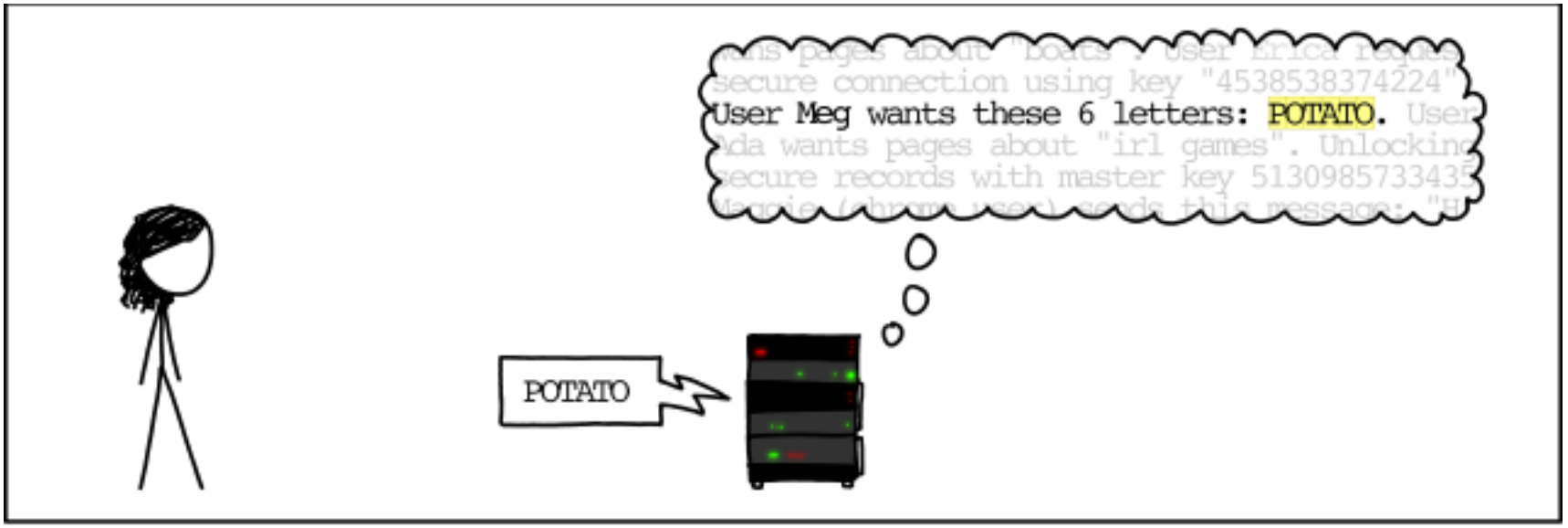


Aside: What is Heartbleed?

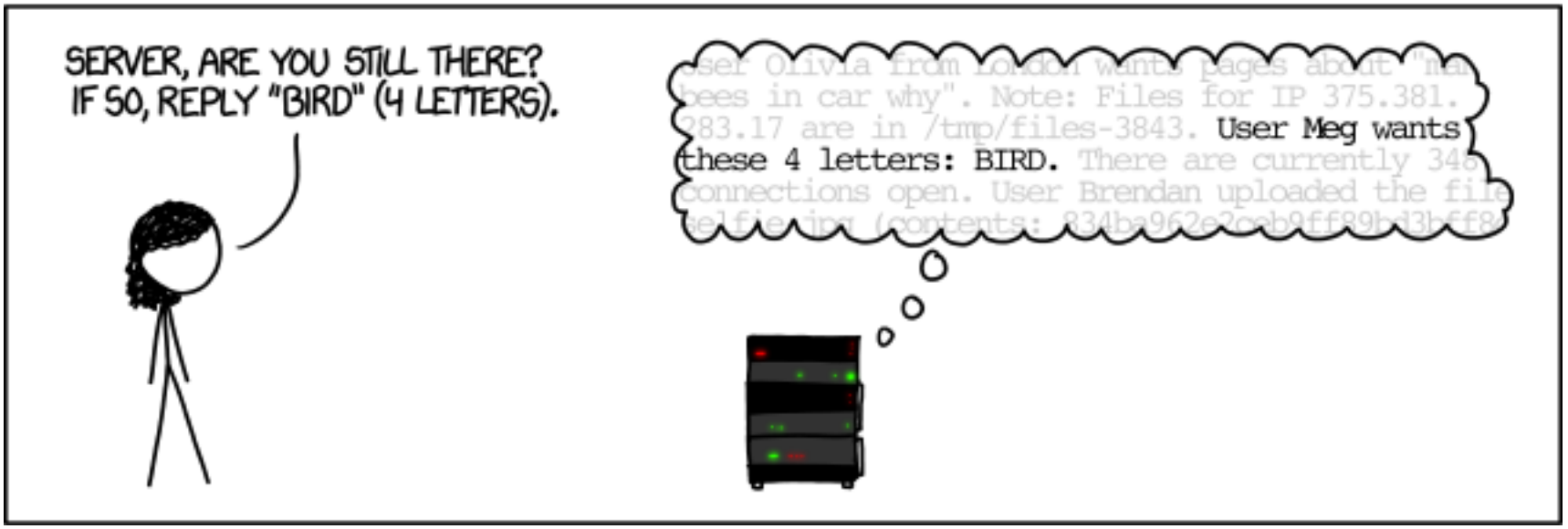


http://xkcd.com/1354/

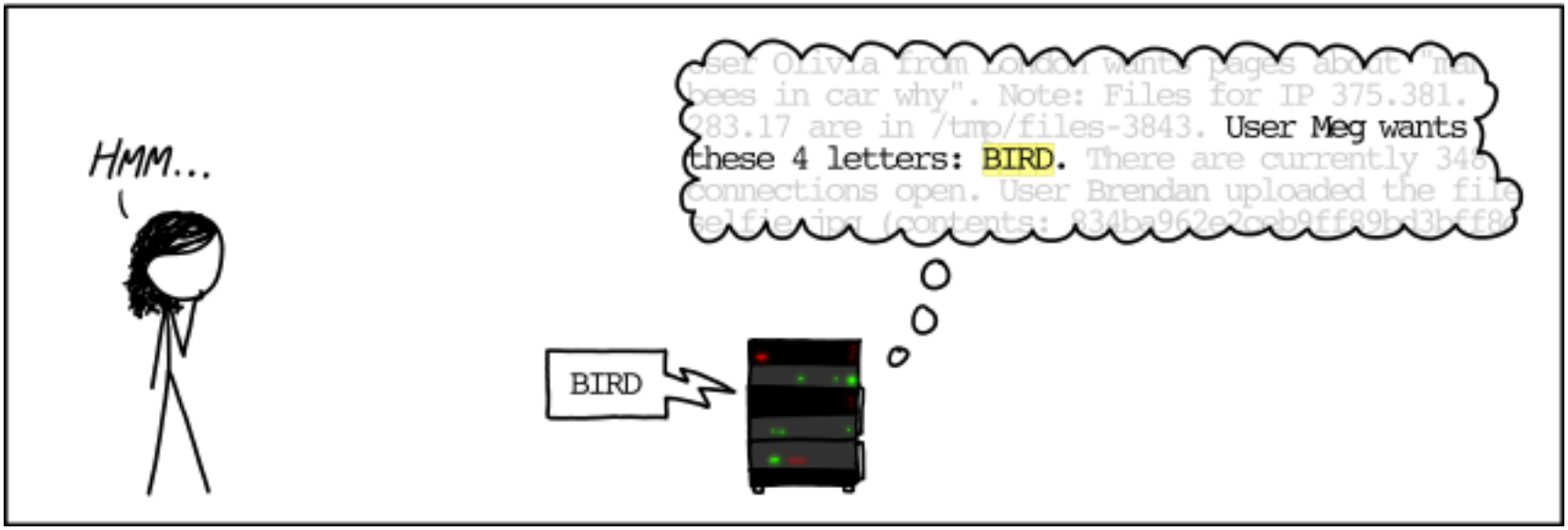
Aside: What is Heartbleed?



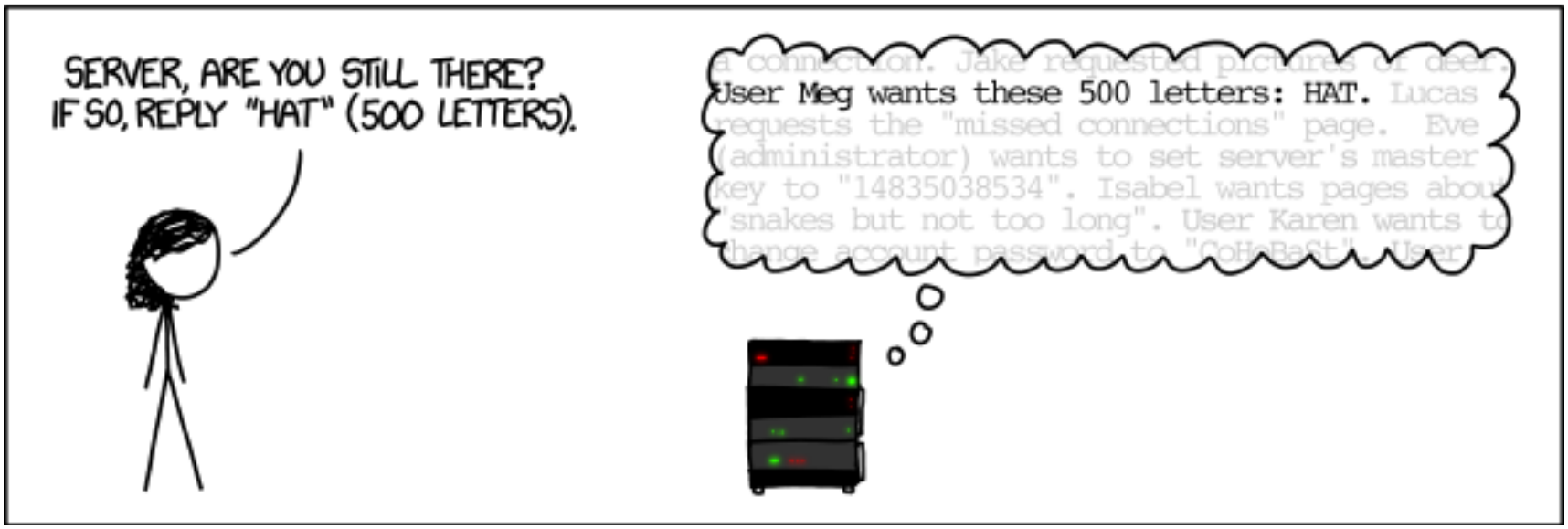
Aside: What is Heartbleed?



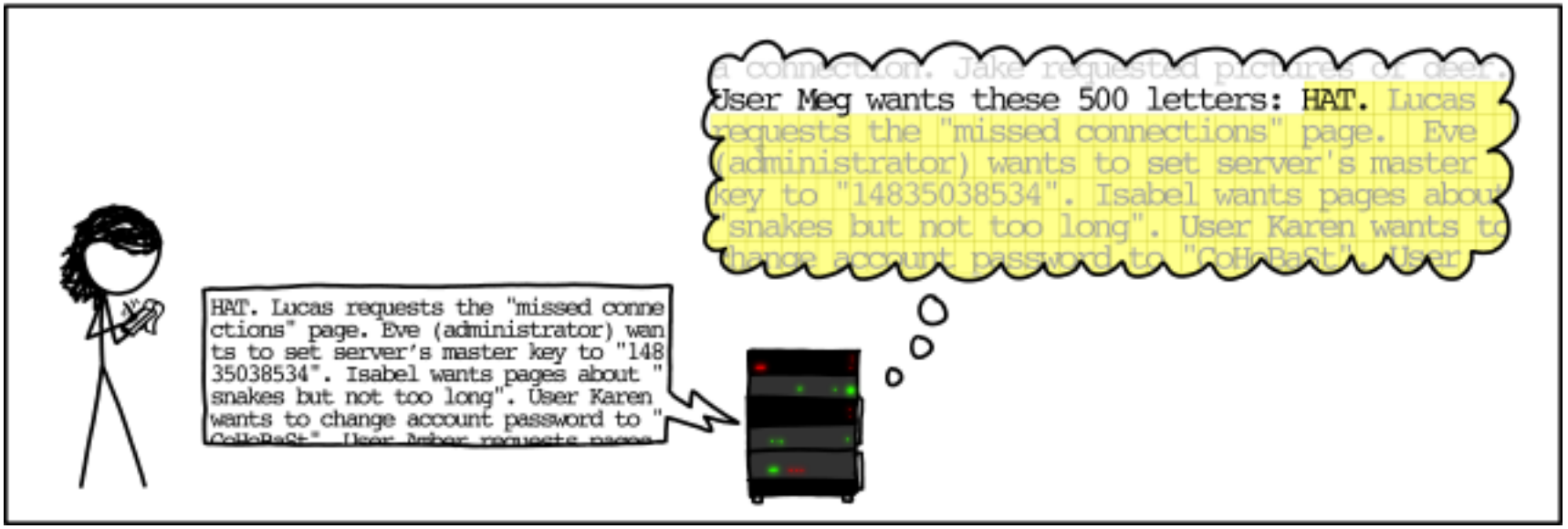
Aside: What is Heartbleed?



Aside: What is Heartbleed?



Aside: What is Heartbleed?



The Vulnerability

- "The peer trusts the attacker-specified length of an attacker-controlled message."
  - Can read private memory on server

The Vulnerability

- "The peer trusts the attacker-specified length of an attacker-controlled message."
  - Can read private memory on server
- Pervasive Bug
  - OpenSSL / TLS used in a lot of places.

How Big of a Deal Was It?

Don't Panic About Recent Zero-Day Linux Kernel Vulnerability, It's Not That Bad

People have overreacted to the latest vulnerability

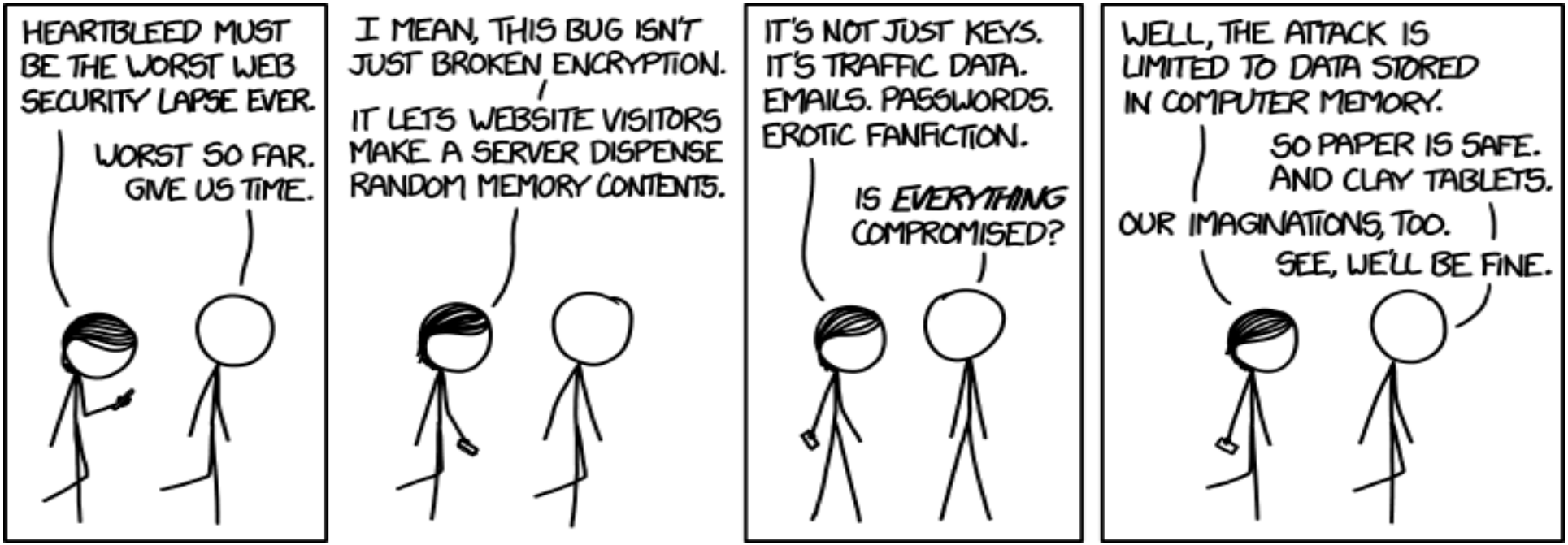
Jan 20, 2016 13:33 GMT · By Silviu Stahie

The kernel vulnerability that was revealed only yesterday got some users panicked, but the truth is that's not really the case.

It's easy to scare users with warnings of vulnerabilities in the Linux kernel, which powers pretty much everything. The vulnerability was aptly named CVE-2016-0728, which is not all that impressive. This is the first indication. When something is bad enough, it usually receives a name, like Heartbleed for OpenSSL. When you're called CVE-2016-0728, it's hard to become a serious threat. This is a joke, of course, and people should be always aware if vulnerabilities, even if they are not called Linus' Doom.

http://news.softpedia.com/news/don-t-panic-about-recent-zero-day-linux-kernel-vulnerability-it-s-not-that-bad-499166.shtml

How Big of a Deal Was It?



http://imgs.xkcd.com/comics/heartbleed.png

Who Was Vulnerable?

- Anybody that used OpenSSL to facilitate TLS connections.
  - Apache HTTPS, Zimbra, various mail and DB servers
- Estimate: 24 – 55% of popular HTTPS hosts impacted initially

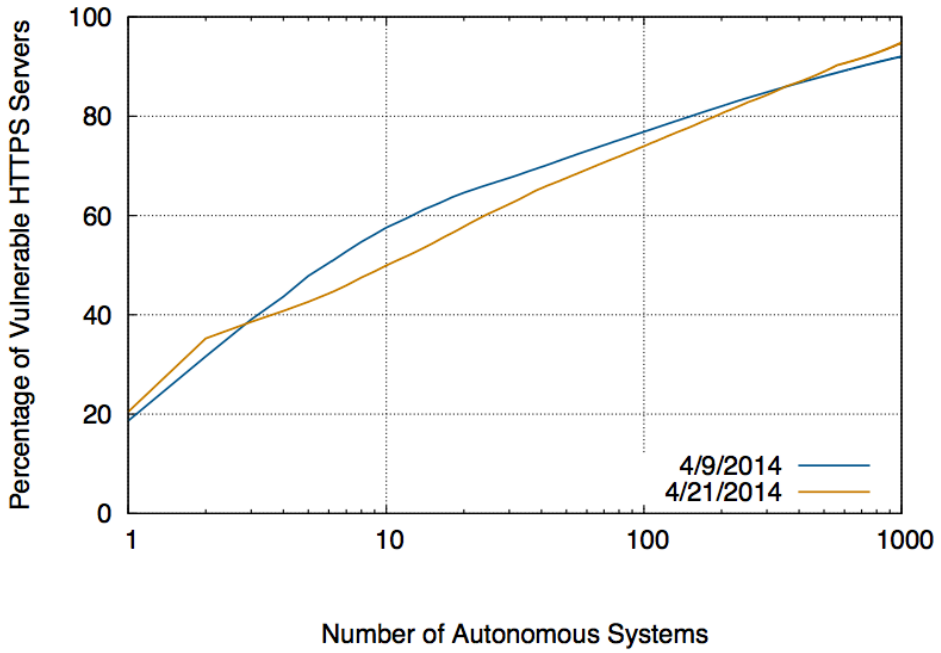
Some Commentary

- The Good
  - Nice graphics clearly illustrate desired points.
  - Reasonably straightforward description of conclusions.

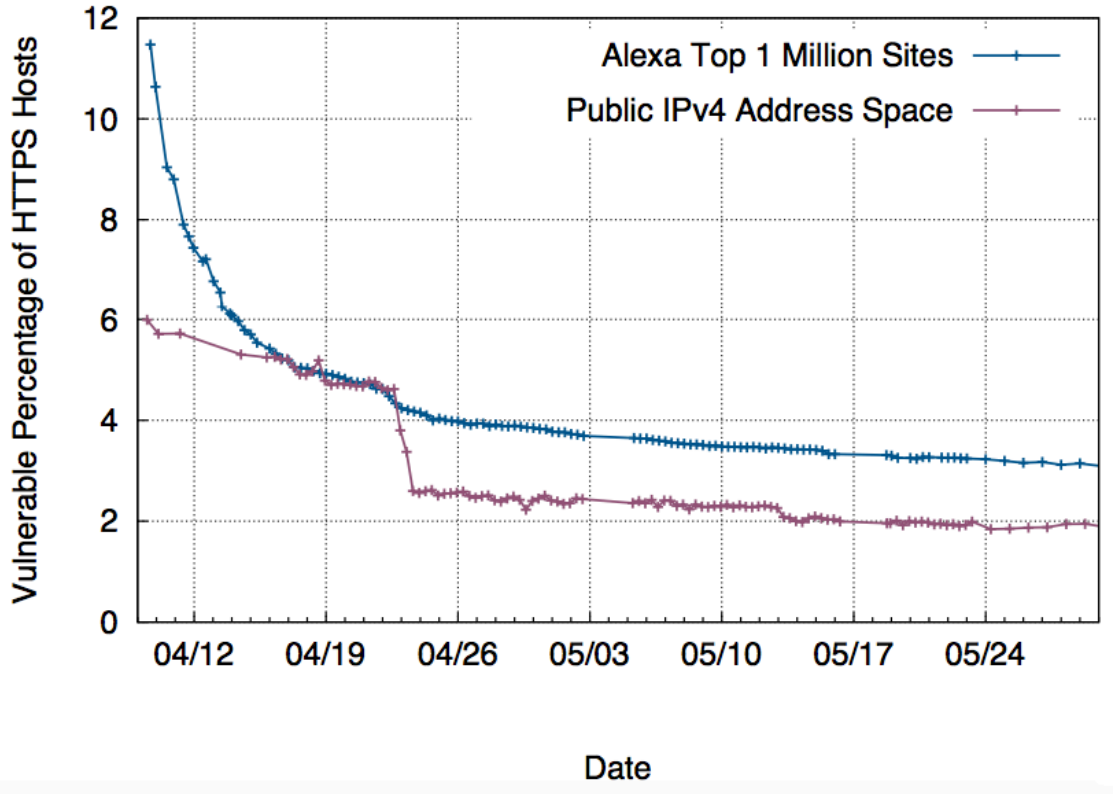
- The Less Good
  - Clearly collected lots of data and showed it.
  - Calls for more discussions, with few explicit solution ideas.

Vulnerability Not Evenly Distributed

- A small number of ASes had a large proportion of vulnerable sites.



Patching


Alexa Top 1 Million Sites
Public IPv4 Address Space
Vulnerable Percentage of HTTPS Hosts
12
10
8
6
4
2
0
04/12
04/19
04/26
05/03
05/10
05/17
05/24
Date

Patching Comparison

- Rate of Patching

  Heartbleed > Debian Keys (publicity?)

- Rate of Certificate Replacement

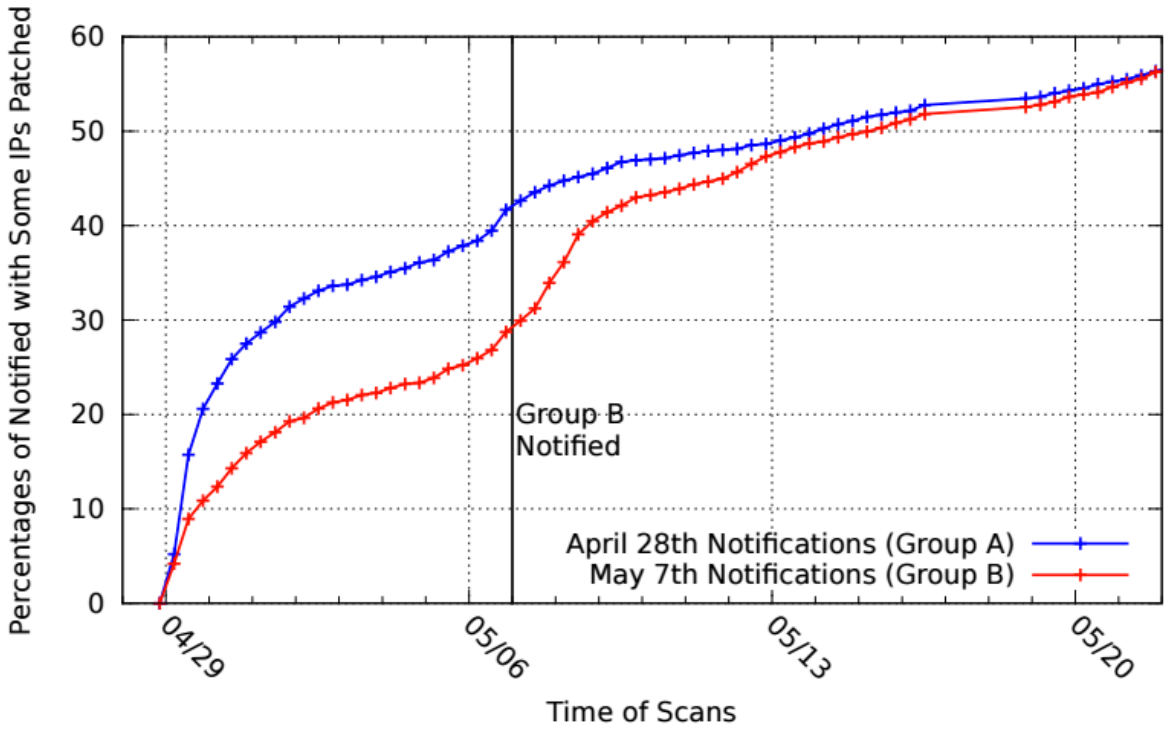
  Debian Keys > Heartbleed (necessity?)

- # Patching > # Replace Certs > # Revoked Old Certs

Attack Scene

- Malicious scans start 22h after disclosure.
  - Targeted attacks likely happened earlier.
  - Not too much internet-wide scanning.
- Amazon host got lots of attacks
  - Preference for scanning dense address spaces?

Notification Spurs Patching



Spurring Patching

- Notification → Patch Application
- Notifications positively received.
- Why Vulnerabilities Persist
  - Sysadmins didn't control servers
  - Missed on internal scans

Community Takeaways

- Maintaining secure systems is hard:
  - "Many server administrators have only a superficial understanding … or failed to understand the consequences"
- Who gets to know first?
  - "Patching was delayed because the Heartbleed disclosure process unfolded in a hasty and poorly coordinated fashion. … The security community needs to be better prepared for mass vulnerability disclosure."
- Patch application happens slowly
  - Explicit notification helps – can this be automated or improved?
- Need better certificate revocation infrastructure.

Further Reading

Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed

Liang Zhang
Northeastern University
liang@ccs.neu.edu

David Choffnes
Northeastern University
choffnes@ccs.neu.edu

Dave Levin
University of Maryland
dml@cs.umd.edu

Tudor Dumitraş
University of Maryland
tdumitra@umiacs.umd.edu

Alan Mislove
Northeastern University
amislove@ccs.neu.edu

Aaron Schulman
Stanford University
aschulm@stanford.edu

Christo Wilson
Northeastern University
cbw@ccs.neu.edu

ABSTRACT

Central to the secure operation of a public key infrastructure (PKI) is the ability to *revoke* certificates. While much of users' security rests on this process taking place quickly, in practice, revocation typically requires a human to decide to reissue a new certificate and revoke the old one. Thus, having a proper understanding of how often systems administrators reissue and revoke certificates is crucial to understanding the integrity of a PKI. Unfortunately, this is typically difficult to measure: while it is relatively easy to determine when a certificate is revoked, it is difficult to determine whether and when an administrator *should have* revoked.

In this paper, we use a recent widespread security vul-

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.3 [**Computer-Communication Networks**]: Network Operations; E.3 [**Data Encryption**]: Public Key Cryptosystems, Standards

Keywords

Heartbleed; SSL; TLS; HTTPS; X.509; Certificates; Reissue; Revocation; Extended validation

1. INTRODUCTION

Further Reading

Surreptitiously Weakening Cryptographic Systems

Bruce Schneier[1] Matthew Fredrikson[2] Tadayoshi Kohno[3] Thomas Ristenpart[2]

[1] *Co3 Systems* [2] *University of Wisconsin* [3] *University of Washington*

February 9, 2015

Abstract

Revelations over the past couple of years highlight the importance of understanding malicious and surreptitious weakening of cryptographic systems. We provide an overview of this domain, using a number of historical examples to drive development of a weaknesses taxonomy. This allows comparing different approaches to sabotage. We categorize a broader set of potential avenues for weakening systems using this taxonomy, and discuss what future research is needed to provide sabotage-resilient cryptography.

1 Introduction

Cryptography is critical to modern information security. While it is undeniably difficult to design and implement secure cryptographic systems, the underlying mathematics of cryptography gives the defender an

Further Reading

IJISC - International Journal of Information Security and Cybercrime *Vol. 3, Issue 2/2014*

Heartbleed - The Vulnerability That Changed the Internet

Ionuț-Daniel BARBU, Ioan BACIVAROV
EUROQUALROM, University POLITEHNICA of Bucharest, Romania
barbu.ionutdaniel@gmail.com, bacivaro@euroqual.pub.ro

Abstract

This article is intended to present the Heartbleed bug and will include information from statistical aspects and impact of the vulnerability to an overview of how it actually works. In addition to this, a reproduction of the exploit is described and some affected software distributions listed. For educational purposes, during this research a vulnerable version of Apache server has been targeted. The well - known, low cost device RaspberryPI built on ARM architecture serves as the hardware platform for the targeted machine and it supports a Linux image. Heartbleed vulnerability has been categorized as a critical vulnerability of the cryptographic software library OpenSSL and its name has proven to be a good choice from various perspectives. 14th of March 2012 is the day when the bug has been released after its introduction in the code. Although the discovery's precise time is questionable by a lot of critics, at least the public disclosure date is known and that is 1st of April

Further Reading

Risk Assessment of Buffer "Heartbleed" Over-read Vulnerabilities

Jun Wang, Mingyi Zhao, Qiang Zeng†, Dinghao Wu, Peng Liu
The Pennsylvania State University, University Park, PA 16802, USA
{jow5222, muz127, dwu, pliu}@ist.psu.edu, †quz105@cse.psu.edu

***Abstract*—Buffer over-read vulnerabilities (*e.g.*, Heartbleed) can lead to serious information leakage and monetary lost. Most of previous approaches focus on buffer overflow (*i.e.*, over-write), which are either infeasible (*e.g.*, canary) or impractical (*e.g.*, bounds checking) in dealing with over-read vulnerabilities. As an emerging type of vulnerability, people need in-depth understanding of buffer over-read: the vulnerability, the security risk and the defense methods.**

This paper presents a systematic methodology to evaluate the potential risks of *unknown* buffer over-read vulnerabilities. Specifically, we model the buffer over-read vulnerabilities and focus on the quantification of how much information can be potentially leaked. We perform risk assessment using the RUBiS benchmark which is an auction site prototype modeled after eBay.com. We evaluate the effectiveness and performance of

adopted due to either excessive runtime overhead, expensive cost of manual work, or insufficient mitigation in practice.

A number of programming techniques for writing solid and secure code have been around to preserve memory safety. Although some of them are initially intended to make the resulted software more reliable, they are helpful in mitigating the buffer over-read vulnerability. Zero out memory blocks or buffers have been advocated and implemented in many systems. In particular, Chow et al. [12] measured the performance overhead on zero out heap buffers and stack frames at deallocation time. They also experimented on zeroing out currently unused stack part periodically. Initializing and padding buffers with special characters are proposed to facilitate easier debugging

Questions to Ponder

- How can patching rates be improved?
- Is there a way to scale in-person notifications?
- What is the division of responsibility between a service provider and their clients for ensuring secure systems?
- Who gets to know first? Who determines?
- Heartbleed was widely broadcast, both in technical and non-technical mediums. Does added awareness improve or degrade security?