

Methods of

“The Matter of Heartbleed”

Stefan Savage
University of California, San Diego

Themes

Ethical Considerations for Vulnerability Disclosure

Where research identifies a vulnerability (e.g., software vulnerabilities in a given program, design weaknesses in a hardware system, or any other kind of vulnerability in deployed systems), we expect that researchers act in a way that avoids gratuitous harm to affected users and, where possible, affirmatively protects those users. In nearly every case, disclosing the vulnerability to vendors of affected systems, and other stakeholders, will help protect users. It is the committee's sense that a disclosure window of 45 days

<https://vuls.cert.org/confluence/display/Wiki/Vulnerability+Disclosure+Policy> to 90 days <https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-faq.html> ahead of publication is consistent with authors' ethical obligations.

Longer disclosure windows (which may keep vulnerabilities from the public for extended periods of time) should only be considered in exceptional situations, e.g., if the affected parties have provided convincing evidence the vulnerabilities were previously unknown and the full rollout of mitigations requires additional time. The authors are encouraged to consult with the PC chairs in case of questions or concerns.

The version of the paper submitted for review must discuss in detail the steps the authors have taken or plan to take to address these vulnerabilities; but, consistent with the timelines above, the authors do not have to disclose vulnerabilities ahead of submission. If a paper raises significant ethical and/or legal concerns, it will be checked by the REC and it might be rejected based on these concerns. The PC chairs will be happy to consult with authors about how this policy applies to their submissions.

Example of a “run fast” paper

Time is of the essence

- Something happened and need to act quickly
- Top 100 sites already patched before they started scanning

Competition

- A highly-visible event → you're not alone
- Al-Bassam Top 10K scan, 2nd IMC 2014 paper (deep dive on certificates)

Cooperation

- Author list has a story
- Groups combine data sets, expertise

Brief history of “run fast” vulnerability measurement

2001: Code red (passive monitoring of hosts patching)

2001: Provos + Honeyman ScanSSH (active scans of ssh servers)

2003: Slammer (passive monitoring of networks firewalled)

2003: Rescorla OpenSSL buffer overflow (active scanning, Slapper)

2009: Yilek et al. Debian weak keys (active scans of 751 Web certs)

2012: Heninger et al. TLS weak keys (primarily embedded devices)

2014: Heartbleed

2021: Log4j ...

Quick reminder: Secure Sockets Layer/Transport Layer Security

SSL released 1995/6 by Netscape

- Many security/feature issues

TLS subsumed SSL, 1.0 released 1999

- IETF standard, currently at version 1.3 (2018)

Provides confidentiality (encryption) and integrity

- Communication is private, data not modified in transit

Authenticates server to client using certificates

- Site sends certificate, client authenticates site controls hostnames in certificate

Strongly associated with the Web (HTTPS)

- Also used by many other services (email, Tor, database, IM, etc.) also using TCP
- OpenSSL one of the major libraries implementing TLS

Datagram TLS in OpenSSL

TLS normally uses a reliable transport protocol (TCP, QUIC)

Datagram TLS designed for unreliable transport protocols (e.g., UDP)

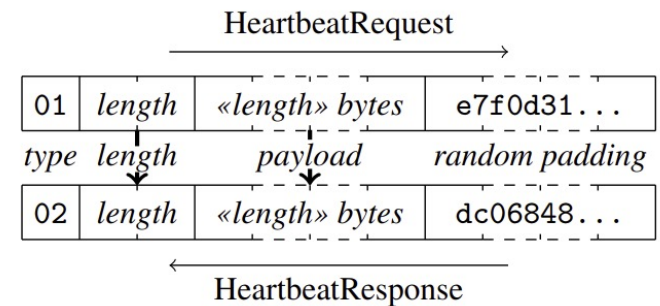
DTLS Heartbeat Extension helps with session management

- Either endpoint can ask if other side is still present
- Not required for TLS since functionality provided by TCP/QUIC

Heartbeat Extension was compiled into OpenSSL library and active

- Even if a server only wants TLS for HTTPS (DTLS not needed)

Heartbleed vulnerability



Client → server: Here is a payload, echo it back to me

– Server → client: Here you go

Client controls payload length

- OpenSSL implementation **did not check length**
- Client can provide a length much greater than size of packet
- Server responds by **reading 64k of random data** from address space

Severe implications

- Trivial to exploit
- Wide-spread vulnerability
- Attacker can potentially get private key, login credentials, other private data

Timeline and remediation

Discovered by Mehta at Google, Google patches

- Others privately notified, including OpenSSL

Codenomicon independently discovers, others notified incl. OpenSSL

- Aside: independent discovery one day apart; how might that happen?

OpenSSL quickly releases new version with fix

Same day CloudFlare and Codenomicon announce on Twitter

- Big news, widely-covered in media, topic of the day, everyone talking about it

Remediation

- Sites can patch by updating to latest OpenSSL
- Sites also need to create new certificates and revoke old ones
 - Potentially compromised private key undermines old certificate

What were the goals of the paper?

Identify and track vulnerable Web sites

Monitor remediation behavior

- Software patching
- Certificate updates

Assess impact on certificate ecosystem

Notify vulnerable sites at scale to encourage remediation
(and see if this mattered)

What was their scanning methodology?

Already scan HTTPS servers on the Internet

- Key question is identifying **vulnerable** servers

How do they do vulnerability scans and navigate doing no harm?

- Send “null” Heartbeat messages: no payload or padding, payload length is 0
- Correct implementations discard such messages
- OpenSSL w/ vulnerability will reply with random padding, nothing else

Daily scans of Alexa Top 1M

Quick aside: Alex Top 1M

Why focus on “top” 1M? What is purpose?

Why do you think this particular list (i.e., Alexa’s) is used?

- Norms

How do you think this list is generated?

- Alexa toolbar in browsers (note, not Amazon Alex, Alexa Internet)? Bias?

Subsequently, Alexa stops publishing list

- Entire community moves to Tranco list

Combines/averages other ranking lists (Cloudflare Radar, Farsigh, Majestic, Cisco)

- Chrome User eXperience list (CRuX) (from Chroma telemetry): IMC 22

Shows a) significant bias in other lists, b) that there is significant geographic variation in “topness”

What was their scanning methodology?

Already scan HTTPS servers on the Internet

- Key question is identifying **vulnerable** servers

How do they do vulnerability scans and navigate doing no harm?

- Send “null” Heartbeat messages: no payload or padding, payload length is 0
- Correct implementations discard such messages
- OpenSSL w/ vulnerability will reply with random padding, nothing else

Daily scans of Alexa Top 1M

Daily scans of random 1% of rest of IPv4

- Different set each day; pros/cons of not using the same 1%?

Start scans two days after public disclosure

Legal and ethical issues for scanning

Why might there be legal and ethical concerns?

- Active measurement of a known vulnerability
- Note reports of even o-byte payloads crashing printers

They have ZMap, why do they only sample 1% of IPv4 / day?

- They still do daily full scans for HTTPS

Are they at much practical legal risk?

What steps help their case?

- o-byte payload, distinguish from other scanners
- Clear identification (reverse DNS, web page)
- Notified operators of vulnerable hosts (overall intent is clear)
- Vulnerability already widely publicized

Aside: Happens to everyone

Essentially one number in each scan

- Web servers vulnerable to Heartbleed

How hard could this be?

- Always more complicated in practice
- Paper spends one page dealing with one design problem (handling timeouts)
- Leads to false negatives (missing vulnerable servers)
- Reasonable approach for estimating upper/lower bound impact on results

Happens to everyone

- But in other studies can have the luxury of re-running experiment
- If not, then deal with it similarly to this paper...awkward, but no choice

Lesson: **Record every kind of potential failure separately**

What patching behavior did they find?

Pre-disclosure

- ICSI Notary project passive monitoring of TLS connections
 - Use multiple data sources
- Google 12 days prior, Akamai 4 days prior

Two days after disclosure (time of their first scan)

- Top 100: All patched (popular sites patched quickly)
- Top 1M: 11% of HTTPS sites vulnerable (estimated 24-55% at disclosure)

Patch rates good for next two weeks ... but levels off

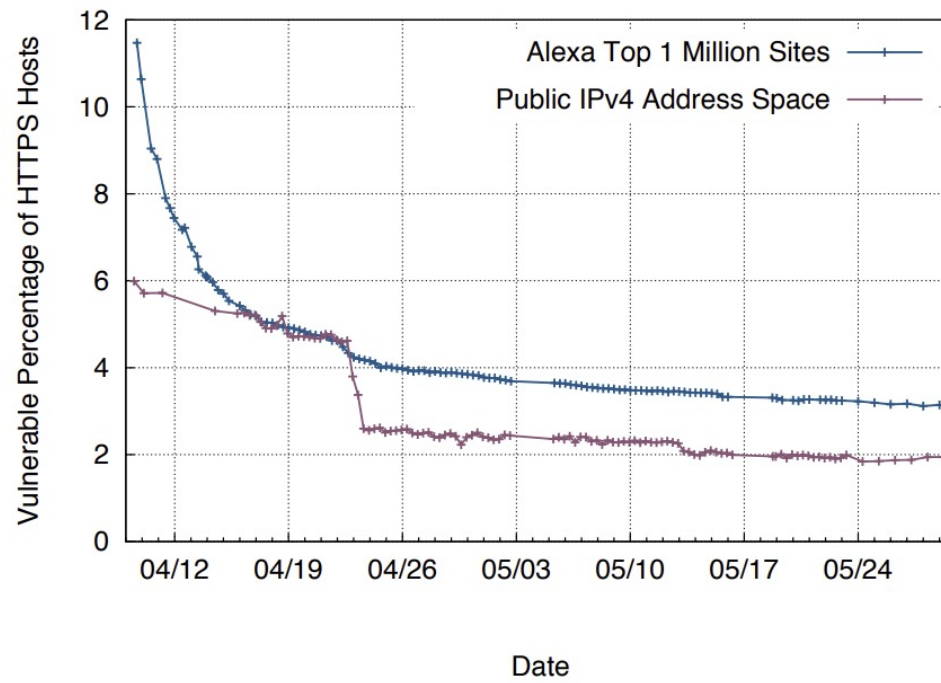


Figure 3: **HTTPS Patch Rate.** We track vulnerable web servers in the Alexa Top 1 Million and the public IPv4 address space. We track the latter by scanning independent 1% samples of the public IPv4 address space every 8 hours. Between April 9 and June 4, the vulnerable population of the Alexa Top 1 Million shrank from 11.5% to 3.1%, and for all HTTPS hosts from 6.0% to 1.9%.

Certificates

Patching is not the only remediation needed

- Need to create new certificate with new private key
- Need to revoke old certificate
- Additional twist to a patching study

New certificates

- Vulnerable sites on day 2: Only 10% replace certs in 30 days (but 73% patched)
- 14% re-used old private key (new cert useless)

Certificate revocation

- Only 19% that replaced also revoked old certificate
- Still, resulted in large bulk revocations
- Costly + over-burdened revocation ecosystem downloading large CRLs
- Revocation ecosystem still in flux today (e.g., Chrome no longer does online checks)

Attacks

Various network monitors + honeypot give visibility of other actors

- Widely publicized → not the only one scanning (cf. Storm)
- Full protocol parsing to ensure observing Heartbleed traffic; actually get data

Pre-disclosure

- Claims no scanning activity: vulnerability not known “in the wild”

Post-disclosure

- Scans start immediately after disclosure
- Many sources, although few exhaustive scans
- Dominated by scan services (e.g., ssllabs.com)

Using it for your IPs vs other IPs

Should these be labeled attacks?

- Does intent matter?

Aside: “Is finding security holes a good idea?”

In 2005, Rescorla published a provocative paper

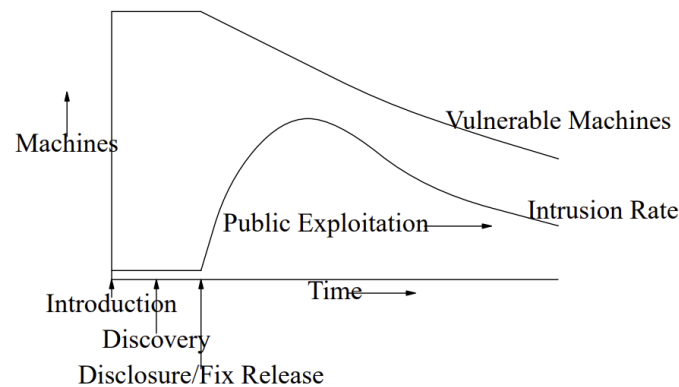


Figure 1 White Hat Discovery process when disclosure and fix release occur together

Does vulnerability finding increase overall social welfare?

- In paper’s framing, answer may be no: depends heavily on “rediscovery probability”
- If rediscovery probability is not high, then publishing fix decreases overall welfare

→ Highlights effects of announcing vulnerability to attackers

- What are other considerations for finding vulnerabilities?

What was the notification methodology?

Set of 212k vulnerable hosts from full IPv4 scan

Identify contacts taken from WHOIS abuse field

- 4.6k contacts: A lot of notifications
- Raises the question of who is most effective to contact

Separate into two random groups

- Group A: Notify immediately (3 weeks after vuln announced)
- Group B: Notify 9 days after Group A
- Why have two groups? Ethical considerations? Mitigations?

Scan vulnerable hosts every 8 hours

Significant effect: 47% increase in patching between A and start of B

- Human responders are generally positive to notifications

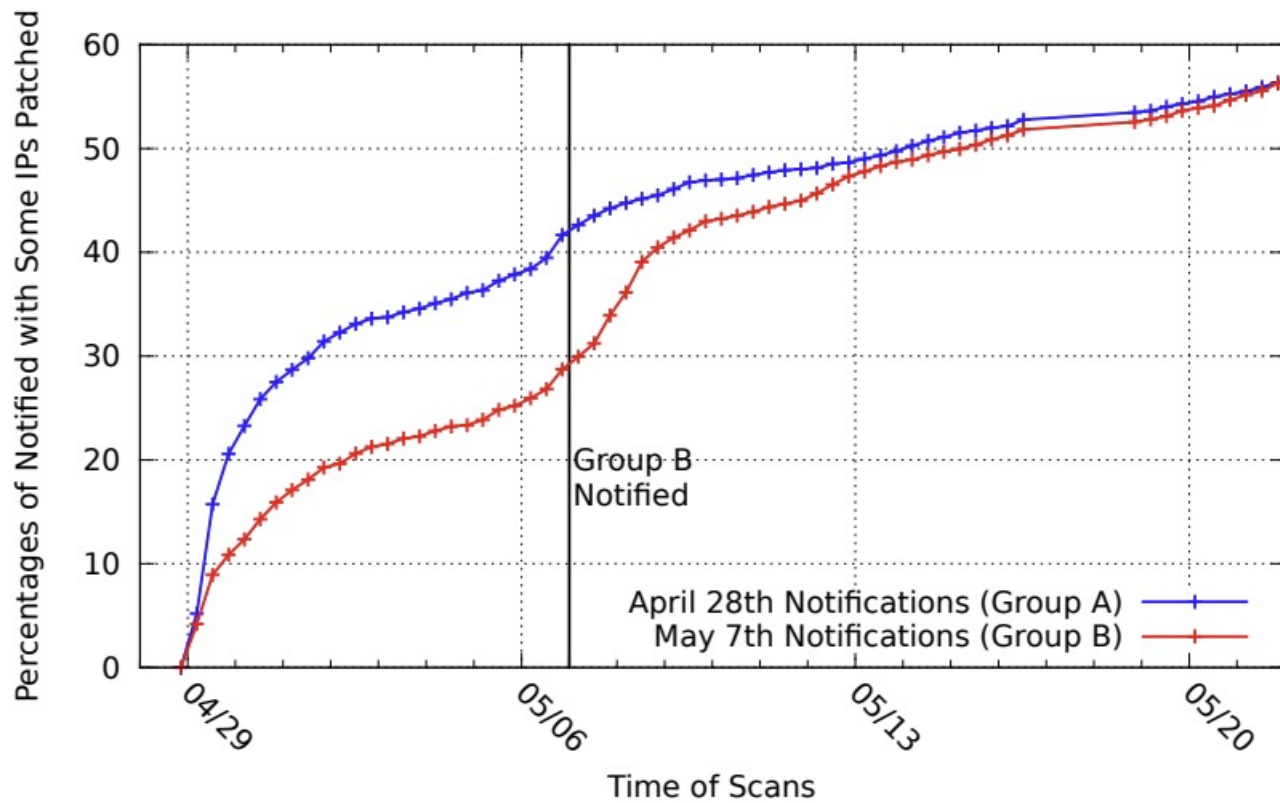


Figure 9: Patch Rates of Group A vs Group B. The patch rates for our two notification sets show that notification had statistically significant impact on patch rate.

Generalizing notifications

“You've Got Vulnerability: Exploring Effective Vulnerability Notifications”

Trend is that notifications are becoming a more frequent aspect of security measurement work

- Foreshadowing expectations by security conferences today

Natural progression after Heartbleed paper to examine the question of best practices

- **How** should you do notifications?
- Start with a best practices paper
- Appendices have example notification messages

Notified about three different vulnerabilities

- Dual-stack IPv4/IPv6, Industrial Control Systems, DDoS amplifiers

Factors in notifications

What are factors that could influence notifications?
(even independent of this paper, think of considerations)

- How well known the vulnerability is
- Severity of issue
- Who you send it to
- Who it comes from
- Length of message
- Detail in message
- Language
- Difficulty of patching
- How many times notified

Summary of results

Who to contact

- WHOIS abuse contact, national CERT, US-CERT (Computer Emergency)
- Not...as useful today

Detail length

```
11:14 (124) www> whois amazon.com | head
Domain Name: AMAZON.COM
Registry Domain ID: 281209_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2024-09-29T09:48:15Z
Creation Date: 1994-11-01T05:00:00Z
Registry Expiry Date: 2025-10-31T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
```

- Highlights many issues factor into patching (e.g., control of, or knowledge about)

Repeat notifications not effective

Stepping back

Useful answers to many methodological choices

Notifications improved patching outcomes for two services

...but majority of systems remained unpatched

- Notifications gave an 11% bump over control

Larger question of what to do to improve patching

- Long-standing hard problem

Should ... this result factor into decision by security conferences to expect notifications?

Questions?

A few things to remember when measuring

Always start with trying to understand your data before measuring

- Don't assume; no substitute for taking a random sample and eyeballing it manually

Do small measurements first and focus on validation

- Internal consistency – testing to make sure obvious things are true
- External consistency – could include, other measurement approach, injecting known data, diurnal patterns, correlation with other events (e.g., failures), etc

Measure/log as much as you can; you can't always go back

- Make sure to think about error cases and log them; mechanism to alert when measurement infrastructure fails

Manual sampling for error

- If you can't measure error comprehensively; take sample (e.g., a few hundred random items) and validate manually – totally reasonable way to estimate error

Project presentations

Monday, 6/8 in this room, 11:30 until done

We have 10 presentations; 6 solo, 4 groups

- We'll do solo first: Forristal, Zhu, Babior, Yuan, Kosta, Hanlon
 - Then groups: Gao/Jeong, Murthy/Dutta/Liu/Wang, Lee/Lee, Quang, Xie/Celik
- For groups, everyone should be involved in presentation
- Everyone stays for the whole thing (i.e., no exiting after your presentation)

8-10 mins – problem, motivation, methodology, challenges/issues, results, open questions (if appropriate)

- Please also send me copy of slides

Papers, no more than 6 pages, due 6/12 to me by email