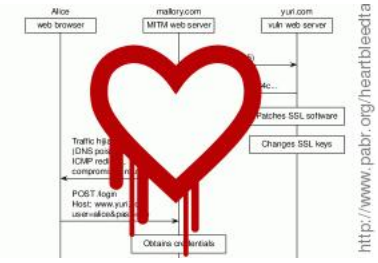
A taxonomy of Heartbleed attacks

One week after the disclosure of the Heartbleed vulnerability, security experts are still discovering new ways it could affect online security. This document attempts to classify attack patterns and their countermeasures as they emerge. It will help you understand whether you really need to change your passwords, why it was unwise to do it too early, and why you should know whether your browser detects revoked certificates.



READ THE HYPERTEXT VERSION HERE:
http://www.pabr.org/heartbleedtax/heartbleedtax.en.html

Revision History		
1.6	2014-06-09	Added EAP.
1.5	2014-05-08	Added electronic voting. Correction DNSSEC.
1.4	2014-04-22	Debate about certificate revocation checking.
1.3	2014-04-21	Added Tor hidden services, HTTPS proxies.
1.2	2014-04-19	Added VPN server. More references.
1.1	2014-04-18	Added Tor traffic correlation. French translation.
1.0	2014-04-16	Initial release.

Table of Contents

1. Background

The Heartbleed vulnerability is possibly the worst thing that ever happened to online trust. Ironically, Heartbleed makes HTTPS less secure than plain HTTP because attackers can obtain sensitive data without even having to intercept traffic. For details about the bug itself, see [HEARTBLEED], [CVE-2014-0160] and [XKCD1354].

Initial reactions focused on patching vulnerable web servers, revoking SSL certificates and changing user passwords. It took a couple more days to realize that Heartbleed also affects client software, non-web SSL traffic and countless embedded devices which will never receive a software update.

2. Notations

In the following sections, attack scenarios are illustrated with Message Sequence Charts. See Section 6, “ Acknowledgements ” for LaTeX source code examples.

Following an old tradition of cryptographic literature, character names are associated with archetypal roles:

- **Alice**, **Bob**: Users of online services.

- **Eve**: A passive attacker (eavesdropper).

- **Trudy**: An active attacker who may exploit the Heartbleed bug by sending specially crafted heartbeat packets over an SSL (e.g. HTTPS) connection.

- **yuri.com**: A web site with vulnerable SSL software.

  Since OpenSSL is so widely used, any web site should be considered as a potential yuri.com until proven otherwise.

- **george.com**: A web site with unaffected or patched SSL software.

  All prominent web sites are probably safe at the time of writing, but thousands of minor sites will probably never comment on the issue, leaving their users wondering. If you have doubts about your favorite web site, try [LASTPASS]. You can also check manually whether the site certificate has a NotBefore date posterior to 7 April 2014. This is a good indication that they have dealt with Heartbleed (although this may yield both false positives and false negatives because a site can renew an expired certificate without changing the keys, and also replace the keys without affecting the dates). As a last resort, [FILIPPO] lets you test whether a web server is still vulnerable. Warning: This service will "attack" any server on your behalf, which might bring legal trouble.

- **Victor**: A user with vulnerable SSL software.

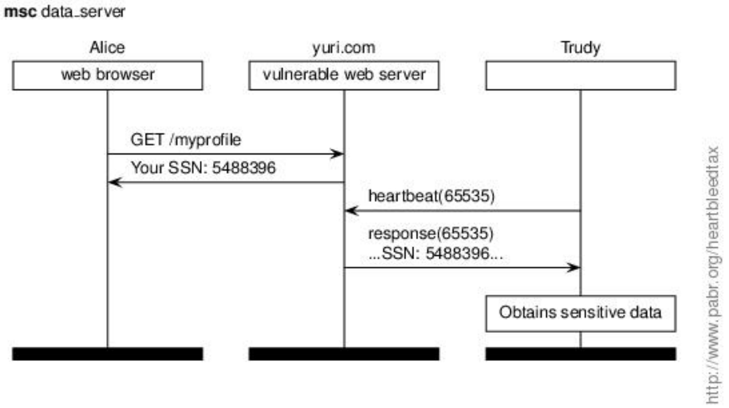
  To determine whether you are in Victor's shoes rather than Alice's, check your version of the OpenSSL and/or libSSL packages. Versions 1.0.1 through 1.0.1f are vulnerable (except when compiled with non-default options). According to [ARS_CLIENTS], Android 4.1.1 is affected. If you are unsure, see [REVERSEHEARTBLEED]. Warning: Do not send the malicious URLs to anyone.

- **mallory.com**: A malicious server which may impersonate web sites or attack vulnerable SSL clients.

- **brad.com**: A host which serves content for prominent web sites, such as advertisements, static resources, web beacons, Javascript libraries, CSS stylesheets, multimedia files.

3. Attack patterns and countermeasures

3.1. Extraction of sensitive data from vulnerable HTTPS servers



In this scenario **Alice** enters or consults sensitive data on **yuri.com** over HTTPS. Plain-text data linger in the memory of the web server. Later, **Trudy** connects as a regular HTTPS clients and exploits Heartbleed.

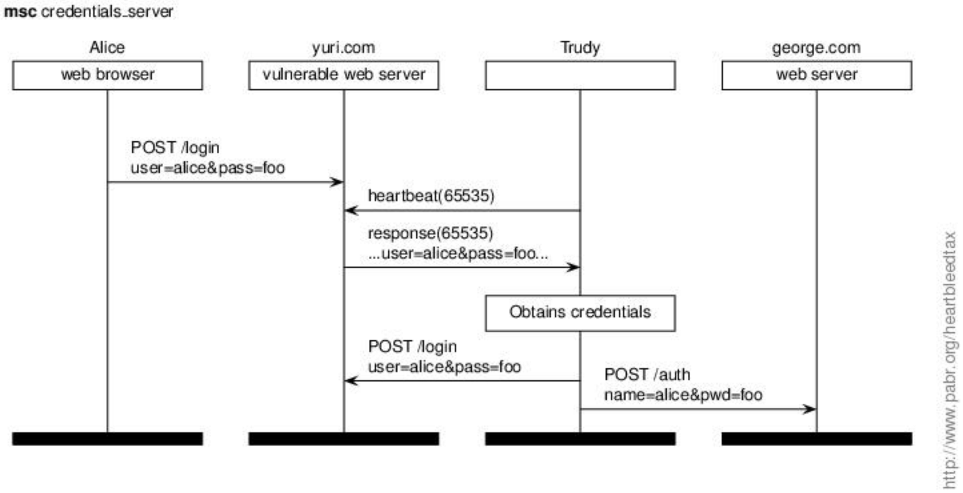
Countermeasures for end users.

- Do not exchange sensitive information with a web site until they tell you they have dealt with Heartbleed.

Field reports.

- Canadian charged in 'Heartbleed' attack on tax agency [http://www.reuters.com/article/2014/04/16/us-cybersecurity-heartbleed-arrest-idUSBREA3F1KS20140416] (reuters.com)

3.2. Extraction of login credentials from vulnerable HTTPS servers



In this scenario **Alice** authenticates with **yuri.com** by supplying a login and password over HTTPS. Later, **Trudy** connects as a regular HTTPS clients and exploits Heartbleed to extract these credentials.

Trudy can then access **Alice**'s accounts on **yuri.com** and on any other web site where she used the same username and password.

Countermeasures for end users.

- Do not enter your password on a web site until they tell you they have dealt with Heartbleed.
- Afterward, change your password.
- In the meantime, beware of phishing attempts, e.g. emails which invite you to come and change your password on a fake web site.

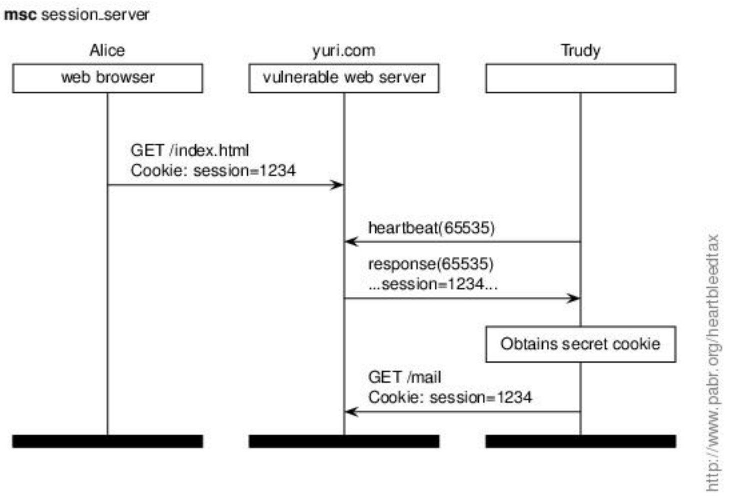
Lessons learned.

- Never use the same password on several web sites.
- Use multi-factor authentication.
- Use SSL mutual authentication (client-side certificates) ?

Field reports.

- "The passwords and personal messages of up to 1.5 million Mumsnet users may have been exposed" [http://www.telegraph.co.uk/technology/internet-security/10766872/Heartbleed-hackers-hit-Mumsnet-website.html] (telegraph.co.uk)

3.3. Session hijacking from vulnerable HTTPS servers

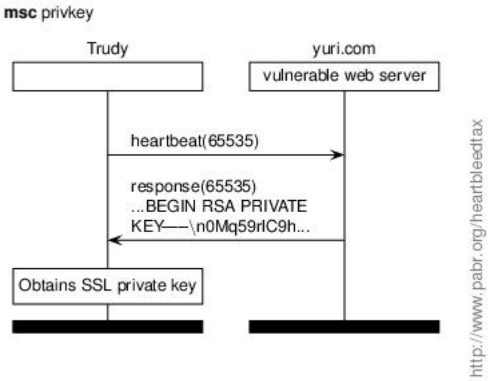


In this scenario **Trudy** extracts session cookies rather than login credentials. This allows her to take control of **Alice**'s account without waiting for her to enter her credentials.

Countermeasures for end users.

- Log out of online services until they have dealt with Heartbleed.

3.4. Extraction of SSL private keys from vulnerable HTTPS servers



In this scenario **Trudy** extracts the SSL/TLS private key of **yuri.com**. Regardless of what happens next (see below), leakage of private keys is always a major failure.

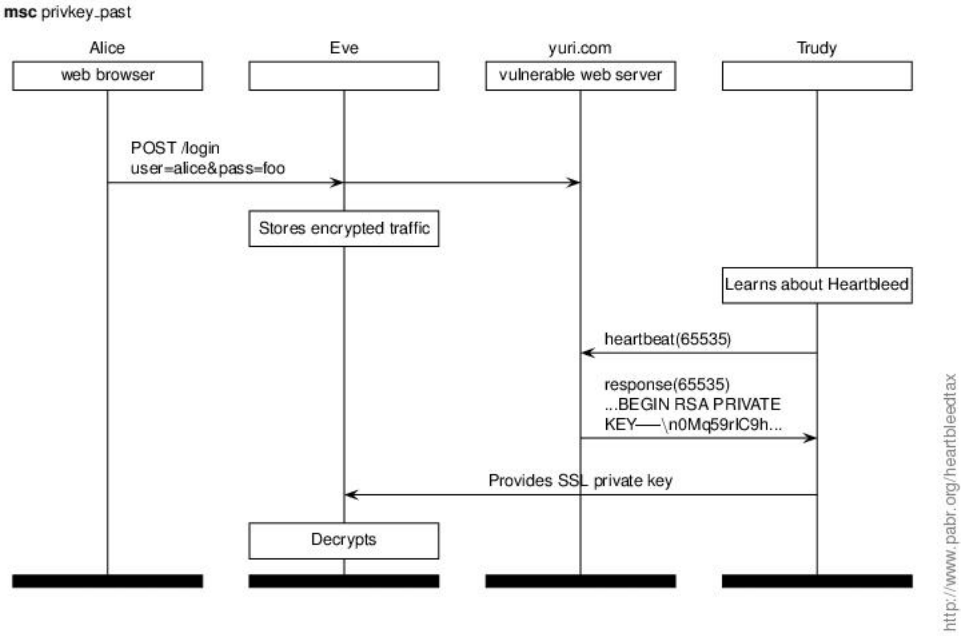
Lessons learned.

- Protect private keys with a hardware security module.

Field reports.

- Confirmed: Heartbleed Exposes Web Server's Private SSL Keys [http://www.securityweek.com/confirmed-heartbleed-exposes-web-servers-private-ssl-keys] (securityweek.com)

3.5. Decryption of old traffic intercepts



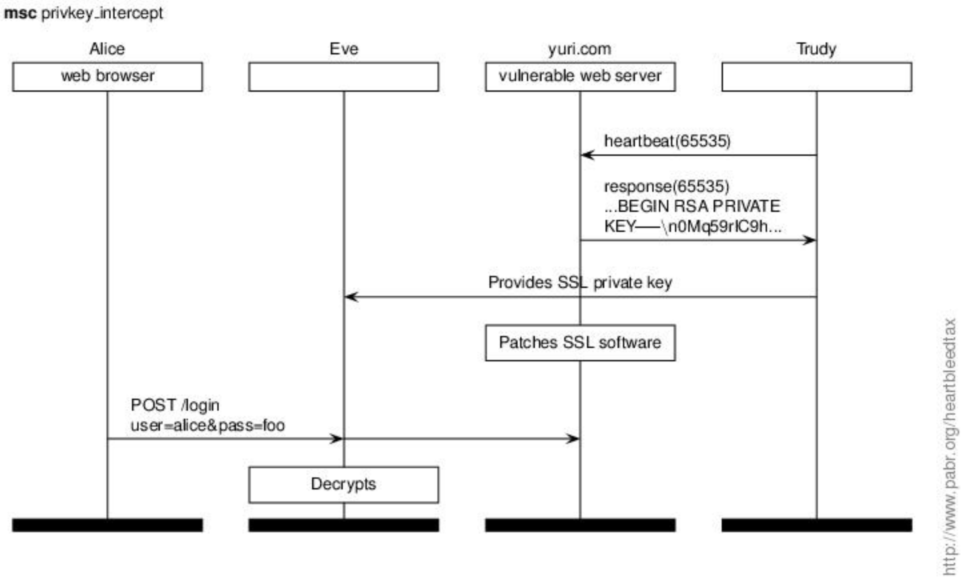
In this scenario **Trudy** extracts the SSL private key of **yuri.com** and uses it to decrypt traffic that was previously intercepted by **Eve**, possible before the Heartbleed bug was even introduced in OpenSSL.

Here comes the return on investment for those entities which are in the business of archiving petabytes of encrypted traffic !

Lessons learned.

- Use perfect forward secrecy.
- Change key pairs regularly.

3.6. Decryption of new traffic intercepts

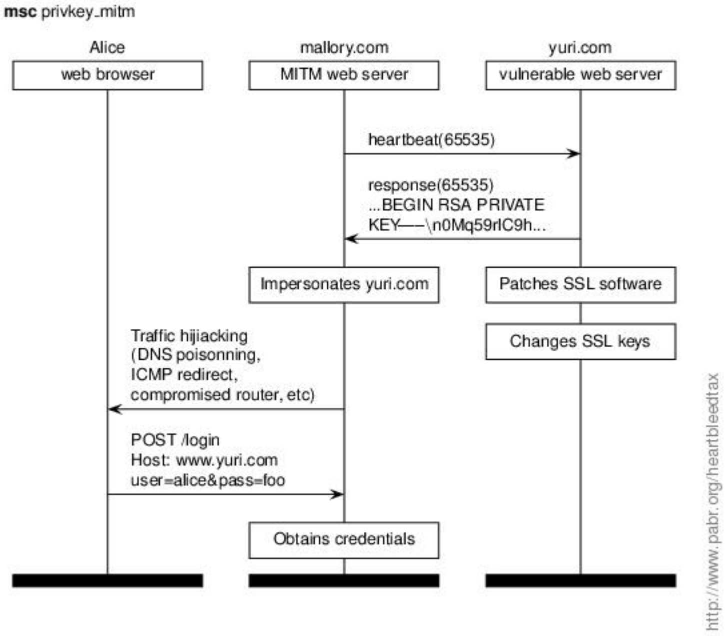


In this scenario **Trudy** extracts the SSL private key of **yuri.com** and uses it to decrypt traffic, even after **yuri.com** has patched its SSL software.

Countermeasures for end users.

- Check that affected web sites have either changed their keys or certified that the keys were never at risk.

3.7. Man-in-the-middle impersonation of online services



In this scenario **mallory.com** impersonates **yuri.com** after extracting its SSL private key. This so-called man-in-the-middle attack (MITM) is more dangerous than passive snooping because **mallory.com** can trick **Alice** into using a compromised certificate. It also allows **mallory.com** to defeat some multi-factor security measures.

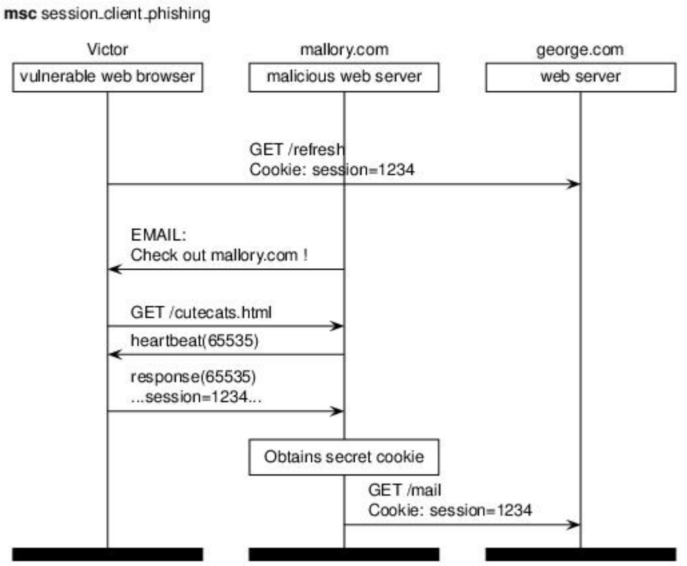
Countermeasures for end users.

- Check whether your browser detects revoked certificates: [REVOKED_GRC].
- If it does not, inspect certificates manually.

Lessons learned.

- Heartbleed will probably be the end of the current certificate revocation infrastructure. [LANGLEY] explains the problem and mentions alternatives such as OCSP stapling and DNSSEC.

3.8. Extraction of data from vulnerable HTTPS browsers via phishing



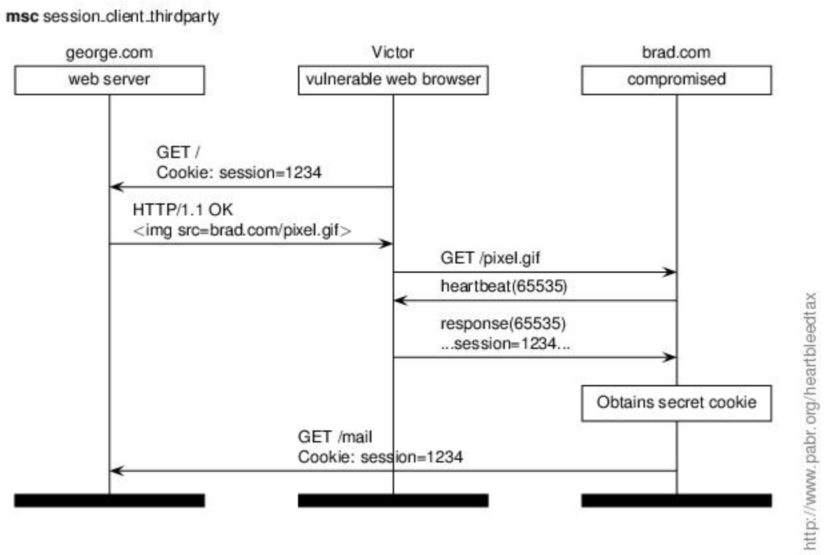
http://www.pabr.org/heartbleedtax

In this scenario **mallory.com** extracts data from **Victor**, a client with vulnerable SSL software.

Countermeasures for users.

- Upgrade your browser software.
- Use distinct user accounts or browser instances for casual web surfing and for secure transactions.

3.9. Extraction of data from vulnerable HTTPS browsers via third-party content

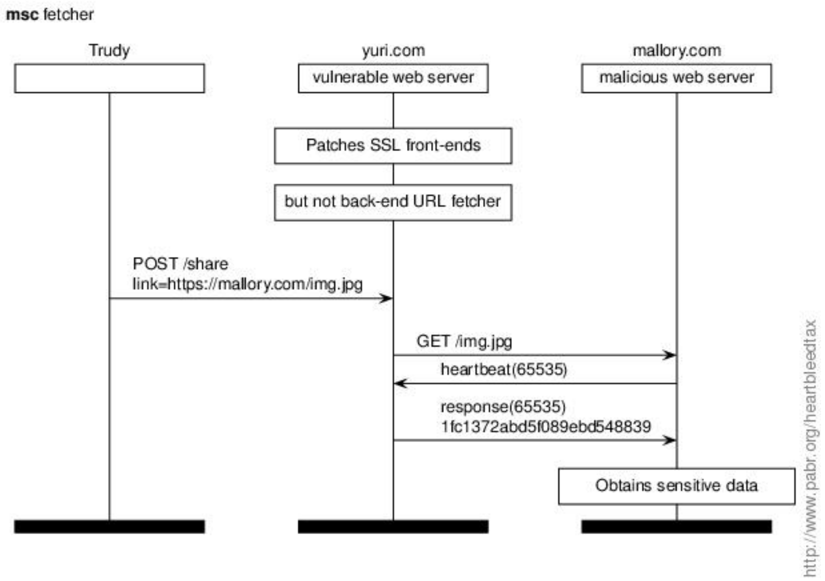


In this scenario **Trudy** takes control of **brad.com**, one of many providers which serve web resources for **george.com**. **Victor** is attacked even though he believed he was only connecting to **george.com**, whom he trusts.

Countermeasures for users.

- Upgrade your browser software.
- Block third-party content.

3.10. Extraction of data from vulnerable URL fetchers



Here **Trudy** tricks **yuri.com** into retrieving a URL from **mallory.com**, which uses that opportunity to extract data from a vulnerable back-end server of **yuri.com**. This scenario affects web crawlers, social networks which pre-render links shared by users, page translation services, HTML compliance testers, etc. However, it is still unclear whether any sensitive data would leak.

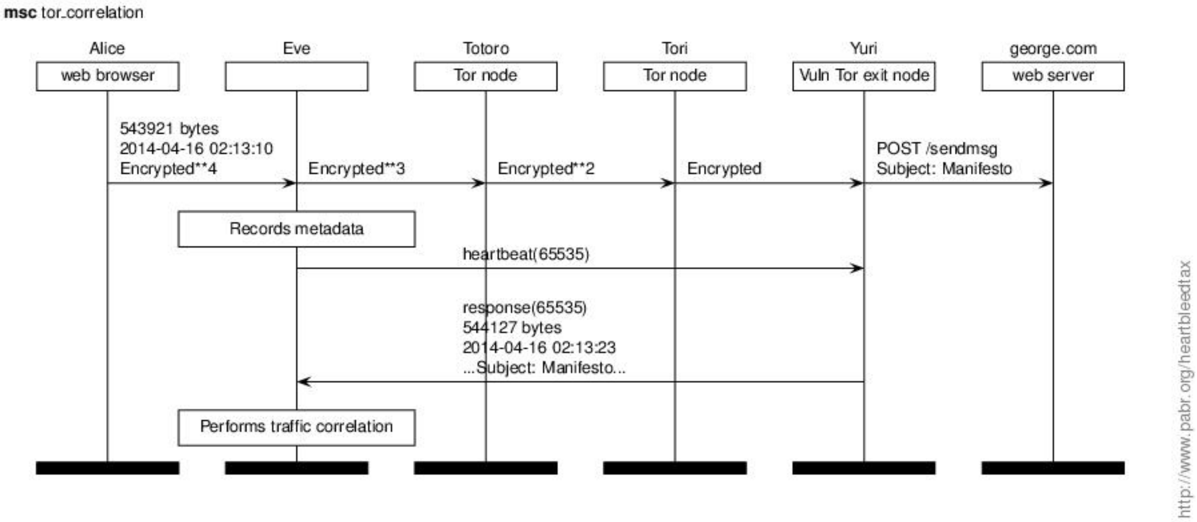
Lessons learned.

- Back-end machines deserve as much care as web front-ends.

Field reports.

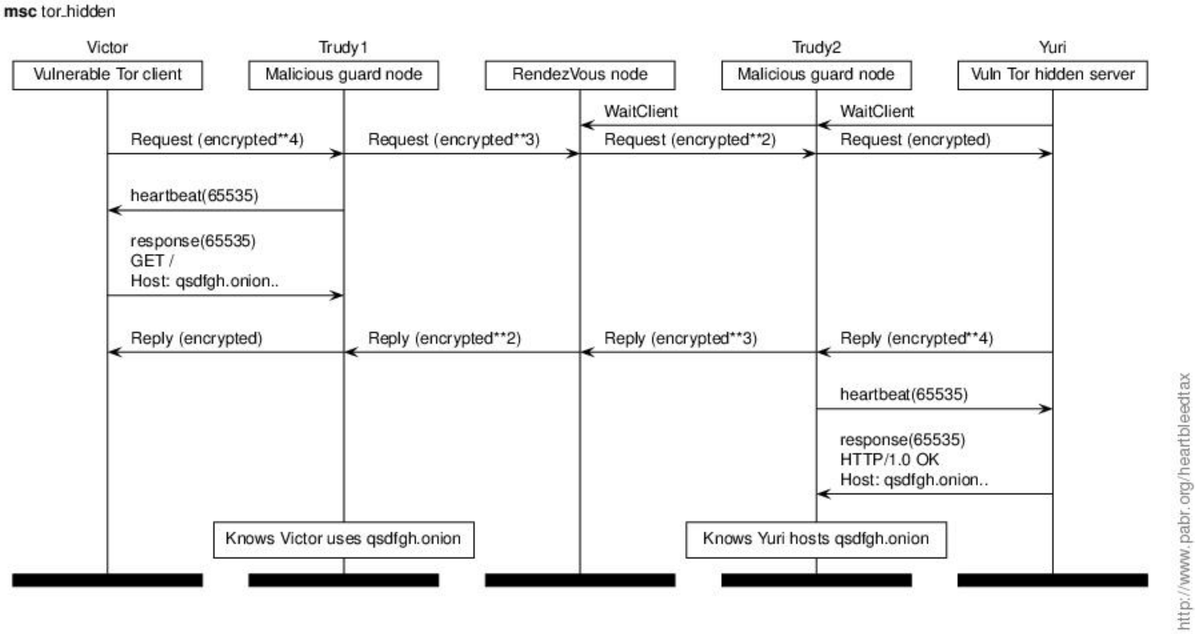
- Testing for "reverse" Heartbleed - What did we find ? [http://blog.meldium.com/home/2014/4/10/testing-for-reverse-heartbleed] (meldium.com)

3.11. Tor traffic correlation



Eve, an evil dictator, wants to establish that **Alice** is using Tor to communicate with **george.com**, a foreign human-rights organization. **Eve** can spy on all Internet traffic within her national boundaries but has no wiretapping authority in other countries. She exploits Heartbleed massively against vulnerable Tor exit nodes in order to match outgoing traffic with her local intercepts.

3.12. De-anonymization of hidden servers and users by malicious Tor nodes

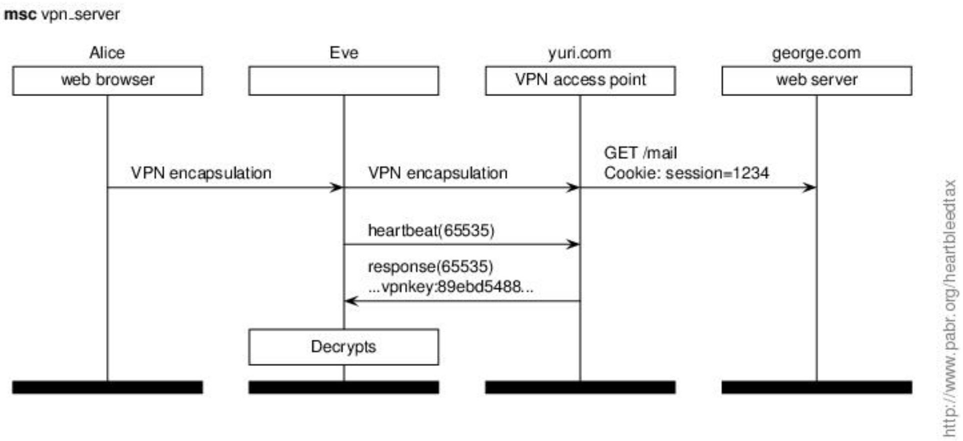


Trudy sets up a number of malicious Tor guard node. She exploits the Heartbleed vulnerability against clients that connects to them, including Tor hidden servers and their users. Although Tor hidden services are encrypted end-to-end, **Trudy** can identify vulnerable users and servers based on plaintext data leaked by Heartbleed at each end. Besides, if she extracts the private key of a hidden service, she can impersonate it.

Field reports.

- "Tor hidden services might leak their long-term hidden service identity keys to their guard relays." [https://blog.torproject.org/blog/openssl-bug-cve-2014-0160] (torproject.org)

3.13. Attacks against VPN servers



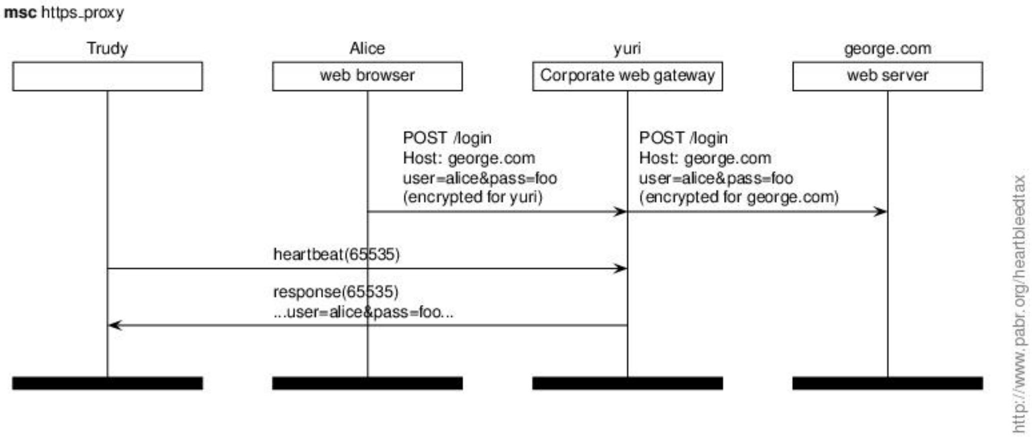
Alice is aware that most public WiFi networks provide no privacy. Therefore she has configured her smartphone and laptop to connect to the Internet through a VPN service provider, **yuri.com**. (Alternatively, she could be running her own VPN server at home, or using the VPN feature that comes bundled with her DSL modem or her NAS box.) **Eve** snoops on the WiFi network that **Alice** is currently using, notices SSL-based VPN traffic from her smartphone, exploits Heartbleed against the destination IP address, and retrieves either VPN keys or plaintext traffic.

Field reports.

- Attackers Exploit the Heartbleed OpenSSL Vulnerability to Circumvent Multi-factor Authentication on VPNs [https://www.mandiant.com/blog/attackers-exploit-heartbleed-openssl-vulnerability-circumvent-multifactor-authentication-vpns/] (mandiant.com)

- "OpenVPN uses OpenSSL as its crypto library by default and thus is affected" [http://community.openvpn.net/openvpn/wiki/heartbleed] (openvpn.net)

3.14. Attacks against vulnerable HTTPS proxies



Alice does online banking from her work computer. She has checked that her bank's website, **george.com**, is not affected by Heartbleed. However, her employer routes outgoing web traffic through **yuri**, a web gateway which enforces acceptable use policies and scans for malware. To inspect HTTPS traffic, **yuri** acts as a certification authority trusted by **Alice**'s browser (it will generate a certificate on the fly for **george.com**). **Trudy**, a co-worker of **Alice**, exploits Heartbleed against the proxy and extracts sensitive information.

Countermeasures for end users.

- Do not exchange sensitive information with a web site until your IT department has dealt with Heartbleed.

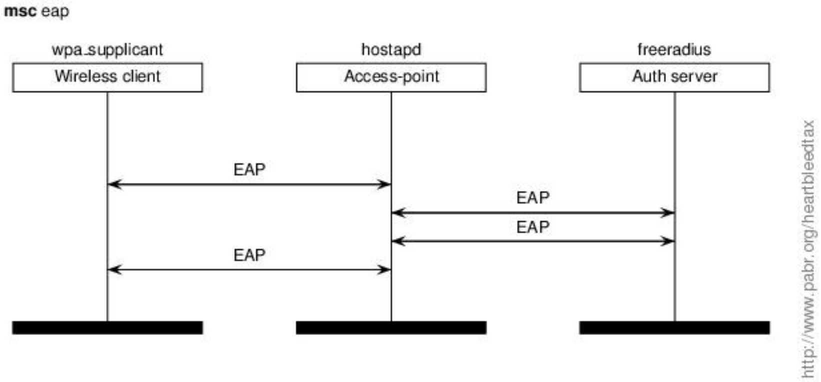
Lessons learned.

- The chain of trust is only as strong as its weakest link.

Field reports.

- List of web gateways and other security appliances affected by Heartbleed [https://www.cert.fi/en/reports/2014/vulnerability788210.html] (cert.fi)

3.15. Attacks against vulnerable EAP implementations



Enterprise networks typically rely on EAP variants which provide stronger security than the PSK mode used in home networks. The EAP-PEAP, EAP-TLS and EAP-TTLS standards happen to use TLS messages (like HTTPS, except not over TCP). Popular implementations (**wpa_supplicant**, **hostapd**, and **freeradius**) are based on **OpenSSL** and may therefore be vulnerable to Heartbleed.

Scenarios.

- Malicious client extracts server certificate from vulnerable AP, then uses it to decrypt traffic.
- Fake AP extracts credentials from vulnerable client, then uses them to connect to the real AP.

In configurations where the AP delegates authentication to a central server, insiders may also attack the server itself.

Note that EAP is used not only in wireless networks, but also for port-based network access control in wired ethernet (IEEE 802.1X).

Field reports.

- **hostap** mailing-list [http://comments.gmane.org/gmane.linux.drivers.hostap/30091].
- [CUPID]

4. Perspectives

Besides HTTPS, several SSL/TLS-based services and protocols might turn out to be affected by Heartbleed:

- Online trusted timestamping and notarization services, where private keys are used for signature rather than privacy
- Automatic software update infrastructures.
- RADIUS, Diameter and similar security protocols, when used over TLS.
- SMTPS, POP3S, IMAPS.
- SIPS.
- Electronic currency protocols and wallet applications.
- Open-source development infrastructures and source code repositories (git).
- Online voting systems.

5. Expected aftermath

Hopefully the Heartbleed fiasco will educate users about online security and stimulate progress, such as:

- Research in secure yet efficient programming languages.
- Revamping the whole X509 certification infrastructure so that client software has no excuse to accept revoked certificates.

6. Acknowledgements

Message Sequence Charts are rendered with [MSCSTY]. Here is an expanded LaTeX source code example: credentials_server_example.tex

Bibliography

[HEARTBLEED] *The Heartbleed Bug* . http://heartbleed.com.

[CVE-2014-0160] *CVE-2014-0160*. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160.

[XKCD1354] *How the Heartbleed bug works* . Randall Munroe. http://xkcd.com/1354/.

[FILIPPO] *Heartbleed Test* . https://filippo.io/Heartbleed/.

[LASTPASS] *LastPass Heartbleed checker* . https://lastpass.com/heartbleed/.

[REVOKED_GRC] *Security Certificate Revocation Awareness Test* . https://revoked.grc.com/.

[ARS_CLIENTS] *Vicious Heartbleed bug bites millions of Android phones, other devices* . http://arstechnica.com/security/2014/04/vicious-heartbleed-bug-bites-millions-of-android-phones-other-devices/.

[REVERSEHEARTBLEED] *Reverse Heartbleed Tester* . https://reverseheartbleed.com/.

[CUPID] *Heartbleed, Cupid and Wireless* . http://www.sysvalue.com/en/heartbleed-cupid-wireless/.

[LANGLEY] *No, don't enable revocation checking* . Adam Langley. https://www.imperialviolet.org/2014/04/19/revchecking.html.

[MSCSTY] *Drawing Message Sequence Charts with LaTeX* . Sjouke Mauw and Victor Bos. http://satoss.uni.lu/software/mscpackage/.