



# Recipes for enabling HTTPS

The DevOps Approach  
to Setting up Robust HTTPS Web Apps



# OWASP

The Open Web Application Security Project

thomas.herlea@trasysgroup.com  
nelis.boucke@archiwise.com  
yo@johanpeeters.com



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences



# OWASP

The Open Web Application Security Project

# Motivation

## HTTPS is

- hard to set up (securely)
- hard to maintain



# OWASP

The Open Web Application Security Project

# Motivation

## 1. Evolving advice on getting HTTPS right

SSL stripping	2009	MiTM during HTTP prevents switching to HTTPS.
Insecure renegotiation	2009	MiTM can perform operations on server on client's behalf.
BEAST	2011	Forced padding verification errors in CBC mode leak plaintext.
CRIME	2013	Forced variable length after TLS compression leaks plaintext.
Lucky 13	2013	Forced variable duration of MAC verification leaks information.
RC4	2013	Session cookie forced into many TLS sessions is leaked by RC4 bias.
Forward secrecy	2013	Mass surveillance + data retention + obtaining server key = attacker decrypts old traffic
BREACH	2013	Like CRIME, but HTTP compression.



# OWASP

The Open Web Application Security Project

# Motivation

1. Evolving advice on getting HTTPS right
2. Poor deployment of known mitigations
  - Still vulnerable to CRIME > **19%**
  - Still supporting insecure SSL 2.0 (after 2 years) > **27%**
  - Still supporting weak and insecure cipher suites > **33%**
  - Still vulnerable to BEAST (after 2 years) > **65%**
  - Still no support for TLS 1.2 (after 5 years) > **80%**



# OWASP

The Open Web Application Security Project

# Motivation

HTTPS is

- hard to set up (securely)
- hard to maintain

Systematic approach needed

- Repeatable
- Knowledge capturing and sharing
- Agility to react on changing advice
- Assurance / Verification



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences



# OWASP

The Open Web Application Security Project

# DevOps

DevOps = Dev and IT Operations convergence

- Repeatable
  - Infrastructure as code, automate procedures
  - Recipes in languages like CFEngine, *Puppet*, Chef
- Knowledge capturing and sharing
  - Code = always up-to-date documentation
  - Build on existing modules
  - Abstraction



# OWASP

The Open Web Application Security Project

# DevOps

- Agility to react on changing advice
  - Shorter release cycles through automation
- Assurance / Verification
  - Source control for traceability
  - Easy to replicate (production) environment for testing and verification



# OWASP

The Open Web Application Security Project

# Puppet

Desired state  
(declarative)



I want Nginx 1.4.2 with  
with ssl enabled and  
specific cipher selection

Resource  
Abstractions





# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences

Our github repo:

<https://github.com/JohanPeeters/secure-webserver.git>



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences

Our github repo:

<https://github.com/JohanPeeters/secure-webserver.git>



# OWASP

The Open Web Application Security Project

# Version hell

Ideally: use LTS => Ubuntu Server 12.04.2

Problems:

- nginx < 1.4.2 does not support TLS 1.2
- openssl < 1.0.1e does not support GCM
- ruby < 2.0.0-p247 suffers from hostname check bypassing

Solution?

compile from source?

but then we've blown the LTS-ness...



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences

Our github repo:

<https://github.com/JohanPeeters/secure-webserver.git>



# OWASP

The Open Web Application Security Project

# Mitigations

SSL stripping	HSTS	correct configuration
Insecure renegotiation	Use OpenSSL version that supports RFC 5746 (> v0.9.8k).	software version and correct configuration
BEAST	No CBC mode prior to TLS 1.1.	cipher list
CRIME	No TLS compression.	software version
Lucky 13	No CBC mode.	cipher list
RC4	No RC4.	cipher list
Forward secrecy	No RSA, PSK or SRP key exchange.	cipher list
BREACH	No HTTP compression.	correct configuration

Sources for mitigations: SSL labs, OWASP TLS cheat sheets, ...



# OWASP

The Open Web Application Security Project

# TDD

- write failing test
- write code to fix the test
- refactor
- repeat

# Where do ciphers come from?



**OWASP**

The Open Web Application Security Project

openssl ciphers -V

IANA TLS Cipher Registry

**CipherSpec**

name  
protocol\_version  
kXchange\_alg  
mode  
....

accepted

rejected



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences

Our github repo:

<https://github.com/JohanPeeters/secure-webserver.git>



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences

Our github repo:

<https://github.com/JohanPeeters/secure-webserver.git>



# OWASP

The Open Web Application Security Project

# Overview

- Motivation
- DevOps
- Demonstrations
  - Puppet basics
  - Configuration of cipher list
  - Configuration of webserver
  - Vagrant as tool for testing
- Experiences



# OWASP

The Open Web Application Security Project

# Experiences

## Is configuring HTTPS hard?

- Cipher lists are fragile
  - easy to make errors in cipher list
  - some errors might stay undetected without testing
- Custom webserver installation was required
  - tradeoff with LTS and stability?

⇒ What are your chances with one-off manual installation or configuration?



# OWASP

The Open Web Application Security Project

# Experiences

## Did DevOps help?

- Systematic approach
- Allows for extensive testing and experimentation



# OWASP

The Open Web Application Security Project

# Experiences

## Proof of concept with limitations:

- Only indirect property testing
- Added risk of using Puppet?
- Only hardening for HTTPS
  - Attacks on other software
  - User management
  - Integrity check of installed software?



# OWASP

The Open Web Application Security Project

## Take away

# systematic HTTPS → DevOps

## Code and test!



# OWASP

The Open Web Application Security Project

# References

Our github repo:

<https://github.com/JohanPeeters/secure-webserver.git>

SSL Labs:

<https://www.ssllabs.com/>

sslyze

<https://github.com/iSECPartners/sslyze.git>

OWASP Transport Layer Protection Cheat Sheet

[https://www.owasp.org/index.php/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet](https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet)