



O-Saft

# Richtig verschlüsseln mit ~~SSL~~/TLS

OWASP Day Germany 2014, Hamburg, 9. Dezember 2014

Achim Hoffmann

Torsten Gigler



## OWASP

The Open Web Application Security Project



## OWASP

German OWASP Day 2014

9. Dezember  
Hamburg





- Achim Hoffmann

[achim@owasp.org](mailto:achim@owasp.org)

- (Pen-)Tester, Trainer, Sprecher
  - spezialisiert auf Web Application Security >15 Jahre
  - Pentest, SCA, WAF ([achim.hoffmann@sicsec.de](mailto:achim.hoffmann@sicsec.de))
- OWASP Germany, Board Member; Project Leader

- Torsten Gigler

[torsten.gigler@owasp.org](mailto:torsten.gigler@owasp.org)

- Interner Sicherheitsberater in einem Großunternehmen >15 Jahre (IT-Infrastruktur- und Anwendungs-Sicherheit)
- OWASP:
  - seit 2 Jahren aktiv, z.B. Projektleiter 'OWASP Top 10 für Entwickler'
  - seit 1 Jahr Mitentwickler des Tools O-Saft

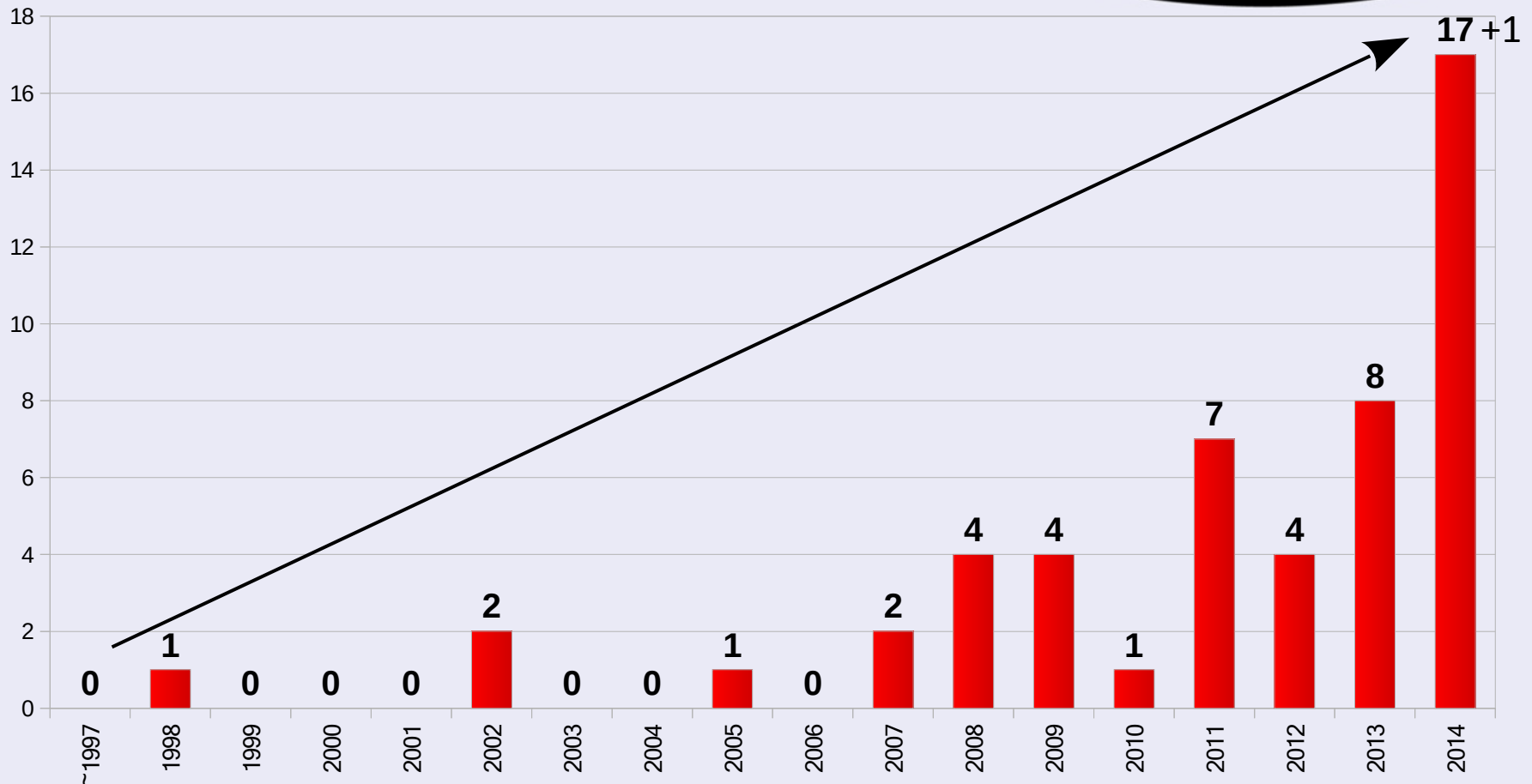
# SSL / TLS – sicher?

## Probleme



# OWASP

The Open Web Application Security Project



12 Jahre 1998 – 2010: 15

4 Jahre 2011 – 2014: 36+1 (8. Dez 2014)



- SSLv2 und SSLv3 deaktivieren
- TLS v1.0 nur wenn unbedingt nötig
- TLS v1.1, TLS v1.2 aktivieren (RC4, BEAST)
- Keine Renegotiation vom Client erlauben
- Kompression in SSL abschalten (CRIME)
- Starke Zertifikate ( $\geq 2048$  Bit, SHA2)
- Starker Diffie-Hellman-Parameter ( $\geq 2048$  Bit)





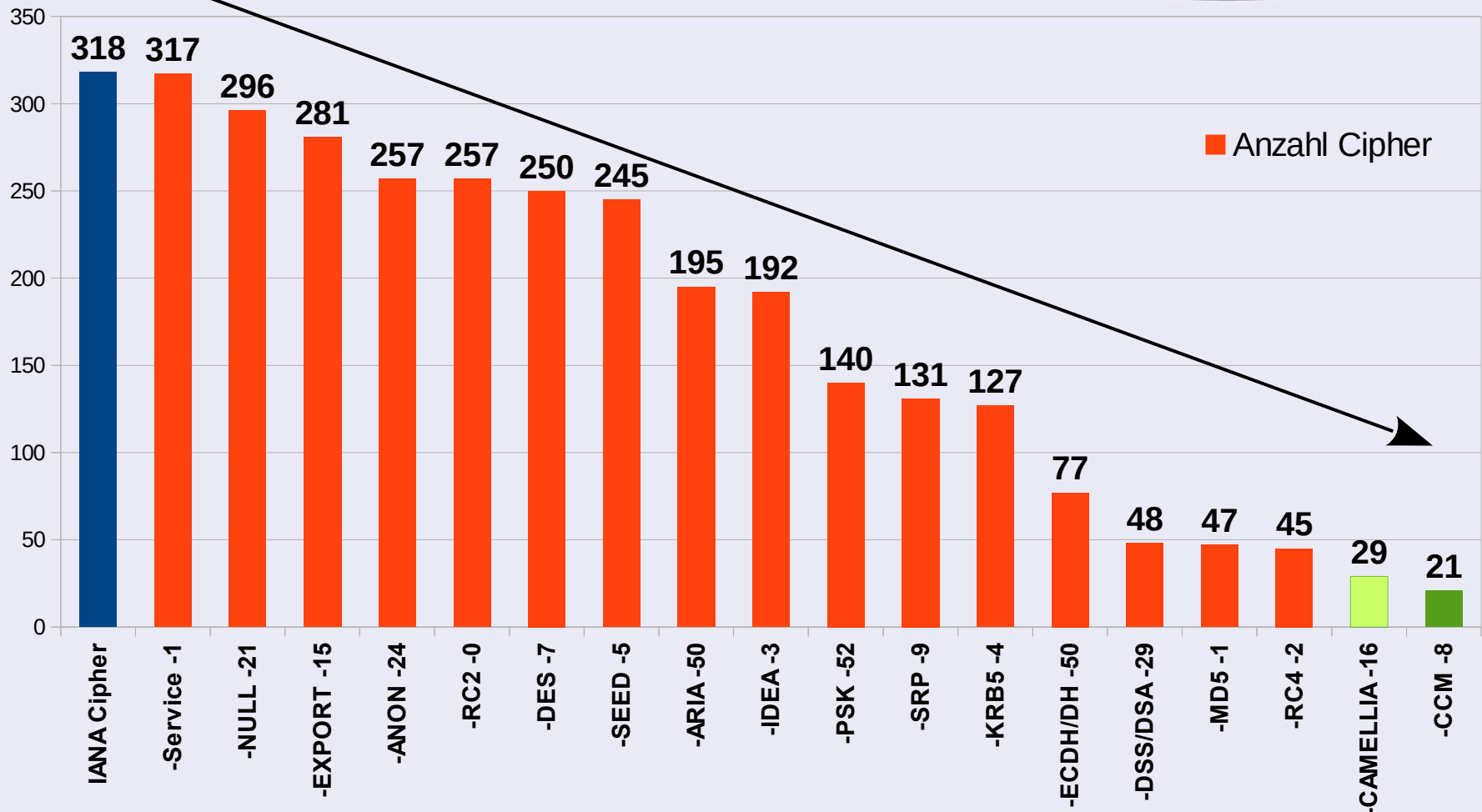


# OWASP

The Open Web Application Security Project

## Cipher Suites

was bleibt übrig?



Von **318** definierten Ciphern bleiben nur **21 sichere und brauchbare** übrig



- keine NULL-Cipher
- keine EXPORT-Cipher
- keine anonymen Cipher (ADH, AECDH)
- keine "WEAK"-Cipher (RC2, RC4, DES, MD5)
- keine Exoten (PSK, SRP, SEED, IDEA, ARIA)
- DSA/DSS meiden (Gefahren bei schlechter Entropie)
- keine Camellia-Cipher (Sicherheit z.Zt. unklar)
- keine CCM-Cipher (da noch nicht unterstützt)



- Reihenfolge festlegen, stärkste Cipher zuerst
- Perfect Forward Secrecy mit hoher Prio (EDH, ECDHE)
- Keysize min. 128 Bits
- GCM-Cipher bevorzugen
- Integritätssicherung mit HMAC (SHA2: z.B. SHA256)
- Cipher mit CBC-Mode temporär (nur wenn für alte Browser notwendig)
- RSA/AES und RSA/3DES für alte Browser am Ende





# OWASP

The Open Web Application Security Project

## Cipher Suites

was übrig bleibt!

### Mit Perfect Forward Secrecy

TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384

TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256

TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256

TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA

TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256

(TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA)♦

~~TLS\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA\*♦~~

TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384

TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256

TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384

TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA

TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256

TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA

~~TLS\_ECDHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA\*~~

### Ohne Perfect Forward Secrecy

TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384

TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256

(TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256)\*

TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA

(TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256)\*

TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA

TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA\*

\*: 3DES nur für ganz alte Browser nötig (→ RSA)

♦: entfällt wg. Inkompatibilität alter Browser mit starken DH-Parametern (≥ 2048 Bit)

✧: von alten Browsern i.d.R. nicht unterstützt

# Protokolle / Cipher Suites

## Beispiel + Browser-Simulation



# OWASP

The Open Web Application Security Project

Protokolle:	Android (ab 2.3.7)	Chrome (ab 30 / Win 7)	Firefox (ab 21/Fedora 19 bzw. 24/Win 7)	Internet Explorer (ab 6 / XP)	Java (ab 6u45, 7u25, 8b132)	OpenSSL (ab 0.9.8y, 1.0.1e)	Opera (ab 12.15 / Win 7)	Safari (ab 6/ iOS ≥ 6.0.1, OS X ≥ 10.6.8)
TLS 1.2	≥ 4.4.2	≥ 30	≥ 27	≥ 11/W7	≥ 8b132	≥ 1.0.1e	—	≥ 6
TLS 1.1	≥ 4.4.2	≥ 30	≥ 27	≥ 11/W7	≥ 8b132	≥ 1.0.1e	≥ 16	≥ 6
TLS 1.0	≥ 2.3.7	≥ 30	≥ 21	≥ 7/Vista, 8/XP	≥ 6u45	≥ 0.9.8y	≥ 12.15	≥ 6
SSL 3 INSECURE → abschalten	≥ 2.3.7	≥ 30	≥ 21	≥ 6/XP	≥ 6u45	≥ 0.9.8y	≥ 16	≥ 6
Cipher:								
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	≥ 4.4.2	—	—	MS14-066/W7	—	≥ 1.0.1e	—	—
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	≥ 4.4.2	≥ 32	—	MS14-066/W7	≥ 8b132	≥ 1.0.1e	—	—
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	≥ 4.4.2	= 30	—	—	—	≥ 1.0.1e	≥ 12.15	≥ 6
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	≥ 4.0.4	≥ 30	≥ 21	—	—	≥ 0.9.8y	≥ 12.15	≥ 6
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	≥ 4.4.2	= 30	—	—	≥ 8b132	≥ 1.0.1e	≥ 12.15	≥ 6
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	≥ 4.4.2	—	—	—	—	≥ 1.0.1e	—	—
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	≥ 4.4.2	≥ 32	≥ 27	—	≥ 8b132	≥ 1.0.1e	—	—
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	—	—	—	—	—	≥ 1.0.1e	—	≥ 6
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	≥ 4.0.4	≥ 30	≥ 24	≥ 7/Vista	—	≥ 1.0.1e	≥ 16	≥ 6
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	≥ 4.4.2	= 30	—	≥ 11/W7	≥ 8b132	≥ 1.0.1e	≥ 16	≥ 6
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	≥ 4.0.4	≥ 30	≥ 24	≥ 7/Vista	≥ 7u25	≥ 1.0.1e	≥ 16	≥ 6
TLS_RSA_WITH_AES_256_GCM_SHA384	≥ 4.4.2	—	—	MS14-066/W7	—	≥ 1.0.1e	—	—
TLS_RSA_WITH_AES_128_GCM_SHA256	≥ 4.4.2	≥ 32	—	MS14-066/W7	≥ 8b132	≥ 1.0.1e	—	—
TLS_RSA_WITH_AES_256_CBC_SHA	≥ 4.0.4	≥ 30	≥ 21	≥ 7/Vista	—	≥ 0.9.8y	≥ 12.15	≥ 6
TLS_RSA_WITH_AES_128_CBC_SHA	≥ 2.3.7	≥ 30	≥ 21	≥ 7/Vista	≥ 6u45	≥ 0.9.8y	≥ 12.15	≥ 6
TLS_RSA_WITH_3DES_EDE_CBC_SHA	≥ 2.3.7	≥ 30	≥ 21	≥ 6/XP, 8/W2003	≥ 6u45	≥ 0.9.8y	≥ 12.15	≥ 6

1. Treffer  
1. Treffer

Daten-Quellen: <https://www.ssllabs.com/ssltest/clients.html>, <https://support.microsoft.com/kb/2992611>



## Was gibt es?

- Meist Command-Line (CLI)
- Wenige mit GUI (meist nur Windows)
- Online-Tools (15+)
- Tools, die auf Schwachstellen bzw. Fehler prüfen (4)
- Tools, die angebotene Cipher prüfen (19)
- Tools, die das Zertifikat, PKI (Zertifikatskette) prüfen (5)



- <https://www.ssllabs.com/>
- <https://sslguru.com/ssl-tools/check-ssl-certificate.html>
- <http://certlogik.com/ssl-checker/>
- <http://www.sslshopper.com/ssl-checker.html>
- <https://www.howsmyssl.com/>
- <https://sslcheck.globalsign.com>
- <https://confirm.globessl.com/ssl-checker.html>
- <https://filippo.io/Heartbleed/>
- <http://possible.lv/tools/hb/>
- <https://ssl-tools.net/heartbleed-test>
- <https://www.cloudflarechallenge.com>
- <http://ccsbug.exposed/>



- **Cipher Checks**

- openssl nmap cnark.pl manyssl.pl [o-saft.pl](#) athena-ssl-cipher-check\_v062.jar sslthing.sh SSLAudit.pl ssl-cipher-check.pl ssldiagnose.exe sslmap.py sslscan ssltest.pl sslyze.py testssl.sh TestSSLServer.jar ssltlstest.exe THCSSLCheck.exe TLSSLed\_v1.3.sh

- **Schwachstellen-Test**

- beast.pl ssl-renegotiation.sh ssltest\_heartbeat.py ssl-heartbleed.sh ssl-ccs-injection.sh thc-ssl-dos [o-saft.pl](#)

- **Nur Zertifikatprüfung**

- Certtool certutil chksslkey cvt jcertchecker keytool SSLCertScanner.exe ssl-cert-check xca ([o-saft.pl](#))



# Tools

kann, kann nicht, ...



## OWASP

The Open Web Application Security Project

Anzahl der  
Cipher, die  
geprüft  
werden  
können

Cipher Checker	SSLv2	SSLv3	TLSv1	TLSv1.2
nmap	?	?	?	?
cnark.pl	6	48	48	?
manyssl.pl	?	?	?	?
SSLAudit.pl	8	27	37	38
sslmap.py	12	229	229	229
sslmap.py --fuzz	16M	65536	65536	65536
ssllscan	3	28	28	0
sslttest.pl	6	25	8	0
sslyze.py	6	-	12	?
ssldiagnose.exe	7	70	79	?
ssl-cipher-check.pl	9	133	133	33
testssl.sh	9	133	133	33
TestSSLServer.jar	3	?	9	?
THCSSLCheck.exe	8	27	27	0
<b>o-saft.pl +cipher</b>	13	133	133	37
<b>o-saft.pl +cipherall</b>	63	65536	65536	65536



# OWASP

The Open Web Application Security Project

O-Saft  
Funktionen

## OWASP – SSL Advanced Forensic Tool (O-Saft)

Builders

+

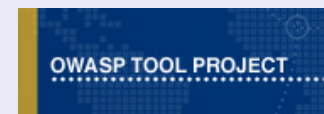
Defenders

Prüfung von:

- Angebotene Cipher(-Liste): +cipher, +cipherall
- Zertifikat, usw. +info
- Schwachstellen: +check

Auswertung und Bewertung: +check, +sni-check

Link: <https://owasp.org/index.php/O-Saft>





## Vorteile:

- Plattform-unabhängig
- Unabhängig von SSL-Bibliotheken (+cipherall)
- Wiederholbare Ergebnisse
- Automatisierbar (Script, Batch, CGI)
- Auch im Intranet (ohne Internet-Zugang) nutzbar
- Keine Root-CA und keine Client-Zertifikate nötig
- Open Source



## Zusatzfunktionen:

- Erkennt/testet beliebige Cipher (bis 65536)
- SSL-Proxy
- STARTTLS (IMAP, LDAP, POP3, RDP, SMTP, XMPP, ...)
- Erkennt serverseitige Priorisierung der Cipher
- Flexible Verwendung von TLS-Erweiterungen
- Prüfung auf bekannte Schwachstellen (BEAST, BREAST, CRIME, CSS, Heartbleed, POODLE, Renegotiation, TIME, ...)



# OWASP

The Open Web Application Security Project

O-Saft  
„Vertrauen“

## Zertifikat:

- Gültigkeit (Zeitraum, Root-CA)
- Wildcards
- Extended Validation (DV, OV, EV)
- Erweiterungen (CRL, OCSP, SRP, STS, TLS session, ..)
- Compliance (BSI TR-02102-2 , FIPS, ISM, PCI, ...)
- „Bugs“ (ungültige Zeichen, Längenrestriktionen, ...)





# OWASP

The Open Web Application Security Project

O-Saft  
Wünsche

## Was noch kommt (wenn es jemand macht):

- Scoring (mit Gewichtung)
- Zertifikatskette bis zur Root-CA
- Browsersimulation (z.B. iOS5, Android ...)
- Prüfung der TLS-Erweiterungen verbessern
- „Bugs“ → Fuzzing



# OWASP

The Open Web Application Security Project

**O-Saft**  
– Technik –

## **Folgende Seiten mit technischen Details ...**



# OWASP

The Open Web Application Security Project

## O-Saft

Interna

Client		Server	O-Saft +cipherall	+info +check
ClientHello	→		✓	✓
		ServerHello Certificate* ServerKeyExchange* CertificateRequest* ServerHelloDone	✓	✓
Certificate* ClientKeyExchange CertificateVerify* [ChangeCipherSpec] Finished	→		—	✓
	←	[ChangeCipherSpec] Finished	—	✓
Application Data	↔	Application Data	—	✓

## In der Praxis:

- Reihenfolge host, port, Optionen ist beliebig
- Erlaubt Syntax (fast) aller anderen Tools
- Erlaubt vollständige URI
- Optionen in Varianten: no-DNS, nodns, no\_DNS
- Die meisten Optionen als „on“ und „off“-Variante
- Ausgabe formatierbar
- Prüfung einzelner Werte
- Kann als CGI benutzt werden (nicht empfohlen)



## In der Praxis:

- `--no-sslv2 --noSSLv3`
- `--header --enabled`
- `--no-http --no-sni --no-dns`
- `--http --sni`
- `--trace --trace-key`
- `--cipherrange=rfc`
- `--ssl-use-reneg`
- `--sni-name=`





# OWASP

The Open Web Application Security Project

## O-Saft Beispiele (https)

### In der Praxis:

**o-saft.pl +cipherall owasp.de**

owasp.de, 443, SSLv2 (0x0002), no SNI,	,	,	
owasp.de, 443, SSLv3 (0x0300), no SNI,	,	,	
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x03000039,	DHE-RSA-AES256-SHA	,	DHE_RSA_WITH_AES_256_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x03000088,	DHE-RSA-CAMELLIA256-SHA	,	DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x03000016,	EDH-RSA-DES-CBC3-SHA	,	EDH_RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x03000033,	DHE-RSA-AES128-SHA	,	DHE_RSA_WITH_AES_128_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x0300000A,	DES-CBC3-SHA	,	RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x03000045,	DHE-RSA-CAMELLIA128-SHA	,	DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x0300000A,	DES-CBC3-SHA	,	RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv1 (0x0301), SNI, Server Order, 0x03000005,	RC4-SHA	,	RSA_WITH_RC4_128_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x03000039,	DHE-RSA-AES256-SHA	,	DHE_RSA_WITH_AES_256_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x03000088,	DHE-RSA-CAMELLIA256-SHA	,	DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x03000016,	EDH-RSA-DES-CBC3-SHA	,	EDH_RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x03000033,	DHE-RSA-AES128-SHA	,	DHE_RSA_WITH_AES_128_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x03000045,	DHE-RSA-CAMELLIA128-SHA	,	DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x0300000A,	DES-CBC3-SHA	,	RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv11 (0x0302), SNI, Server Order, 0x03000005,	RC4-SHA	,	RSA_WITH_RC4_128_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x0300009F,	DHE-RSA-AES256-GCM-SHA384	,	DHE_RSA_WITH_AES_256_GCM_SHA384
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x0300009E,	DHE-RSA-AES128-GCM-SHA256	,	DHE_RSA_WITH_AES_128_GCM_SHA256
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x0300006B,	DHE-RSA-AES256-SHA256	,	DHE_RSA_WITH_AES_256_SHA256
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000039,	DHE-RSA-AES256-SHA	,	DHE_RSA_WITH_AES_256_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000088,	DHE-RSA-CAMELLIA256-SHA	,	DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000016,	EDH-RSA-DES-CBC3-SHA	,	EDH_RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000067,	DHE-RSA-AES128-SHA256	,	DHE_RSA_WITH_AES_128_SHA256
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000033,	DHE-RSA-AES128-SHA	,	DHE_RSA_WITH_AES_128_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000045,	DHE-RSA-CAMELLIA128-SHA	,	DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x0300000A,	DES-CBC3-SHA	,	RSA_WITH_DES_192_CBC3_SHA
owasp.de, 443, TLSv12 (0x0303), SNI, Server Order, 0x03000005,	RC4-SHA	,	RSA_WITH_RC4_128_SHA
owasp.de, 443, TLSv13 (0x0304), SNI,	,	,	



# OWASP

The Open Web Application Security Project

## O-Saft Beispiele (MX-SMTP)

### In der Praxis:

```
checkAllCiphers.pl --starttls=smtp --mx owasp.org:25
```

```
#####  
# 'checkAllCiphers.pl' (part of OWASP project 'O-Saft'), Version: 2014-11-19  
# NET::SSLhello_2014.11.19  
#####
```

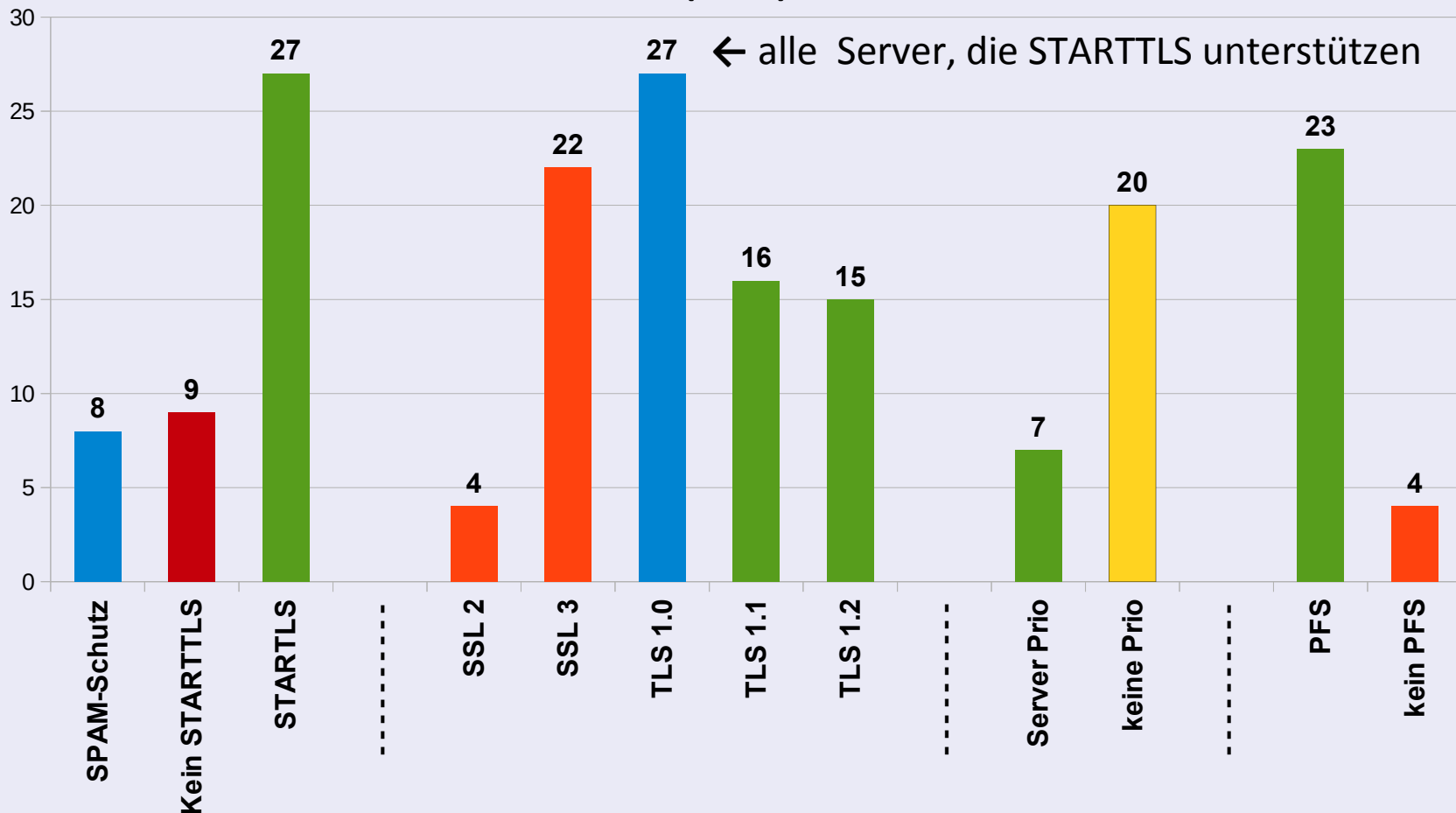
```
# get MX-Records:
```

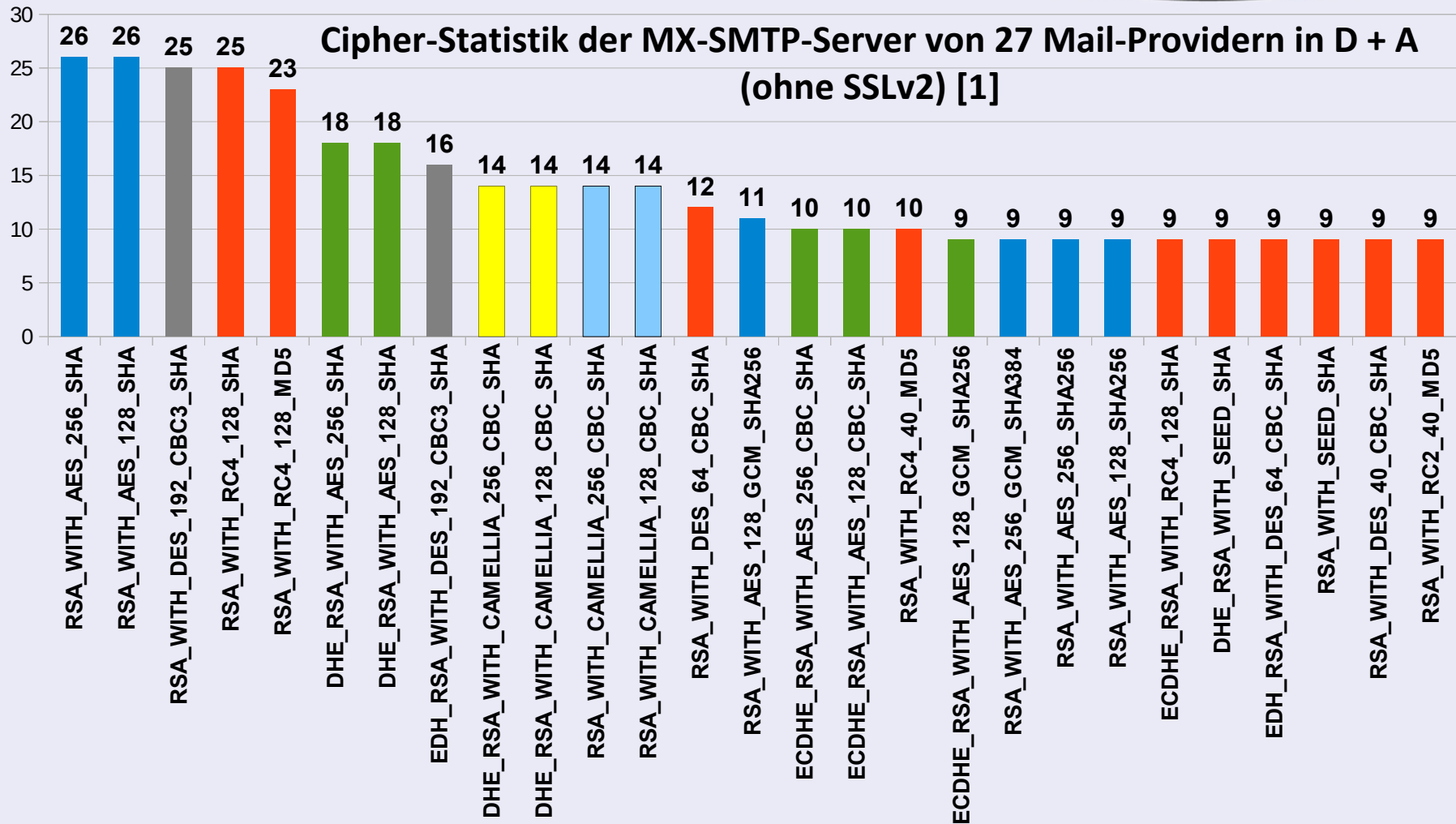
owasp.org, 25, MX	,	ASPMX4.GOOGLEMAIL.COM,	Prio,	30
owasp.org, 25, MX	,	ALT1.ASPMX.L.GOOGLE.COM,	Prio,	20
owasp.org, 25, MX	,	ASPMX3.GOOGLEMAIL.COM,	Prio,	30
owasp.org, 25, MX	,	ASPMX5.GOOGLEMAIL.COM,	Prio,	30
owasp.org, 25, MX	,	ASPMX.L.GOOGLE.COM,	Prio,	10
owasp.org, 25, MX	,	ALT2.ASPMX.L.GOOGLE.COM,	Prio,	20
owasp.org, 25, MX	,	ASPMX2.GOOGLEMAIL.COM,	Prio,	30

```
-----  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv2 (0x0002), no SNI, ,  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x0300C011, ECDHE-RSA-RC4-SHA , ECDHE_RSA_WITH_RC4_128_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x0300C013, ECDHE-RSA-AES128-SHA, ECDHE_RSA_WITH_AES_128_CBC_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x0300C014, ECDHE-RSA-AES256-SHA, ECDHE_RSA_WITH_AES_256_CBC_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x03000005, RC4-SHA , RSA_WITH_RC4_128_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x03000004, RC4-MD5 , RSA_WITH_RC4_128_MD5  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x0300002F, AES128-SHA , RSA_WITH_AES_128_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x0300000A, DES-CBC3-SHA , RSA_WITH_DES_192_CBC3_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, SSLv3 (0x0300), no SNI, Server Order, 0x03000035, AES256-SHA , RSA_WITH_AES_256_SHA  
  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x0300C011, ECDHE-RSA-RC4-SHA , ECDHE_RSA_WITH_RC4_128_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x0300C013, ECDHE-RSA-AES128-SHA, ECDHE_RSA_WITH_AES_128_CBC_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x0300C014, ECDHE-RSA-AES256-SHA, ECDHE_RSA_WITH_AES_256_CBC_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x03000005, RC4-SHA , RSA_WITH_RC4_128_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x03000004, RC4-MD5 , RSA_WITH_RC4_128_MD5  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x0300002F, AES128-SHA , RSA_WITH_AES_128_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x0300000A, DES-CBC3-SHA , RSA_WITH_DES_192_CBC3_SHA  
ASPMX4.GOOGLEMAIL.COM, 25, TLSv1 (0x0301), SNI, Server Order, 0x03000035, AES256-SHA , RSA_WITH_AES_256_SHA ...
```



**Statistik der SSL/TLS-Protokolle und Prioritätsvorgabe  
der MX-SMTP-Server von 44 (→36) Mail-Providern in D + A + CH**





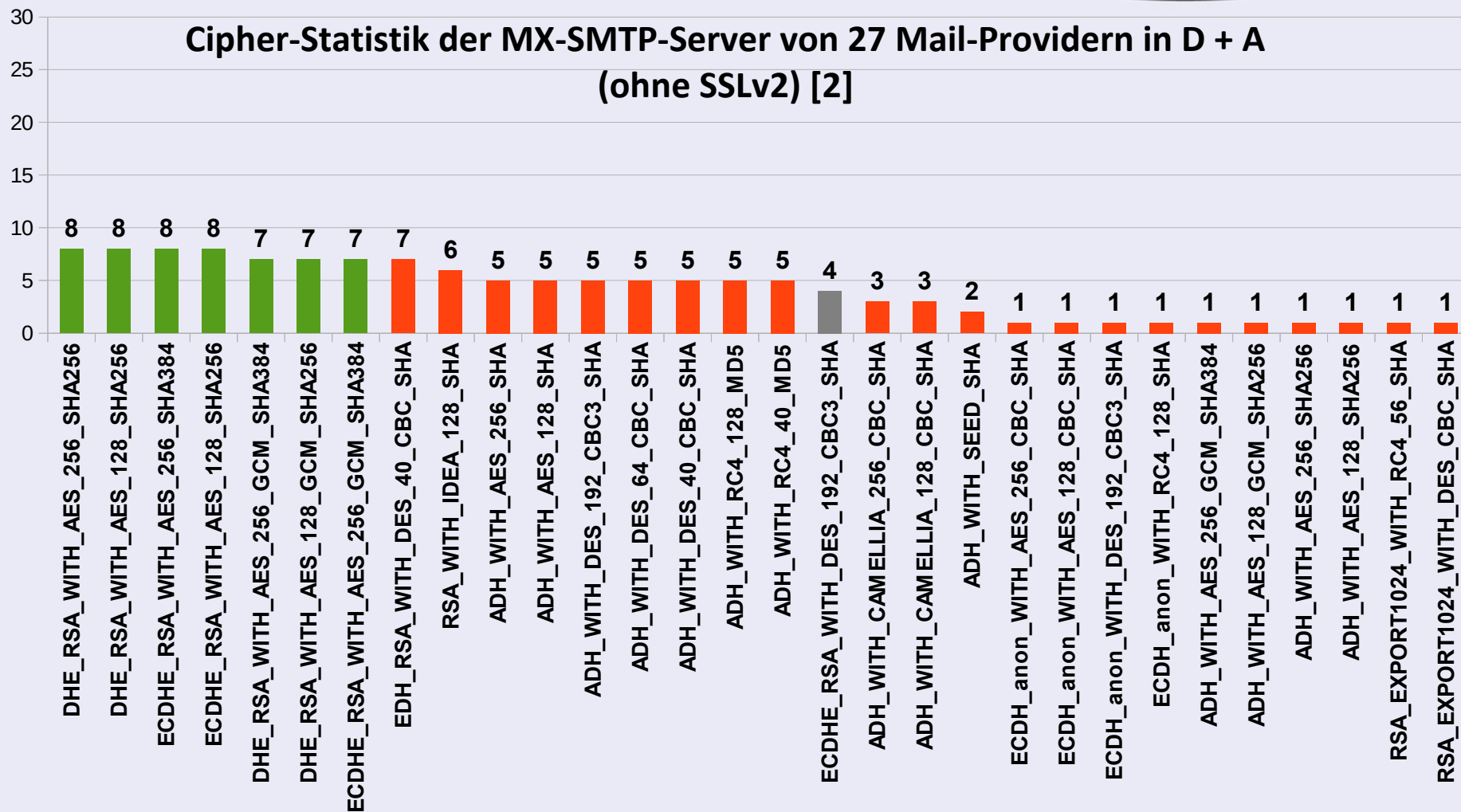


# OWASP

The Open Web Application Security Project

## O-Saft

### Beispiele: MX-SMTP-Server [3]







## Weitere Informationen:

- **OWASP – SSL Advanced Forensic Tool (O-Saft)**  
(<https://owasp.org/index.php/O-Saft>)
- **OWASP Transport Layer Protection Cheat Sheet**  
([https://owasp.org/index.php/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet](https://owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet))
- **OWASP Top 10 für Entwickler**  
([https://owasp.org/index.php/Germany/Projekte/Top\\_10\\_fuer\\_Entwickler-2013/A6-Verlust\\_der\\_Vertraulichkeit\\_sensibler\\_Daten#tab=JAVA2](https://owasp.org/index.php/Germany/Projekte/Top_10_fuer_Entwickler-2013/A6-Verlust_der_Vertraulichkeit_sensibler_Daten#tab=JAVA2))

Fragen?



**OWASP**

The Open Web Application Security Project

