Mirror mirror on the wall

Filippo De Grandi
DISI
University of Trento
Trento
filippo.degrandi@studenti.unitn.it

I. Setup

The vulnerable environment was built using a custom Docker image based on `ubuntu:18.04`. The image integrates `NGINX 1.9.0` with `OpenSSL 1.0.1u`. Several attempts were made to install `OpenSSL 1.0.1f`, but compilation errors prevented its use. The use of the latter version would have allowed testing additional vulnerabilities, such as Heartbleed.

The Docker container automates also the generation of a self-signed certificate and key with very weak security parameters.

Certificate Generation bash

```bash
RUN openssl genrsa -out /usr/local/nginx/conf/server.key 512
    openssl req -new -x509 -key /usr/local/nginx/conf/server.key -out /usr/local/nginx/conf/server.crt -days 365 -subj "/C=IT/ST=Lab/L=Lab/O=TheOrgOfItalianPasta/OU=Lab/CN=pasta@italy.gov"
    chmod 600 /usr/local/nginx/conf/server.key
    openssl dhparam -out /usr/local/nginx/conf/dhparam.pem 512
```

The relevant configuration files are:

Dockerfile

- Compiles NGINX against `OpenSSL 1.0.1u` using the `--with-openssl` flag.
- Enables deprecated options (`enable-ssl2`, `enable-des`, `enable-rc4`, `enable-weak-ssl-ciphers`) to ensure weak cipher suites are available.
- Generates a self-signed certificate and key (512-bit RSA) with a very low security margin.
- The container exposes port `4444` for `HTTPS`.

nginx.conf

- Enables insecure SSL/TLS versions: `SSLv2`, `SSLv3`, `TLSv1`, and `TLSv1.2`.
- Sets weak cipher suites, explicitly including `RC4`, `DES`, and `aNULL` ciphers.
- Disables `HSTS` and certificate transparency.
- Uses `512-bit` DH parameters, allowing weak ephemeral key exchanges.

These configurations were intentionally designed to make the server vulnerable to a wide range of TLS attacks.

II. Analysis with TLSAssistant

The `Nginx` webserver was analyzed through the Docker version of `TLSAssistant`, yielding two different reports:

- `full.pdf`: Complete analysis across all modules.
- `mitzvah_nomore.pdf`: Focused analysis of RC4-related vulnerabilities.

The division in two different analysis comes from the fact that the RC4 vulnerabilities are (strangely) only found if analyzed independently from the other modules. They have been merged into one single `merged.pdf` for the sake of this report.

III. Results Overview

TLSAssistant identified the server as potentially vulnerable to a large number of attacks, including:

Category	Vulnerabilities	Root Cause
Protocol Weaknesses	SSLv2, SSLv3 support; BEAST [1]; DROWN [2]	Deprecated TLS versions enabled
Cipher Weaknesses	Bar Mitzvah [3], RC4 NOMORE [4], Sweet32 [5]	Use of RC4 and DES ciphers
Compression-related	BREACH [6]	TLS compression and gzip active
Session Issues	3SHAKE [7]	Missing `extended_master_secret`
HSTS / HTTPS Misconfigurations	HSTS not set / HTTPS not enforced	Missing security headers
Configuration Issues	ALPACA [8]	Different protocols & Multi-domain / Wildcard certificates

The Recap sections in `merged.pdf` shows most modules (e.g., BEAST, 3SHAKE, DROWN, BREACH) marked as Potentially Vulnerable. `mitzvah_nomore.pdf` confirms Bar Mitzvah (CVE-2015-2808) and RC4 NOMORE vulnerabilities caused by the enabled RC4 cipher.

References

[1] “BEAST attack.” [Online]. Available: https://en.wikipedia.org/wiki/Transport_Layer_Security#BEAST_attack

[2] “DROWN attack.” [Online]. Available: https://en.wikipedia.org/wiki/DROWN_attack

[3] “Mitzvah attack.” [Online]. Available: https://en.wikipedia.org/wiki/Bar_mitzvah_attack

[4] “No More MITM attack.” [Online]. Available: https://www.rc4nomore.com/

[5] “SWEET32 attack.” [Online]. Available: https://sweet32.info/

[6] “BREACH attack.” [Online]. Available: https://www.infosecinstitute.com/resources/hacking/the-breach-attack/

[7] “3Shake attack.” [Online]. Available: https://blog.cryptographyengineering.com/2014/04/24/attack-of-week-triple-handshakes-3shake/

[8] “ALPACA attack.” [Online]. Available: https://alpaca-attack.com/