Third party memo

Confidential

Customer	PeeringDB
Product	PeeringDB application
Version	1.0
Author	Computest
Date	April 5, 2018

Computest

Introduction

PeeringDB is a popular web service for managing information about interconnection policies between networks. The foundation behind PeeringDB plans to open-source the application and has asked Computest to perform a security audit. The audit aims to find all vulnerabilities in the existing code base, so that open sourcing the code base will not result in great added security risk.

Approach and scope

Approach

Computest has performed a manual code audit on the source code of the PeeringDB application. The code audit is performed using the following checklist:

- Certified Secure Web Application Secure Development Checklist v4.2

The reviewer does not go through the code base item by item, but he does make sure that at the end of the audit, each item on the checklist has been checked and can be answered with Pass / Fail / Not applicable. While this approach serves to make the audit as exhaustive and reproducible as possible, it is still performed within the error margins of manual inspection.

When a vulnerability is found, Computest will assign the vulnerability a vulnerability score. To determine the vulnerability score, the Common Vulnerability Scoring System (CVSS) version 2 is used. Detailed information about this scoring system can be found at www.first.org/cvss/. Each vulnerability is rated with a numeric score ranging from 1 to 10, with 10 being the highest score making the vulnerability the most severe.

Scope

The code-audit was conducted on the source code of the PeeringDB application. The test was conducted against version 2.8.4.

```
peeringdb version 2.8.4
commit id: b216acb3ed561ae32f6362eea850a1f35eac1e19
```

A number of external libraries form an integral part of PeeringDB and have therefore been included in the review:

```
django-peeringdb
https://github.com/peeringdb/django-peeringdb/
commit id: b68877f2111042bf61da6a0b77bc646c9305205f
```

```
peeringdb-py version 0.5.1 (downloaded via pip)
https://pypi.python.org/pypi/peeringdb
sha256: 0b3a3ce2c6ea8490fdc43a2a51d9b1fbe6d85093a5f7e1b09e3ea131f6eb130e
```

```
django-namespace-perms version 0.5.0 (downloaded via pip)
https://pypi.python.org/pypi/django-namespace-perms
sha256: b8174c02d4f219d5a4b43c956f42315057709148e4b01f59008638c2db950399
```

```
django-handleref version 0.2.0 (downloaded via pip)
https://pypi.python.org/pypi/django-handleref
sha256: 5fa95eb4589250526c23843e06aa0372c7f267b4456e949c8dc157df54792b9c
```

The following libraries form an important part of PeeringDB, but have not been reviewed:

- Xbahn (replication library);
- Facsimile implementation (configuration management). Computest has reviewed PeeringDB's server configuration files at best effort.

Reproduction of issues was conducted against the following URL:

- https://beta.peeringdb.com/

Result summary

In general Computest is of the opinion that the code base is well structured, documented and well protected against the most common types of attacks. The choice of framework (Django) contributes to this fact, as it is known for its built-in security features and structured design.

During the review, several vulnerabilities were discovered that were fixed by the PeeringDB team and re-checked by Computest. The remaining results are discussed here.

Computest identified several vulnerabilities that could be clasified as low-risk. The vulnerabilities do not have a structural nature, and as a result mitigation should be possible with minor adjustments.

Three of the five discovered vulnerabilities concern lack of or insufficient brute-force protection mechanisms. The other two are defence-in-depth measures that would add to the security quality, but lack of which does not pose a direct threat.

Considering the low impact of the remaining vulnerabilities, Computest is of the opinion that the PeeringDB application is of sufficient security quality.