```
safety check -r requirements.txt --full-report
Warning: unpinned requirement 'freezegun' found in requirements.txt,
unable to check.
╒══════════════════════════════════════════════════════════════════════
═════════╕
│
│                               /$$$$$$            /$$
│                              /$$__  $$          | $$
│           /$$$$$$$  /$$$$$$ | $$  \__//$$$$$$  /$$$$$$   /$$   /$$
│          /$$_____/ |____  $$| $$$$   /$$__  $$|_  $$_/  | $$  | $$
│         |  $$$$$$   /$$$$$$$| $$_/  | $$$$$$$$  | $$    | $$  | $$
│          \____  $$ /$$__  $$| $$    | $$_____/  | $$ /$$| $$  | $$
│          /$$$$$$$/|  $$$$$$$| $$    |  $$$$$$$  |  $$$$/|  $$$$$$$
│         |_______/  \_______/|__/     \_______/   \___/   \____  $$
│                                                          /$$  | $$
│                                                         |  $$$$$$/
│  by pyup.io                                              \______/
│
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ REPORT
│
│ checked 36 packages, using default DB
╞════════════════════════════╤═══════════╤══════════════════════════╤══
═════════╡
│ package                    │ installed │ affected                 │
ID       │
╞════════════════════════════╪═══════════╪══════════════════════════╪══
═════════╡
│ django                     │ 1.7.9     │ <1.7.11                  │
25714    │
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ The get_format function in utils/formats.py in Django before 1.7.x
before    │
│ 1.7.11, 1.8.x before 1.8.7, and 1.9.x before 1.9rc2 might allow
```

```
remote       │
│ attackers to obtain sensitive application secrets via a settings key
in      │
│ place of a date/time format setting, as demonstrated by SECRET_KEY.
│
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ django                       │ 1.7.9     │ <1.8.10                    │
33074    │
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ The password hasher in contrib/auth/hashers.py in Django before
1.8.10 and   │
│ 1.9.x before 1.9.3 allows remote attackers to enumerate users via a
timing   │
│ attack involving login requests.
│
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ django                       │ 1.7.9     │ <1.8.10                    │
33073    │
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ The utils.http.is_safe_url function in Django before 1.8.10 and
1.9.x before │
│ 1.9.3 allows remote attackers to redirect users to arbitrary web
sites and   │
│ conduct phishing attacks or possibly conduct cross-site scripting
(XSS)     │
│ attacks via a URL containing basic authentication, as demonstrated
by        │
│ http://mysite.example.com\@attacker.com.
│
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ django                       │ 1.7.9     │ <1.8.15                    │
25718    │
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ The cookie parsing code in Django before 1.8.15 and 1.9.x before
1.9.10,     │
│ when used on a site with Google Analytics, allows remote attackers
to bypass │
│ an intended CSRF protection mechanism by setting arbitrary cookies.
│
╞══════════════════════════════════════════════════════════════════════
═════════╡
│ django                       │ 1.7.9     │ >=1.7,<1.7.10              │
25728    │
╞══════════════════════════════════════════════════════════════════════
```

```
══════════╡
│ The (1) contrib.sessions.backends.base.SessionBase.flush and (2)
│
│ cache_db.SessionStore.flush functions in Django 1.7.x before 1.7.10,
1.4.x    │
│ before 1.4.22, and possibly other versions create empty sessions in
certain  │
│ circumstances, which allows remote attackers to cause a denial of
service     │
│ (session store consumption) via unspecified vectors.
│
╞════════════════════════════════════════════════════════════════════
══════════╡
│ django                        │ 1.7.9     │ >=1.7,<1.7.10            │
25727    │
╞════════════════════════════════════════════════════════════════════
══════════╡
│ contrib.sessions.middleware.SessionMiddleware in Django 1.8.x before
1.8.4,   │
│ 1.7.x before 1.7.10, 1.4.x before 1.4.22, and possibly other
versions allows │
│ remote attackers to cause a denial of service (session store
consumption or  │
│ session record removal) via a large number of requests to
│
│ contrib.auth.views.logout, which triggers the creation of an empty
session    │
│ record.
│
╞════════════════════════════════════════════════════════════════════
══════════╡
│ werkzeug                      │ 0.9.6     │ <0.11.11                 │
35661    │
╞════════════════════════════════════════════════════════════════════
══════════╡
│ Cross-site scripting (XSS) vulnerability in the render_full function
in       │
│ debug/tbtools.py in the debugger in Pallets Werkzeug before 0.11.11
(as used │
│ in Pallets Flask and other products) allows remote attackers to
inject       │
│ arbitrary web script or HTML via a field that contains an exception
message. │
╞════════════════════════════════════════════════════════════════════
══════════╡
│ ipython                       │ 2.2.0     │ <3.2.2                   │
33132    │
╞════════════════════════════════════════════════════════════════════
══════════╡
│ Cross-site scripting (XSS) vulnerability in the file browser in
```

```
│
│ notebook/notebookapp.py in IPython Notebook before 3.2.2 and Jupyter
│
│ Notebook 4.0.x before 4.0.5 allows remote attackers to inject
arbitrary web  │
│ script or HTML via a folder name.  NOTE: this was originally
reported as a   │
│ cross-site request forgery (CSRF) vulnerability, but this may be
inaccurate. │
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ ipython                    │ 2.2.0     │ <3.2.2                   │
33133    │
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ The editor in IPython Notebook before 3.2.2 and Jupyter Notebook
4.0.x       │
│ before 4.0.5 allows remote attackers to execute arbitrary JavaScript
code    │
│ via a crafted file, which triggers a redirect to files/, related to
MIME     │
│ types.
│
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ newrelic                   │ 2.28.0.26 │ >=1.1.0.192,<=2.106.0.87 │
35805    │
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ New Relic agents run explain plans for Slow Transaction Traces and
Slow SQL  │
│ Queries. Previous versions of the agents would run an explain plan
on the    │
│ SQL query by prepending the query with explain. This may cause an
issue when │
│ there are multiple statements separated by semicolons in a single
query. The │
│ first statement in the string returns its explain plan, but any
subsequent   │
│ statement after that may execute as a general SQL statement.
Depending on    │
│ the language, library, and database, the agent may return the
results of the │
│ additional statements to New Relic. It is also possible that the
additional  │
│ statements could execute an additional INSERT or UPDATE command.
With this   │
│ security update, New Relic agents will no longer run explain plans
on any    │
│ query that contains a semicolon as a statement separator.
```

```
│
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ requests                   │ 2.4.0     │ <2.6.0                   │
26102    │
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ requests 2.6.0 fixes handling of cookies on redirect. Previously a
cookie    │
│ without a host value set would use the hostname for the redirected
URL       │
│ exposing requests users to session fixation attacks and potentially
cookie   │
│ stealing.
│
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ requests                   │ 2.4.0     │ <=2.19.1                 │
36546    │
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ The Requests package before 2.19.1 sends an HTTP Authorization
header to an  │
│ http URI upon receiving a same-hostname https-to-http redirect,
which makes  │
│ it easier for remote attackers to discover credentials by sniffing
the       │
│ network.
│
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ requests                   │ 2.4.0     │ >=2.1,<=2.5.3            │
26103    │
╞═════════════════════════════════════════════════════════════════════
═════════╡
│ The resolve_redirects function in sessions.py in requests 2.1.0
through      │
│ 2.5.3 allows remote attackers to conduct session fixation attacks
via a      │
│ cookie without a host value in a redirect.
│
╘═════════════════════════════════════════════════════════════════════
═════════╛
```