3.0.3 Release Notes

Channels 3.0.3 fixes a security issue in Channels 3.0.2

CVE-2020-35681: Potential leakage of session identifiers using legacy `AsgiHandler`

The legacy `channels.http.AsgiHandler` class, used for handling HTTP type requests in an ASGI environment prior to Django 3.0, did not correctly separate request scopes in Channels 3.0. In many cases this would result in a crash but, with correct timing responses could be sent to the wrong client, resulting in potential leakage of session identifiers and other sensitive data.

This issue affects Channels 3.0.x before 3.0.3, and is resolved in Channels 3.0.3.

Users of `ProtocolTypeRouter` not explicitly specifying the handler for the `'http'` key, or those explicitly using `channels.http.AsgiHandler`, likely to support Django v2.2, are affected and should update immediately.

Note that both an unspecified handler for the `'http'` key and using `channels.http.AsgiHandler` are deprecated, and will raise a warning, from Channels v3.0.0

This issue affects only the legacy channels provided class, and not Django's similar `ASGIHandler`, available from Django 3.0. It is recommended to update to Django 3.0+ and use the Django provided `ASGIHandler`.

A simplified `asgi.py` script will look like this:

```
import os

from django.core.asgi import get_asgi_application

# Fetch Django ASGI application early to ensure AppRegistry is populated
# before importing consumers and AuthMiddlewareStack that may import ORM
# models.
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "mysite.settings")
django_asgi_app = get_asgi_application()

# Import other Channels classes and consumers here.
from channels.routing import ProtocolTypeRouter, URLRouter

application = ProtocolTypeRouter({
    # Explicitly set 'http' key using Django's ASGI application.
    "http": django_asgi_app,
    ),
})
```

Please see Deploying for a more complete example.