Research of CVE-2016-9014 vulnerability

Author: Siim Raudsepp
Introduction

CVE-2016-9014 is a Django web framework bug that affected versions 1.8.x before 1.8.16, 1.9.x before 1.9.11 and 1.10.x before 1.10.3. It allowed remote attackers to conduct DNS rebinding attacks by leveraging failure to validate the HTTP Host header against settings.ALLOWED_HOSTS when the DEBUG flag was set to true. Since Django doesn’t have modules that allow remote code execution, this attack could give the attacker the ability to execute SQL code on the database or conduct further XSS attacks (a type of injection, in which the website doesn’t validate form data and thus the attacker can infect the server with malicious scripts).

The exploit

DNS rebinding is a type of computer hack. In the attack, the attacker registers a domain and redirects it to a DNS server under the attackers control. The DNS server is configured to respond with a short TTL record, meaning that data is kept in the network for a very short time, which in turn prevents the browser from caching the response. When a victim enters the attackers site, the server sends the malicious scripts to the user, which initiates a new connection to the attackers site. This connection provides a correct IP address to the attackers domain and the victim’s private IP address. Now the malicious script can gain network-level access to the victim’s computer, just by knowing the victim’s private IP address.

In Django, this exploit could be used to obtain access to the web application under attack. It is done by sending a fake Host header to the web application. This makes the web application think that the request is safe and accept the request, which contains malicious code. The code can then access the cookies of any victim that connects to the site.
The fix

The original part of the code, that allowed DNS rebinding attacks by not validating the hostname when DEBUG was True.

```python
"""Return the HTTP host using the environment or request headers."
host = self._get_raw_host()

# There is no hostname validation when DEBUG=True
if settings.DEBUG:
    return host

domain, port = split_domain_port(host)
if domain and validate_host(domain, settings.ALLOWED_HOSTS):
    return host
else:
    msg = "Invalid HTTP_HOST header: %r." % host
```

The fix, which adds localhost to allowed_hosts by default:

```python
"""Return the HTTP host using the environment or request headers."
host = self._get_raw_host()

# Allow variants of localhost if ALLOWED_HOSTS is empty and DEBUG=True.
allowed_hosts = settings.ALLOWED_HOSTS
if settings.DEBUG and not allowed_hosts:
    allowed_hosts = ['localhost', '127.0.0.1', '[:1]']

domain, port = split_domain_port(host)
if domain and validate_host(domain, allowed_hosts):
    return host
else:
    msg = "Invalid HTTP_HOST header: %r." % host
```

The fix is sufficient, because now Django will always validate host, before returning it.
References

3. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-9014
6. https://github.com/django/django/commit/7fe2d8d940fddddd1a02c4754008a27060c4a03e9