EP3 Redes: DOS por requests

Renan Ribeiro, João Paulo

2023-12-4

Plone

Content Managment Systems

- Software responsável por armazenar e editar “conteúdo”
- “conteúdo” sendo normalmente documentos/arquivos e dados importantes usados para alguma organização
- Normalmente precisa restrito e seguro, usado em intranets

Sobre o Plone

- Utiliza o modelo frontend $\rightarrow$ backend
- Backend implementado em python usando um servidor de aplicação chamado Zope
- Frontend implementado em react
- Backend permite administração avançada
- Backend também providencia uma API pública HTTP em JSON (i.e REST)

O escopo do Plone

O plone é usado em diversas organizações importantes:

- U.S. Federal Bureau of Investigation (FBI)
- A ONU
- O estado de New South Wales na Australia
- Agência Europeia de Ambiente
- Governo Brasileiro

A Vulnerabilidade

A Vulnerabilidade

O backend do plone precisa diferenciar entre requests normais e requests da API REST, e usa o identificador `++api++` na URL para diferenciar as da API REST:

- `http://backend:8000/++api++`

  Link de API, servidor devolve JSON

- `http://backend:8000/plone-add-website`

  Link normal, servidor devolve HTML

A Vulnerabilidade

Porém o backend não sabe lidar com URLS onde `++api++` aparece múltiplas vezes:

- `http://backend:8000/++api++/++api++/++api++`

Em particular, a performance é abismal quando o número de `++api++`s é alto, permitindo uma negação de serviço com facilidade, **sem nenhum privilégio necessário**.

Tempo de resposta

Número de ++api++s	Tempo de resposta
1	0.141s
4	0.175s
8	0.2s
16	0.8s
20	4s
22	10s
24	24s
26	1m 30s

A Vulnerabilidade

O reconhecimento de `++api++` do backend é implementado no servidor Zope e configurado no Plone, e a causa específica da performance degradada ainda não foi documentada.

Entretanto, o código que permite que múltiplos `++api++` existam é conhecido e foi tratado

O Exploit

O código problemático

```
    """
    Traversal adapter for the ``++api++`` namespace.
    It marks the request as API request.
    """

    def traverse(self, name_ignored, subpath_ignored):
        mark_as_api_request(self.request, "application/json")
        return self.context
```

A solução

```
def traverse(self, name_ignored, subpath_ignored):
    name = "/++api++"
    url = self.request.ACTUAL_URL
    if url.count(name) > 1:
        # Redirect to proper url.
        while name + name in url:
            url = url.replace(name + name, name)
        if url.count(name) > 1:
            # Something like:
            # .../++api++/something/++api++
            # Return nothing, so a NotFound is raised.
            return
        # Raise a redirect exception to stop
        # execution of the current request.
        raise Redirect(url)
    mark_as_api_request(self.request, "application/json")
    return self.context
```

The End