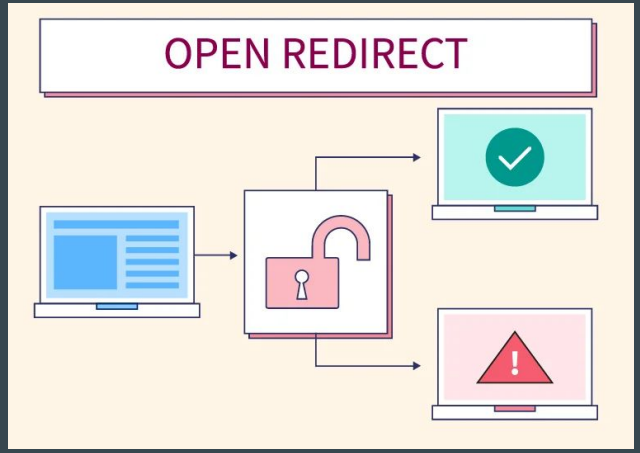
Open Redirect

Seguridad e Integridad de Sistemas



Prof. GIANNI, Fernando Julian

¿Qué es una vulnerabilidad Open Redirect?

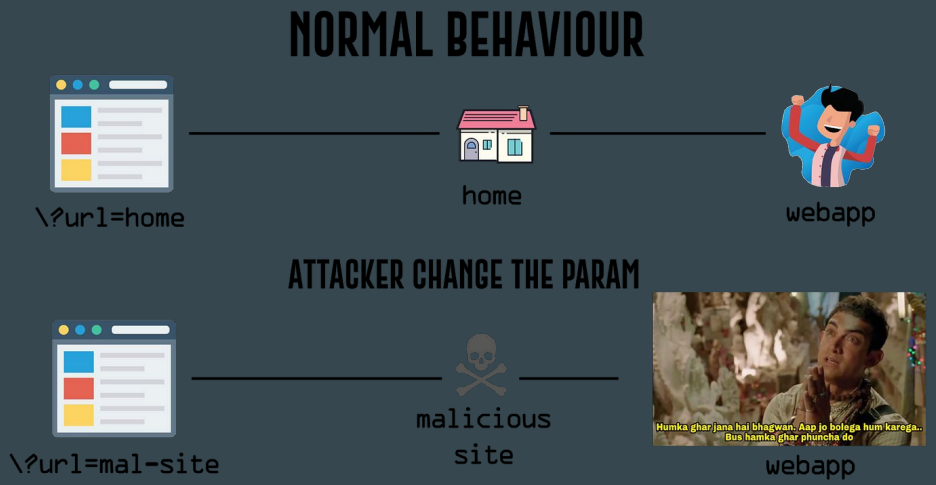
Esta vulnerabilidad Client-Side se produce cuando una aplicación utiliza datos controlados por el usuario para redireccionar a otro sitio. Esto le permite a un atacante realizar ataques de phishing e ingeniería social.

Ej (válido):

https://google.com.ar/login?url=https://maps.google.com

Ej (malicioso):

https://google.com.ar/login?url=https://evil.site.com.ar



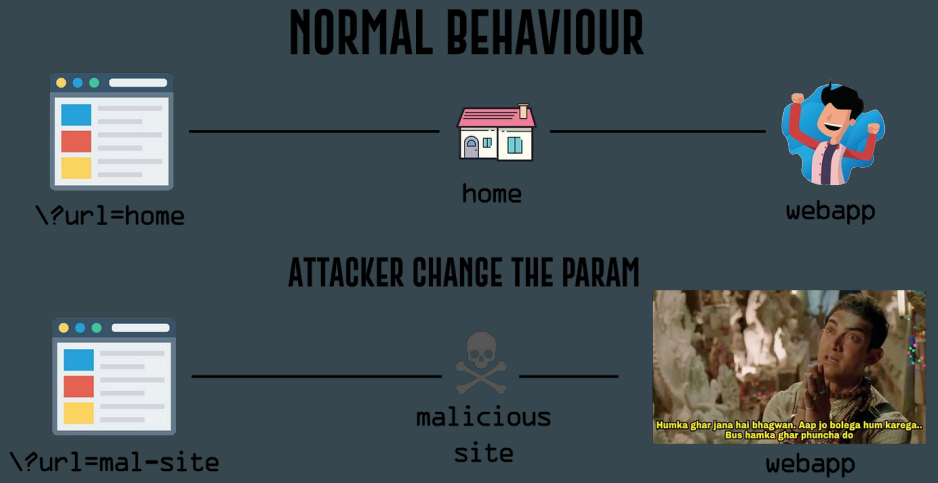
¿Cómo se explota?

Métodos de explotación:

Phishing: Los atacantes pueden redirigir a los usuarios a sitios web falsos para robar información confidencial.

Malware: Los usuarios pueden ser redirigidos a sitios que distribuyen malware, comprometiendo la seguridad de sus dispositivos.

Suplantación de identidad: Los atacantes pueden manipular las URL de redirección para realizar acciones no autorizadas en nombre del usuario.



¿Cómo encontrala?

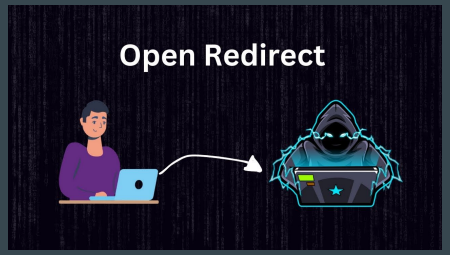
1) Revisión de logs

```
where ds >= '2023-08-04T11_00_00'
and ds <= '2023-08-04T11_10_00'
and http_x_public = 'true'
and request_method = 'GET'
AND cast(status AS varchar) LIKE '30%'
and REGEXP_LIKE(
request_uri,
'(%253D|%3D|=)http(s|)(%253A|:)(%252F|%2F|\\/)(%252F|%2F|\\/)'
)
```

2)

```
?url={payload}
?target={payload}
?destination={payload}
?redir={payload}
?redirect={payload}
/redirect/{payload}
/coi-bin/redirect.cgi?{payload}
/out?{payload}
?view={payload}
/login?to={payload}
?returnTo={payload}
?checkout_url={payload}
```

Mitigación vulnerable




```
  if (req.query.redirectToWhatsapp && req.query?.channel_url) {
    const { channel_url: channelUrl = '' } = req.query;
-   res.redirect(channelUrl ? channelUrl.replace('/list', '/whatsApp') : '/cx/webchat/whatsApp');
-   return;
+   const isMeliUrl = channelUrl => domains.some(e => channelUrl.includes(e.domain));
+   if(isMeliUrl(channelUrl)) {
+     res.redirect(channelUrl ? channelUrl.replace('/list', '/whatsApp') : '/cx/webchat/whatsApp');
+     return;
+   }
  }
  next();
```

Explotación

/webpage?url=https://mercadolibre.com.br@google.com

/webpage?url=https://google.com/?queryparam=mercadolibre.com.br

/webpage?url=https://mercadolibre.com.br.evil-site.com

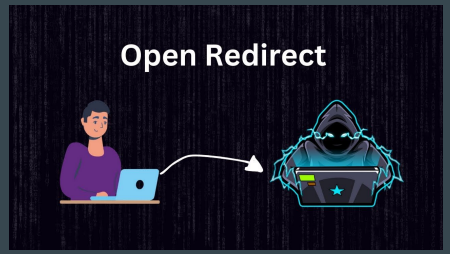
Mitigación vulnerable

```
try {
    final URL url = new URL(value);
    final String host = url.getHost().toLowerCase(locale.ROOT);
    return new ReplicationCallback(host, value);
    } catch (MalformedURLException e) {
        throw new InvalidReplicationCallbackException(e, "unsafe_callback", value);
}
```

Explotación

```
-rerab9bue9trb1pfsdjs9brmad2p6rf?go=https://google.com\\.mercadolivre.com/
```

Mitigación




```
const router = require('ragnar').router();

const preventOpenRedirect = require('nordic/prevent-open-redirect');

const customConfig = {
    params: ['q', 'go'],
    handleInvalidRedirectParam: (req, res, next) => {
        const invalidParams = res.locals.invalidParams;
        const siteId = req.platform.siteId;

        res.redirect('https://www.mercadolibre.com.ar/error?id=${siteId}');
},
whitelist: ['https://www.microsoft.com', 'https://www.google.com'],
protocols: ['https:'],
};

router.get('/login', preventOpenRedirect(customConfig), (req, res) => {
    res.send('Hello, world!');
});

module.exports = router;
```

Laboratorio

https://portswigger.net/web-security/dom-based/open-redirection/lab-dom-open-redirection

Impacto

- Phishing y suplantación de identidad.
- Ataques de malware.
- Manipulación de URL.
- Daño a la reputación.
- Pérdida de datos sensibles.

Conclusiones

- No confiar en datos provenientes del usuario para realizar redirecciones.
- En caso de necesitarlo, validar con listas blancas que el dominio ingresado sea permitido por la aplicación.