

Do Not Trust Me Using Malicious IdPs for Analyzing and Attacking Single Sign-On

MAR 10TH, 2016

[论文下载](#)

Abstract & Introductcion

- 提出一种新的方法来分析SSO协议——引入一个恶意IdP。
- 使用这种方法发现了针对OpenID的四类攻击。
- 对OpenID的安全性进行系统化分析，11/16个实现都存在漏洞，包括SourceForge、Drupal、ownCloud等。
- 开发了自动化分析工具OpenID Attacker。

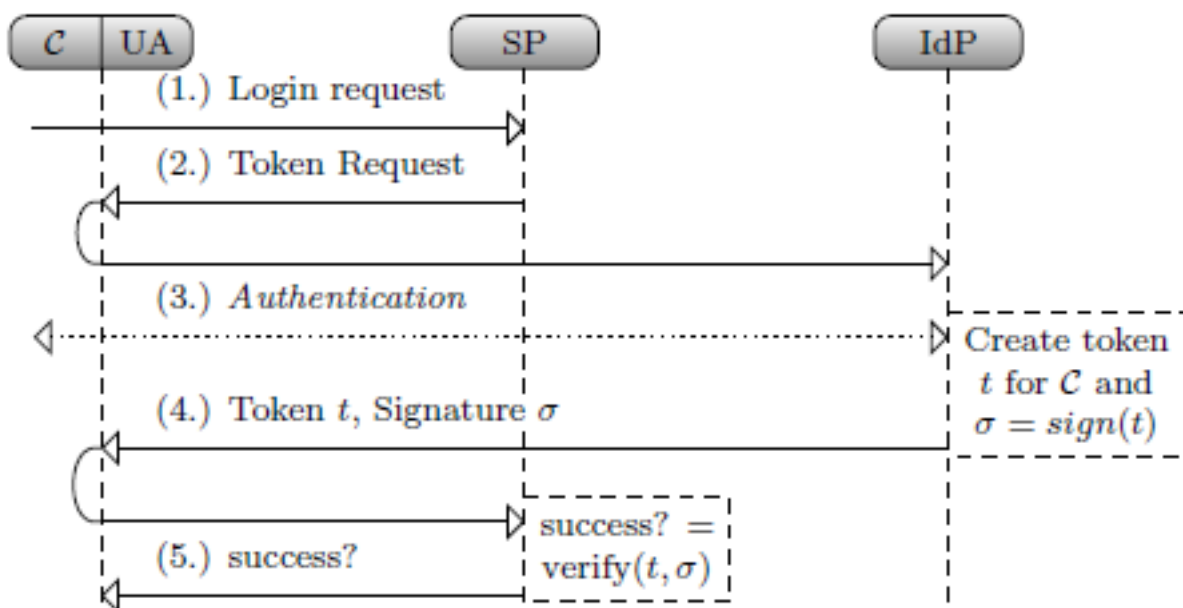


Fig. 1: Single Sign-On (SSO) overview.

Computational and Security Model

- Computational Model: OpenID中有一种开放信任关系，SP信任任意IdP创建的token，只要这个IdP在client提供的URL.IDc检索到的文档中。

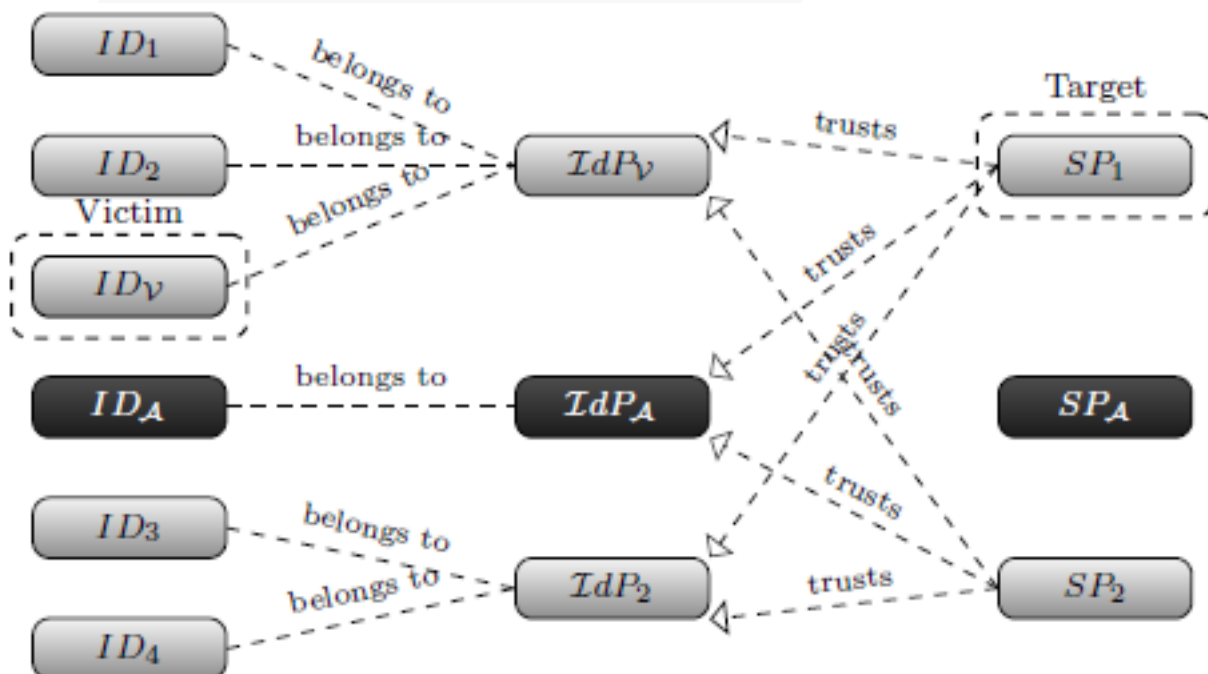


Fig. 3: SSO in the real world involves multiple clients, multiple IdPs and multiple SPs. SP_1 can even *trust* tokens of IdP_A , but only for its corresponding clients, i.e. ID_A .

- SSO Attacker Paradigm: 攻击者的目标是获取他无权访问的资源（主要是存在SP上的资源）。攻击者可以扮演恶意client/SP/IdP的角色。

OpenID: Technical Background

- OpenID中，client的一个身份是用一个URL表示的，定义为URL.ID_C，相应的URL.IdP_C、URL.SP。
- OpenID协议由3个阶段组成：（1）Discovery；（2）Association；（3）Token processing。

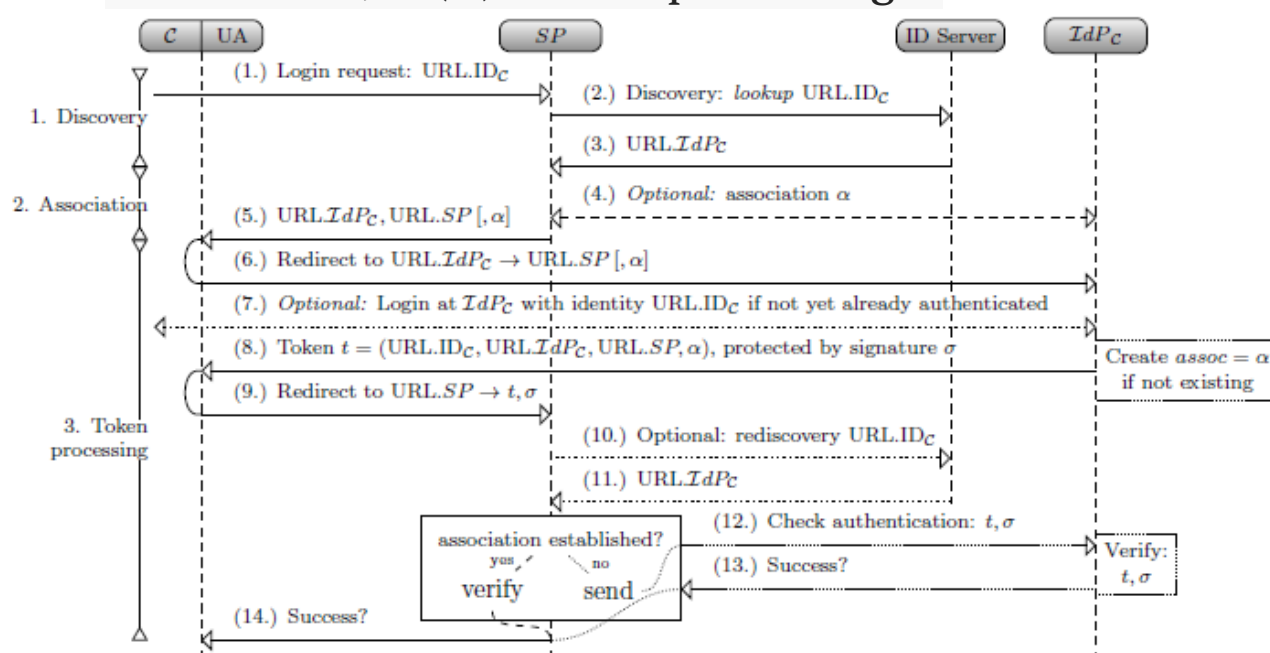


Fig. 4: The OpenID protocol flow.

```

<html><head><title>
<link rel="openid2.provider"
      href="https://myidp.com/" />
</head><body></html>

```

Listing 1.1: Minimal HTML discovery document.

Novel Attacks

- Token Recipient Confusion (TRC): 缺乏对URL.SP参数的验证。（这个攻击算不上是新发现的攻击，在其它

SSO中都有存在，该攻击能够实现还有一个因素是OpenID提供两种方法校验签名)

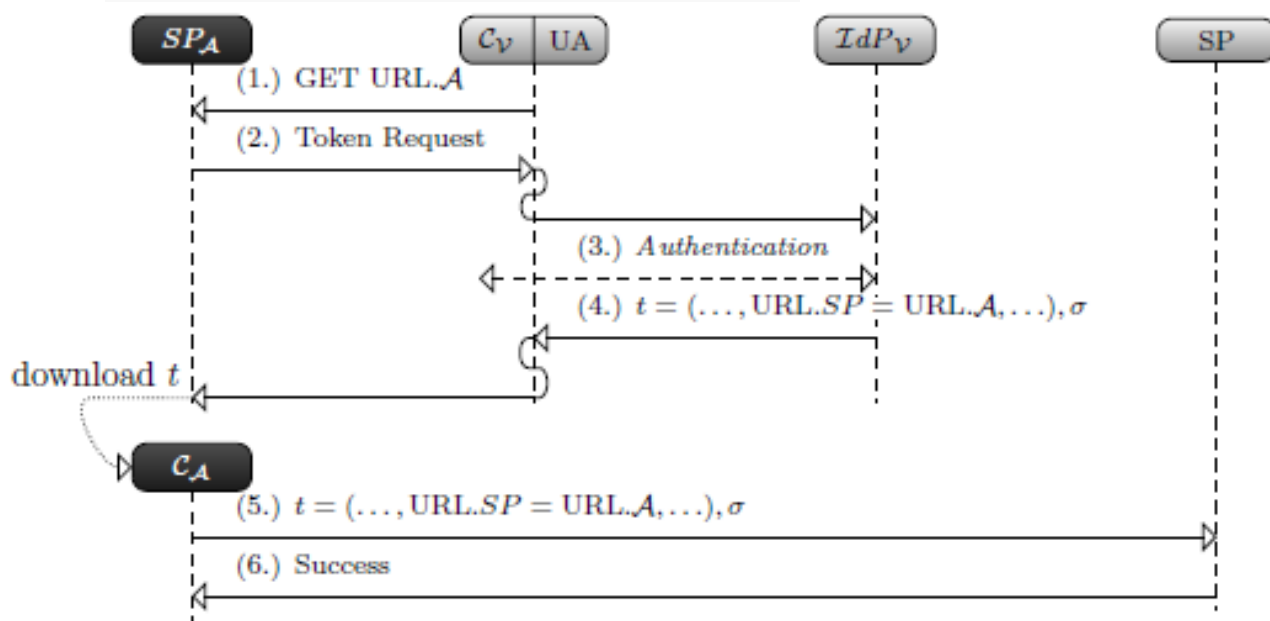


Fig. 5: Token Recipient Confusion Attack.

- Key Confusion (KC)：强制令目标SP使用攻击者选择的key去校验一个伪造token。（精巧的攻击）

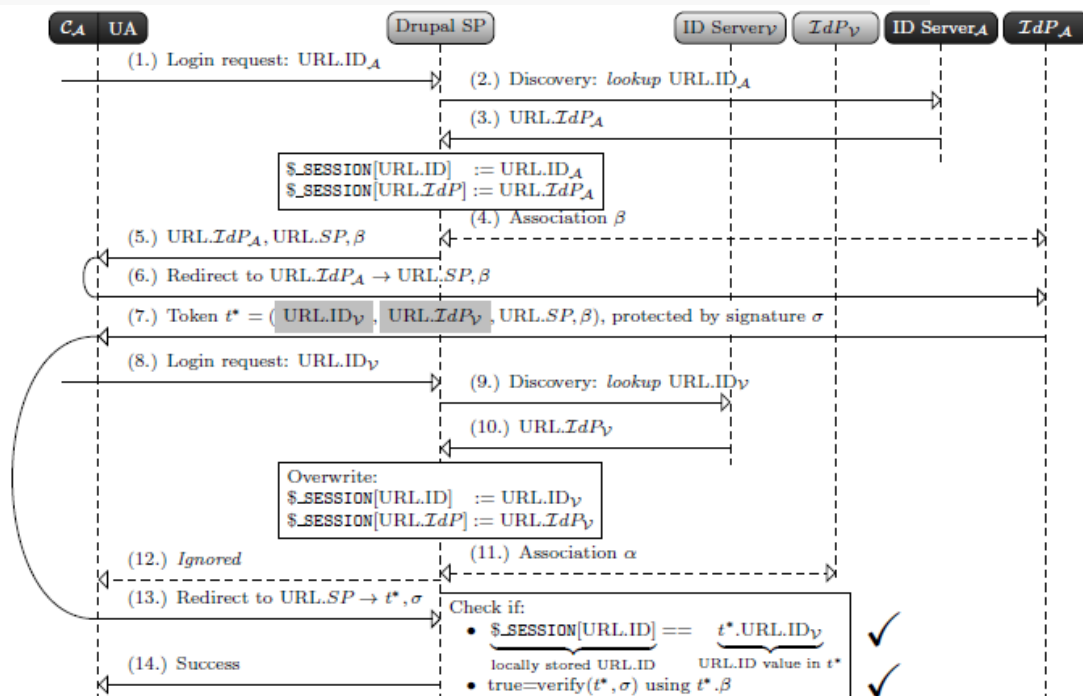


Fig. 7: Key Confusion attack on Drupal: Before the token t^* in Step (7.) is forwarded to Drupal in Step (13.), the attacker C_A starts a second login request in Step (8.) using the victim's identity $URL.ID_V$. This overwrites the $URL.ID$ and $URL.IdP$ data stored in $\$_SESSION$ and prevents the second discovery.

- ID Spoofing (IDS) : 恶意IdP可以创建一个token包含受害者的id。
- Discovery Spoofing (DS) : OpenID特性产生，略。
(攻击条件复杂)

Methodology

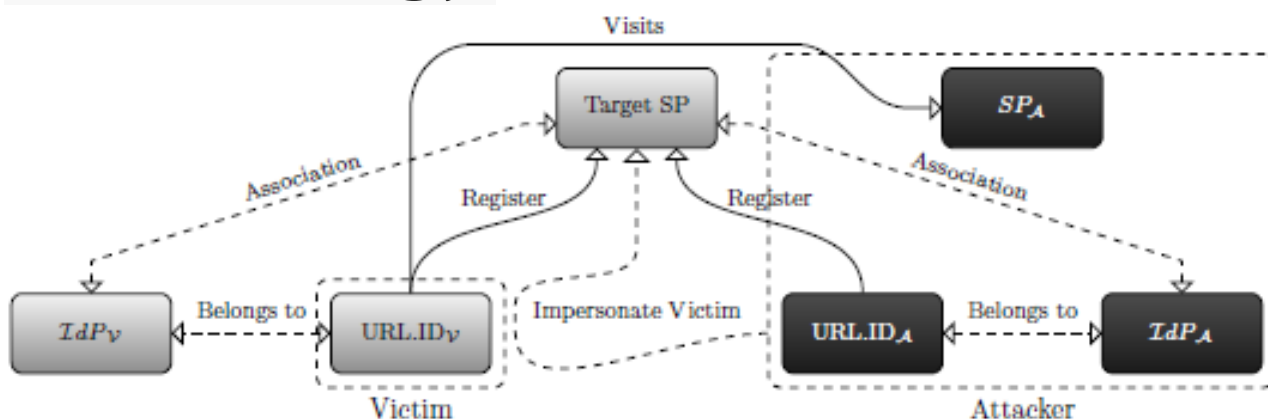


Fig. 6: Evaluation setup and goal.

OpenID Attacker

File

Server Config HTML Discovery XRDS Discovery Valid User Attack Data Attack Overview Profiles Log Viewer

Log Viewer

Select a log entry

Type	Date	Text
Token Attack	14.05.14 10:24:11	Token generated
Association	14.05.14 10:24:10	Association established: myAssocHandle
Token Valid	14.05.14 10:22:56	Token generated
Association	14.05.14 10:22:55	Association established: myAssocHandle
HTML	14.05.14 10:22:55	Requested HTML Document for 'http://[redacted]/simpleid/www/index.php...

Request:

```

GET
/simpleid/www/?openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F1.1
HTTP/1.1

openid.sreg.policy_url:http://p.sf.net/sourceforge/privacy
openid.sreg.optional:nickname,email,fullname,country,language
openid.ns:http://specs.openid.net/auth/2.0
openid.identity:http://[redacted]/simpleid/www/index.php
openid.claimed_id:http://[redacted]/simpleid/www/
openid.ns.sreg:http://openid.net/extensions/sreg/1.1
openid.mode:checkid_setup
openid.realm:https://sourceforge.net
openid.assoc_handle:myAssocHandle
openid.return_to:https://sourceforge.net/account/openid_verify

```

Response

```

GET:
openid.ns:http://specs.openid.net/auth/2.0
openid.op_endpoint:http://[redacted]/simpleid/www/
openid.claimed_id:https://me.yahoo.com/a/[redacted]
openid.response_nonce:2014-05-14T08:24:11Z0
openid.mode:id_res
openid.identity:https://me.yahoo.com/a/[redacted]
openid.return_to:https://sourceforge.net/account/openid_verify
openid.assoc_handle:myAssocHandle
openid.signed:op_endpoint,claimed_id,identity,return_to,response_nonce
openid.sig:7Afk5Zjt354fZ2bsCa7le3Bhw7Kwqq9lwnybigvDc4
openid.ns.sreg:http://openid.net/sreg/1.0

```

Clear Log

Fig. 8: IDS attack on Sourceforge. The OpenID Attacker *log viewer* window lists all exchanged OpenID messages. The Screenshot shows that the SP requests a token for URL.ID_A, but the tools ignores the wish and responds with a token for URL.ID_V.