# Security Response in the Age of Mass Customized Attacks

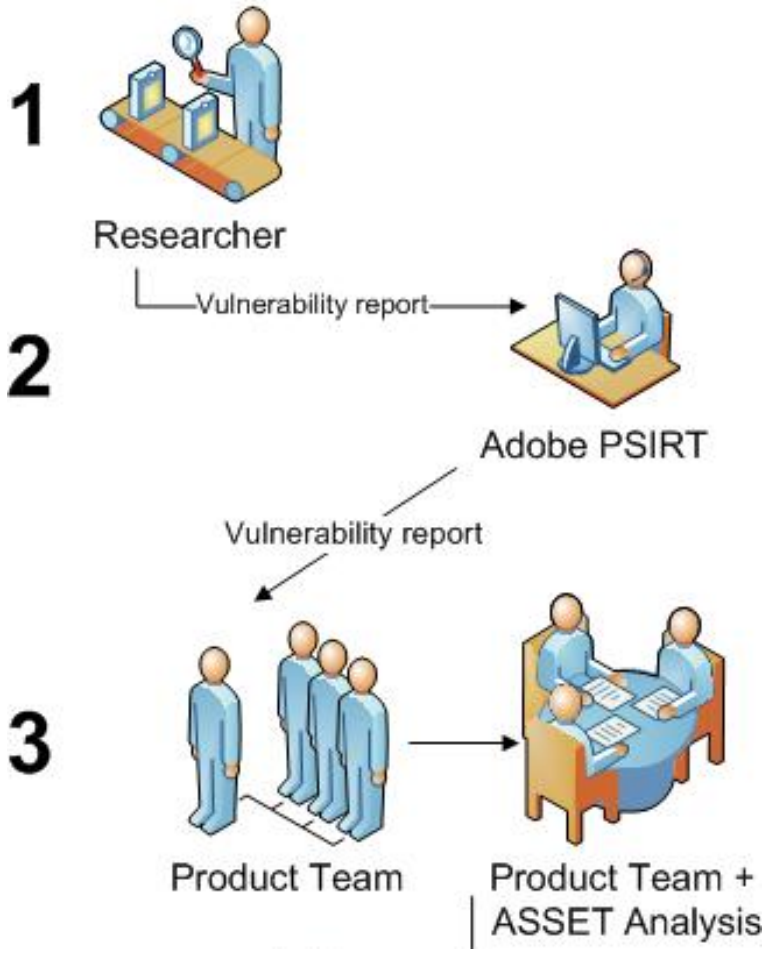
Peleus Uhley and Karthik Raman

# Overview

- Intro
- Attack Evolution
- Mass Customization
- Mass Commercialization
- Looking to the Future
- Conclusion

# About PSIRT

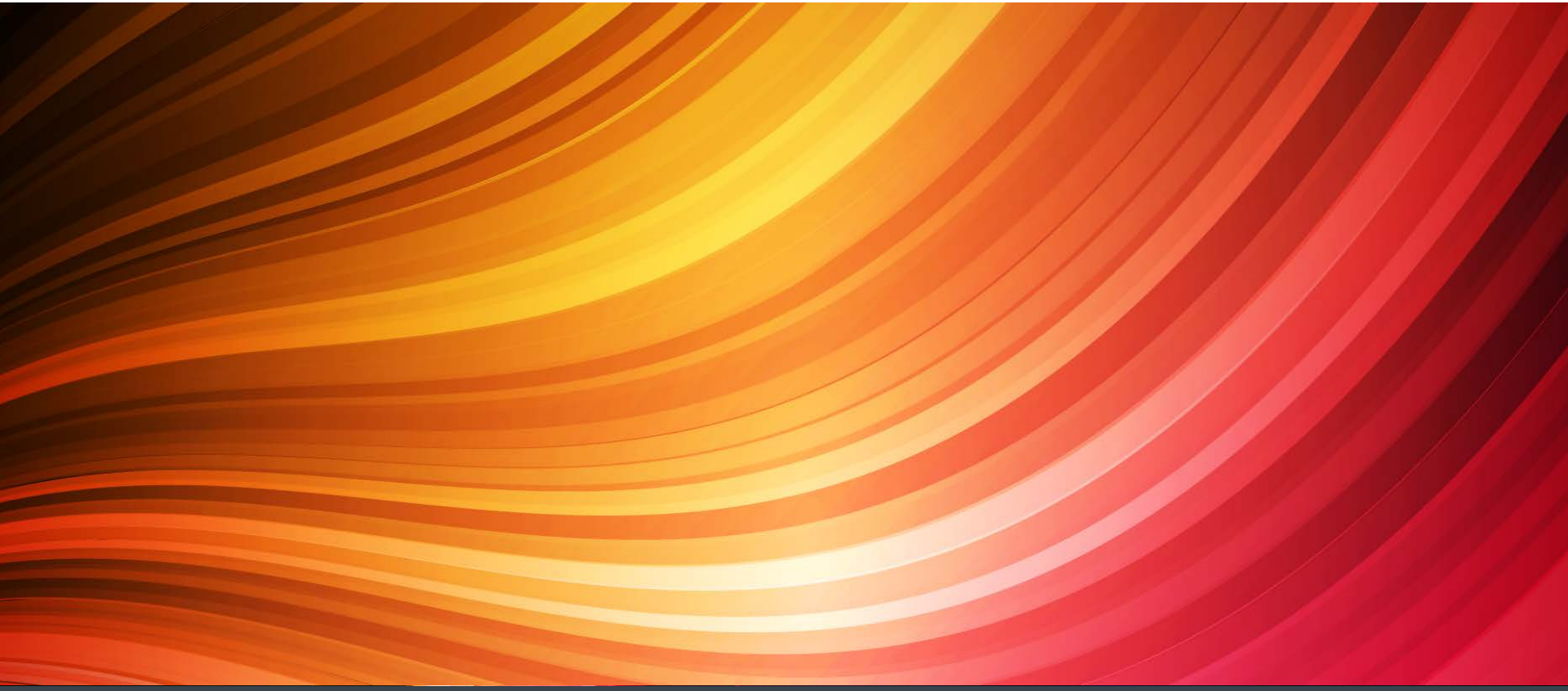
- Adobe PSIRT = Adobe Product Security Incident Response Team
- PSIRT is part of ASSET, the Adobe Secure Software Engineering Team



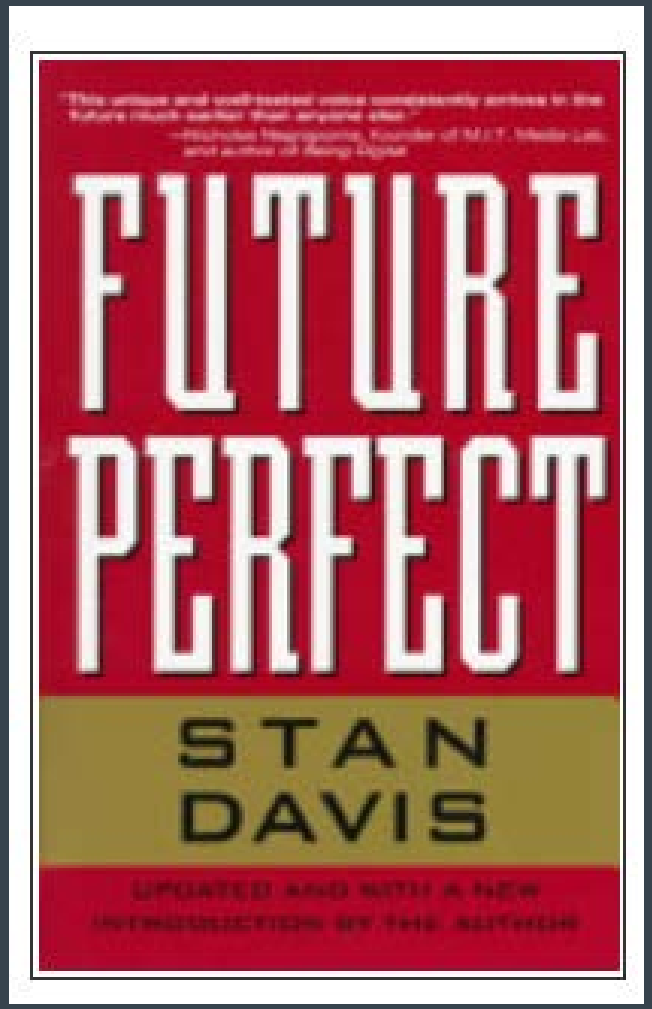
# Adobe PSIRT's Role

- Work with product teams to create fixes
- Work with researchers to verify fixes
- Publish bulletins
- Drive Adobe's involvement in MAPP

# Attack Evolution



# Mass Customization



# Mass Customization

*Customization and personalization of products and services for individual customers at a mass production price.*

*- Stan Davis*

# Mass Production

# Customization

# Mass Malware = Mass Produced

- Exploit kits
  - BlackHole, Phoenix, Mpack, Crimepack, Eleonore
  - Multiple browser and vulnerable plugin versions supported
- New modules can be added
  - 0-day exploits repurposed and added periodically
  - Payloads are also modules
- Components can be hosted anywhere
  - Serve exploits from anywhere
  - Serve malware from anywhere
- Low cost, ~$1000
- Organized groups or gangs

## Suspected hackers behind Carberp botnet, Eurograbber arrested

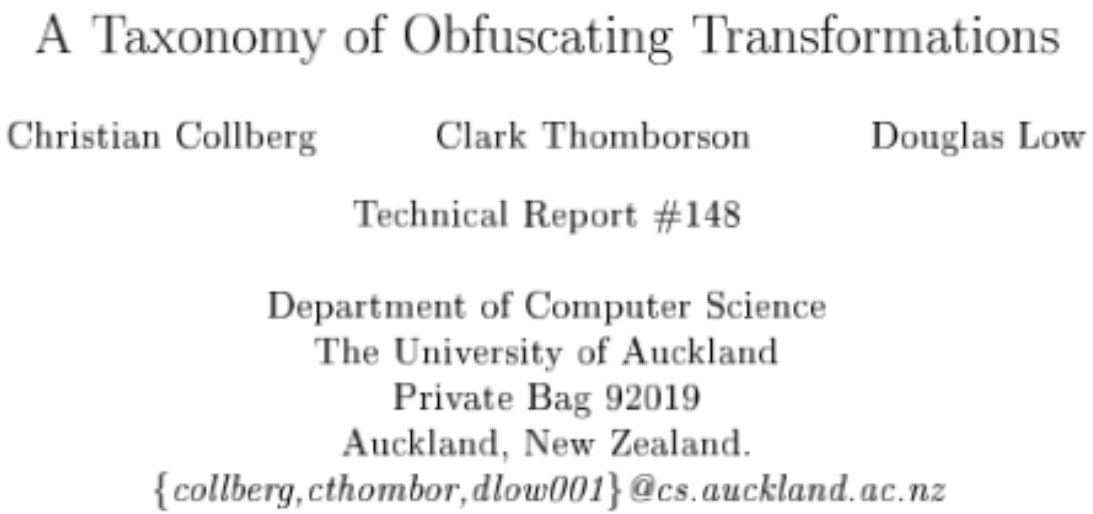
***Summary:*** *The masterminds allegedly behind a cybercrime ring which stole millions of dollars from the financial industry and consumers have been arrested.*

By Charlie Osborne for Zero Day | April 5, 2013 -- 08:35 GMT (01:35 PDT)

Follow @ZDNetCharlie

# Mass Malware Characteristics

- Potency
- Resilience
- Cost (proportional to versions supported)



A Taxonomy of Obfuscating Transformations

Christian Collberg Clark Thomborson Douglas Low

Technical Report #148

Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland, New Zealand.
{collberg,cthombor,dlow001}@cs.auckland.ac.nz

# Mass Malware Technical Characteristics

- Support multiple OS platforms
- Support multiple payloads
- Support multiple deployment scenarios
- Complex obfuscation
- CVE-2010-0188
  - Redkit, Cool, Blackhole, Nuclear, Grandsoft, Sweet Orange, NucSoft, Hierarchy, Techno Xpack, Phoenix

# Case Study in Mass Malware: CVE-2010-0188

```
KugogkdoNiqew = new Date(2011,11,4,2);
var ZewuVzozy='';
var HesexiruQaw = function(){return {e:eval}}().e;
function TuebupYtuwada(){
    var LoruhPupajuzyf='',FpynineGokedyru=[];
    var ZewuVzozy='';
    var HyqypafajIpuruk = function(){return {e:'split'}}().e;;
    var TuhLviw='2011';
    TuhLviw = '';
    var QRoser = NAz.rawValue[HyqypafajIpuruk](',');
    KycIlynovej='le'+ZewuVzozy+'ng'+ZewuVzozy+'th';
    var RyjilnaxYbanusaso = QRoser[KycIlynovej] / 2;
    var FsupoloxahekUgtaq = 'fro'+KugogkdoNiqew.getHours()+'arCode';
FsupoloxahekUgtaq=FsupoloxahekUgtaq.replace(2,'mCh');
CodadEsodisacirova=String[FsupoloxahekUgtaq];
    for (var KugaxatRocy = 0; KugaxatRocy &lt; RyjilnaxYbanusaso; KugaxatRocy++) {
        TaniruVistimolkypov=QRoser[KugaxatRocy+RyjilnaxYbanusaso] - QRoser[KugaxatRocy];
        LoruhPupajuzyf += CodadEsodisacirova(TaniruVistimolkypov);
    }
    return LoruhPupajuzyf;
}
var TaniruVistimolkypov=TuebupYtuwada();
HesexiruQaw(TaniruVistimolkypov);
```

# CVE-2010-0188

```
var funcEval = function() {
    return {
        e: eval
    }
}().e;

function preProcess() {
    var funcSplit = function() {
        return {
            e: 'split'
        }
    }().e;;
    var intArray = subFormFieldName.rawValue[funcSplit](',');
    var halfLengthOfIntArray = intArray['length'] / 2;
    funcFromCharCode = String['fromCharCode'];
    for (var i = 0; i < halfLengthOfIntArray; i++) {
        tmpIntegerValue = intArray[i + halfLengthOfIntArray] - intArray[i];
        intAccumulator += funcFromCharCode(tmpIntegerValue);
    }
    return intAccumulator;
}
var ProcessExploit = preProcess();
funcEval(ProcessExploit);
```

# CVE-2010-0188

```
function preProcess() {
    var funcSplit = function() {
        return {
            e: 'split'
        }
    }().e;;
    var subFormFieldName = '93,136,111,218,224,106,120,284,127,147,247,273,83,1
    var intArray = subFormFieldName.split(',');
    var halfLengthOfIntArray = intArray['length'] / 2;
    funcFromCharCode = String['fromCharCode'];
    var intAccumulator;
    for (var i = 0; i < halfLengthOfIntArray; i++) {
        tmpIntegerValue = intArray[i + halfLengthOfIntArray] - intArray[i];
        intAccumulator += funcFromCharCode(tmpIntegerValue);
    }
    return intAccumulator;
}
var RunExploit = preProcess();
eval(RunExploit);
```

# CVE-2010-0188

```
var _GD = "7414543e6471e52c5e1356366b50e27390deb27d04l6e62e5fl6b77971777970le34343161
var _ZZ = "7414543ec405e52c5e135636f212e273a32ab27d04a6e02e4c47b77971777970l234343161
var _IB = "8441afefb369d3b9352746dd19730681";
_II = app;
_R = new Array();

function getVersion() {

    var _H = app.viewerVersion.toString();
    _H = _H.replace('.', '');
    while (_H.length < 4) {
        _H += '0';
    }
    var ret =  parseInt(_H, 10);
    return ret;
}
function paddToHalfSecondParamLen(_I, _M) {
    while (_I.length * 2 < _M) {
        _I += _I;
    }
    return _I.substring(0, _M / 2);
}
function _FA(_MM) {
```

# CVE-2010-0188

```
    }
    _A = 'SUkqADggAABB';
    _MF = _SR('QUFB', 10984);
    _J = 'QQcAAAEDAAEAAAAwIAAAAQEDAAEAAAABAAAAAwEDAAEAAAABAAAABgEDAAEAAAAB
    _L = _A + _MF + _J + _RS;
    _D = decode(_CP, _IB);
    if (_D.length % 2) {
        _D += unescape('%00');
    }
    _X = _LX(_D);
  execShellCode = primeShellcode(_X); // <== Spray heap with exploit code
    var NAz = _L; // <== Tiff image
}
_YS();
```

# The System of Mass Production

## Table 2-3 Principles of Mass Production

From the American System

- Interchangeable parts
- Specialized machines
- Focus on the process of production
- Division of labor

Additional Principles

- Flow
- Focus on low costs and low prices
- Economies of scale
- Product standardization
- Degree of specialization
- Focus on operational efficiency
- Hierarchical organization with professional managers
- Vertical integration

# CVE-2011-2462

```
function urpl(k,sc){
var c = "\x75";
var kc=k+c;
var re = /MM/g;
sc = sc.replace(re,kc);
return sc;
//}//.\#
}
```

```
s/urpl\\\((pattern1),(pattern2)\\\)/deobfuscate($1,$2)/eg;
```

```
function spray_heap946(ppp)
{
xyz(ppp);zy946();xxyyzz();xxxyyyzzz();xxxxyyyyzzzz();xxxxxyyyyyzzzzz();xxxxxxyyyyyyzzzzzz();
if(ver>9.0)
    xxxxxxxyyyyyyyzzzzzzz();
xxxxxxxxyyyyyyyyzzzzzzzz();xxxxxxxxxyyyyyyyyyzzzzzzzzz();zzzzzzzzzzz();zzzzzzzzzzzz();
}
```

# CVE-2011-2462

```
if (ver>20)
{
datagood(9,8);
while(1);
}
else
{    if(ver>10.7)
        {
            databad(7,9);
            while(1);
        }

    else
        {
            if(ver>10.0)
                {
                    while(1);
                }
            else
                 {
```

# CVE-2011-2462

*Like most JavaScript observed in other malicious files, checks are done for the proper version number before the main routines are executed. What is interesting about this document is that it* ***checks for versions that do not exist and makes a point to redirect the user to an infinite loop assuming they are running a version greater than 10.***

*-Brandon Dixon*

# CVE-2011-2110

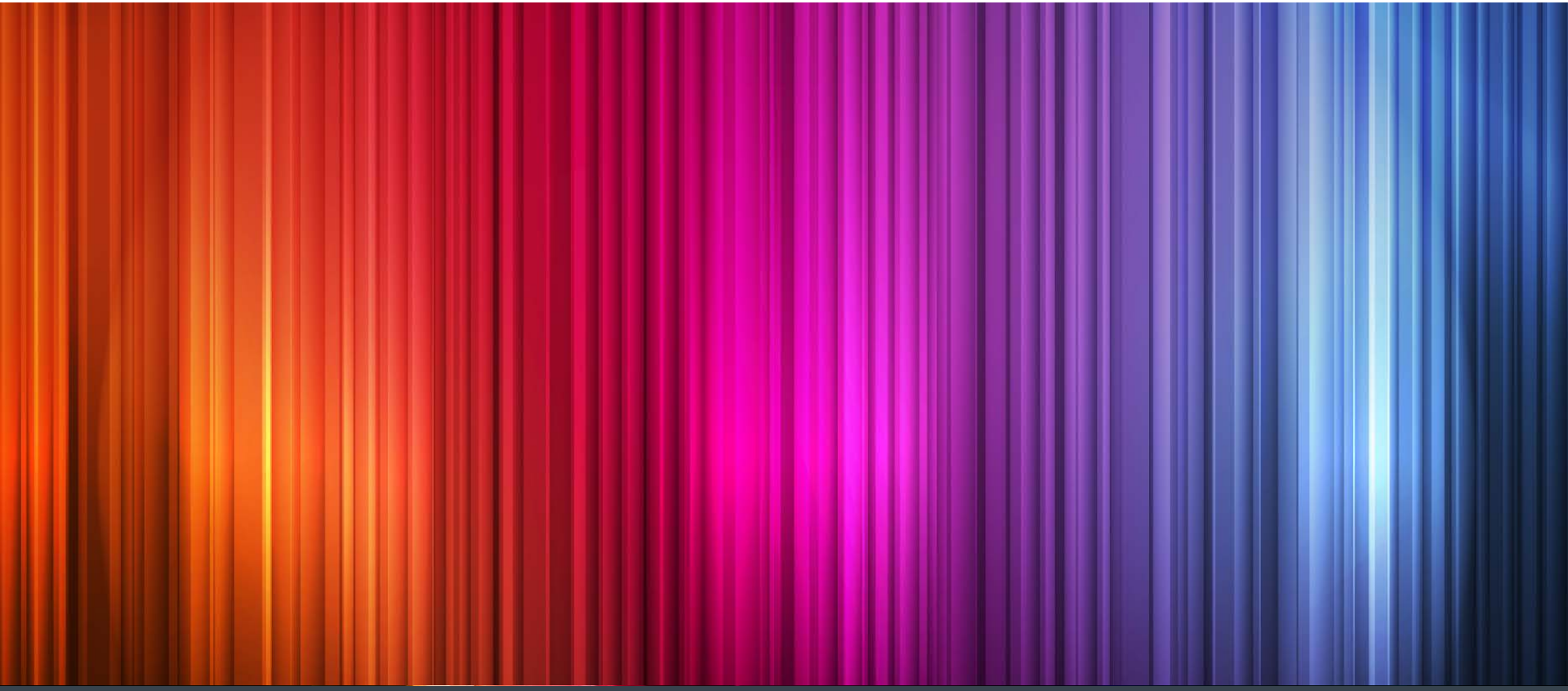
- Dynamically passing obfuscated data

  main.swf?info=02E6B1525353CAA8AD555555AD31B3D73034B657AA31B4B5AFB5B2B537AF55543549AEB550AC55303736B337AF51D3527B7AF4C66B7E

- Targeting specific versions

```
if ((((((Capabilities.version.toLowerCase() == "win 10,3,181,14")) ||
((Capabilities.version.toLowerCase() == "win 10,3,181,22")))) ||
((Capabilities.version.toLowerCase() == "win 10,3,181,23")))){
```

# Mass Customization

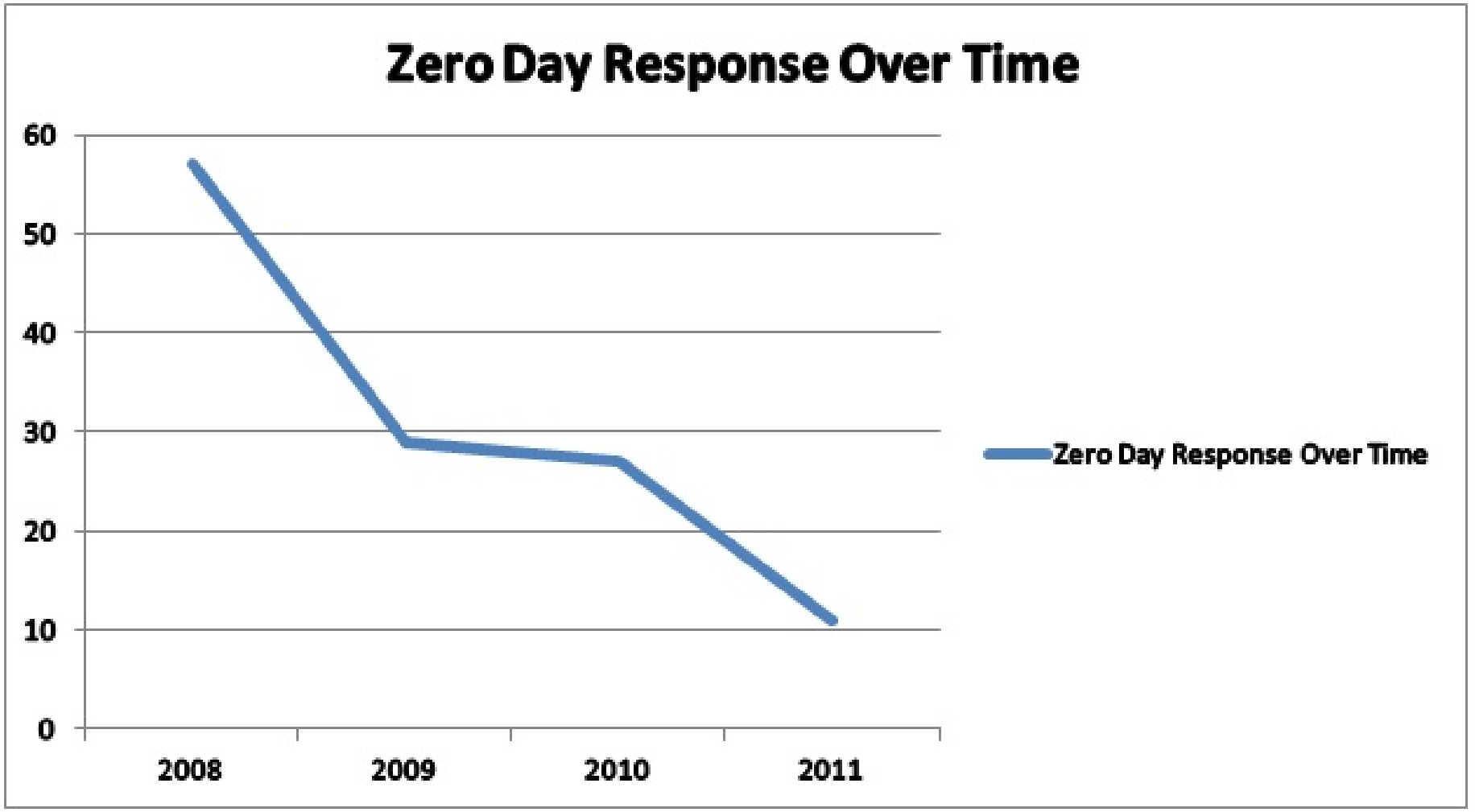


# What Drives Mass Customization

**Table 3-1 Features of the Competitive Landscape of the 1990s**

- Time-based competition
- Proliferating variety
- Just-in-time production
- Regional marketing
- Continual improvement
- Shortening product life cycles
- Market-driven quality
- Globalization
- Networked organizations
- Micromarketing
- Increased customization
- Lean production
- Cycle time reduction
- Total quality management
- Flattened hierarchies
- Computer-integrated manufacturing
- Process re-engineering
- Heightened importance of services
- Fragmented markets
- Quick response
- Flexible manufacturing systems
- Database marketing

# Response and Feedback



# Mass Customized Attacks

# Case Study 1: CVE-2013-0633

- Mass malware characteristic:  Version checking

Diavlo

```
if (Capabilities.version.indexOf("WIN 11") < 0){
    throw (new Error("unsupported"));
};
```

# Modular Design - CVE-2013-0633

```
1   package {
2       import flash.text.engine.*;
3       import __AS3__.vec.*;
4       import flash.display.*;
5       import flash.net.*;
6       import flash.utils.*;
7
        public class FontTest extends Sprite {
9
10          static var counter:uint = 0;
11
12          private var Diavlo:Class;
13
            public function FontTest(){ ... }
            function loadFont(_arg1:String):TextLine{ ... }
            function buildShellcode(_arg1:Vector.<int>, _arg2:uint, _arg3:uint, _arg4:uint){ ...
            function buildROP(_arg1:Vector.<int>, _arg2:uint, _arg3:uint):int{ ... }
            function getMemoryAt(_arg1:Vector.<int>, _arg2:uint, _arg3:uint):uint{ ... }
            function writeMemoryAt(_arg1:Vector.<int>, _arg2:uint, _arg3:uint, _arg4:uint){ ... }
418
419     }
420 }//package
```

# Case Study 2: CVE-2013-0634 (aka Lady Boyle)

- Attack used complex memory layout to achieve information leak
- Tied back to "july.swf" (CVE-2012-5054)
- First exploit to target Safari and Firefox on a ***Mac***
- Windows version delivered via Office documents
- Windows version had payloads for 32-bit and 64-bit version
- The malicious 32-bit payload was digitally signed

# Multiple Win Platforms - CVE-2013-0634

```
switch (_local19) {
    case "windows 7":
        break;
    case "windows server 2008
r2":
        break;
    case "windows server
2008":
        break;
    case "windows server 2003
r2":
        break;
    case "windows server
2003":
        break;
    case "windows xp":
        break;
    case "windows vista":
        break;
    default:
        return (this.empty());
};
```

```
switch (_local27)  {
    case "win 11,5,502,146":
       …
    case "win 11,5,502,135":
       …
    case "win 11,5,502,110":
       …
    case "win 11,4,402,287":
       …
    case "win 11,4,402,278":
       …
    case "win 11,4,402,265":
       …
```

# Case Study 3: CVE-2013-0640

- Reader 0-day is not one bug!
- One buffer overflow
- One information leak
- One sandbox escape
- JavaScript was heavily obfuscated

```
sHOGG\('014.031.4.',3571,9173\)
s/sHOGG\\\(('.+?'),(\d+?),(\d+?)\\\)/deobfuscate($1,$2,$3)/eg
+5+2*2*13\);r+=ue\(t+19*3\);r+=ue\(t+3*19\);r+=ue\(t+19*3\);r+=ue\(t
+19*3\);r+=ue\  if \(vOLENCI["dIAVOLO"] == "XXX"\)  2*2*2*2*2+5*5\)
ue\(t+19*3\);r+=ue\(t+19*3\);r+=ue\(t+19*3\);r+=ue\(
```

- Used Reader extended features

# Supported Versions - CVE-2013-0640

- 11.0.1.36
- 11.0.0.379
- 10.1.5.33
- 10.1.4.38 (plus sub-version for three languages)
- 10.1.3.23
- 10.1.2.45
- 10.1.0.534
- 9.5.3.305
- 9.5.2.0 (plus sub-version for three languages)
- 9.5.0.270

# Case Study 4: CVE-2013-0648

- Flash sandbox escape
- Appeared to still be under development:
  - Targeted a single older version of Flash Player
  - Only targeted Firefox
- ActionScript had shell code for MSIE, Firefox, Opera and Chrome
- Required two SWFs and a web page

# Digital Evidence of Commercialization

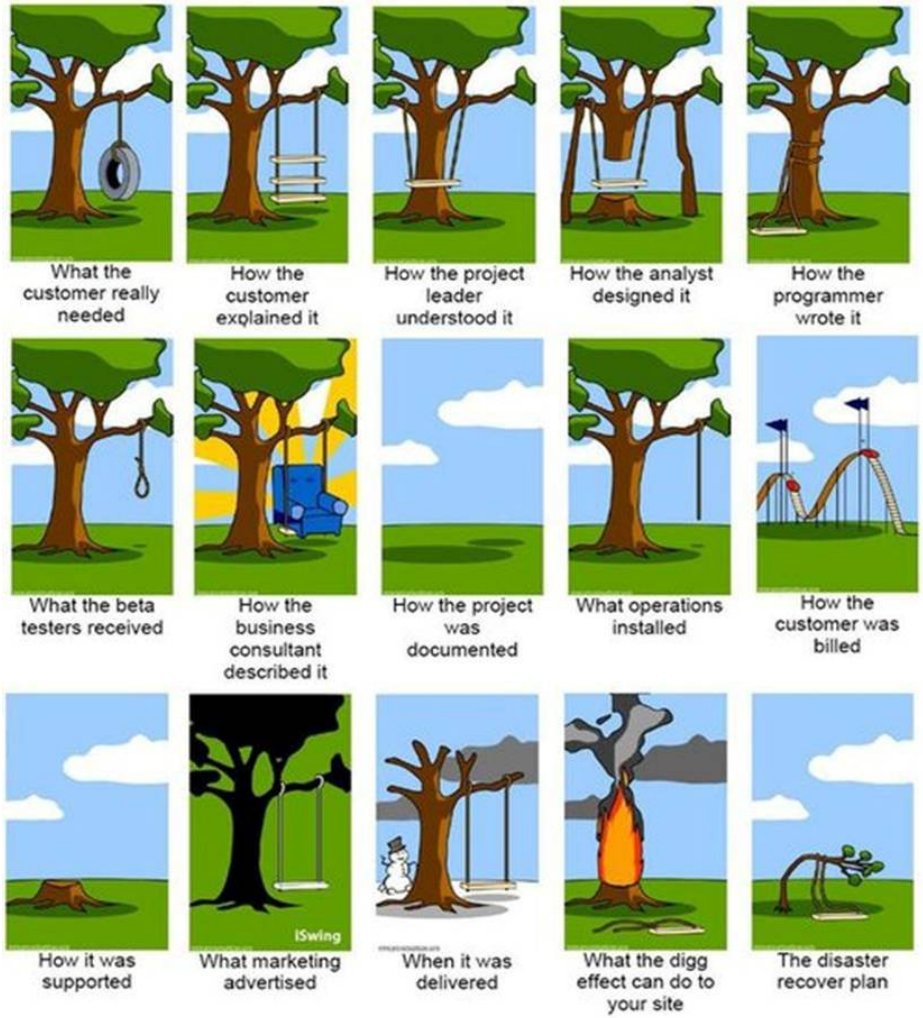
# Definition of APT

- There are multiple definitions of APT:
  - The group behind my embarrassing XSS attacks
  - The reason you should buy my new magical security widget
  - Groups that conduct international cyber attacks for economic or military gain

- One government TLA representative described APT as, "Any attack that involves a project manager."

# Evidence of Project Managers?

- Reversing the spec from the code leads to the following assumptions:
  - "Support all versions of Reader"
  - "Support all versions of Windows"
  - "Support all current versions of Flash"
  - "Support all browsers"
- Reader 0-day was approx. 8,750 SLOC of JavaScript alone
- As complexity increases, will this cartoon soon apply to exploits?



# Looking Towards the Future

# More Focused Attacks

- Increased attack resources and automation will lead to a lower cost of entry and more focused attacks
- Ability to target platforms with smaller distribution numbers such as Macs
- Loose coupling means faster turnaround for copy cat attacks (MiniDuke, itaDuke, etc.)

# Multi-Vendor/Multi-Product Customizations

- Pwn2Own 2013
  - Chrome + Windows kernel exploit
  - Flash Player + IE10 exploit

- CVE-2013-0648 involved two Adobe products

- Peter Vreugdenhil discussed a multi-vendor PDF exploit at CanSecWest 2013

# Response Must Grow

- As attacks become more complex, response becomes more complex
  - Flash sandbox escape -2 bugs
  - Reader sandbox escape - 3 bugs
  - PinkiePie's Pwnium attack - 6 bugs
  - Sergey Glazunov's Pwnium attack - ***14 bugs***
- Analysis must include shell code payloads
- Defense in depth approach
- Multi-vendor/multi-product coordination

# Possible Benefits for Defenders

- A quick patch on to a critical link in the chained exploit may buy time to address the other bugs

- For AV/IDS/IPS vendors:
  - Disadvantage: Need signatures for each component
  - Advantage: Possibly target the frameworks which have less flexibility in changing

- Greater exploit complexity means that attackers have a greater chance for bugs in their own code (we hope ☺ )

# Future Defensive Research: Normalization

**Software Transformations to Improve Malware Detection**

Mihai Christodorescu[1], Somesh Jha[1], Johannes Kinder[2], Stefan Katzenbeisser[2], and Helmut Veith[2]

[1] University of Wisconsin, Madison, {mihai, jha}@cs.wisc.edu
[2] Technische Universität München, {kinder, katzenbe, veith}@in.tum.de

Malware Normalization

Mihai Christodorescu* Johannes Kinder† Somesh Jha* Stefan Katzenbeisser† Helmut Veith†

*University of Wisconsin, Madison
{mihai,jha}@cs.wisc.edu

†Technische Universität München
{kinder,katzenbe,veith}@in.tum.de

**Using Code Normalization for Fighting Self-Mutating Malware**

Danilo Bruschi, Lorenzo Martignoni, Mattia Monga
Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39, 20135 Milano
{bruschi,martign,monga}@dico.unimi.it

**Imposing Order on Program Statements to Assist Anti-Virus Scanners**

Arun Lakhotia and Moinuddin Mohammed
*University of Louisiana at Lafayette*
*arun@louisiana.edu*

# Conclusion

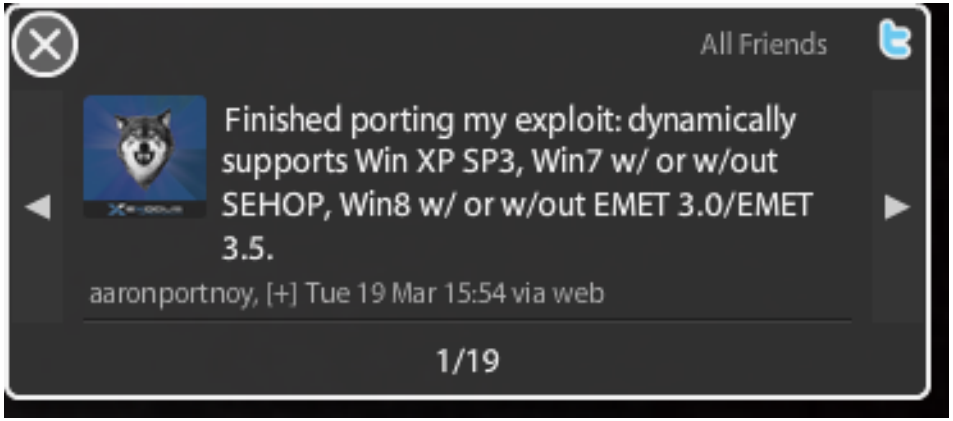
# Conclusion


**Table 3-1 Features of the Competitive Landscape of the 1990s**

- Time-based competition
- Proliferating variety
- Just-in-time production
- Regional marketing
- Continual improvement
- Shortening product life cycles
- Market-driven quality
- Globalization
- Networked organizations
- Micromarketing
- Increased customization
- Lean production
- Cycle time reduction
- Total quality management
- Flattened hierarchies
- Computer-integrated manufacturing
- Process re-engineering
- Heightened importance of services
- Fragmented markets
- Quick response
- Flexible manufacturing systems
- Database marketing

# Response to Mass Customization

- Flexible manufacturing systems/product modularization
  - Response becomes proportional to exploit complexity
  - Shell code payloads must be analyzed for additional bugs
  - Must be prepared for multi-product/multi-vendor situations
- Cycle time reduction, shortening product lifecycle
  - Quick response & distribution due to ease of incorporating into mass malware
  - Increased rate of updates
- Increased customization
  - Defense in depth for meta bugs
  - Robust patch testing



# Questions?

- ASSET Blog
  - https://blogs.adobe.com/asset/
- Adobe PSIRT
  - psirt@adobe.com
- Email
  - puhley@adobe.com
  - kraman@adobe.com
- Twitter
  - https://twitter.com/PeleusUhley

# Credits

- Ford Model T: http://commons.wikimedia.org/wiki/File:Restored_Ford_Model_T.jpg
- Principles of Mass Production, Competitive Landscape in 1990s, Comparisons: http://books.google.com/books?isbn=0875849466
- Mass Customization: The New Frontier in Business Competition: http://books.google.com/books?id=2_3PMy4LQHkC
- Bugatti Replica: http://commons.wikimedia.org/wiki/File:Long_Beach_custom_car_show_1991_-_Flickr_-_exfordy_%285%29.jpg
- ZDNet Zero Day blog: http://www.zdnet.com/suspected-hackers-behind-carberp-botnet-eurograbber-arrested-7000013580/
- Lady Boyle: Image: http://www.behindthevoiceactors.com/_img/chars/char_82998.jpg
- Development process: http://wisevishvesh.files.wordpress.com/2010/10/sdlc.jpg?w=869
- Chaouki Bekrar: https://twitter.com/thezdi/status/310181147445444609/photo/1


Adobe