

# Defending Computer Networks

## *Lecture 3: More On Vulnerabilities*

Stuart Staniford

Adjunct Professor of Computer Science


# Logistics

- First Homework Today
  - Find on class website
  - Due next Wednesday 9/11 5pm in 317 lab
    - Try not to all come at last minute
- Office hour change
  - Permanent:
    - Tues is 1:30-2:45pm (not 3pm)
  - Temporary
    - Weds 9/11 will be 1:30-5pm for HW1 grading
      - Could be TA in part, if dept can find one by then
    - Tues office hours will be in lab next week also

# More Logistics

CS 5434 - Defending Com x

www.cs.cornell.edu/courses/cs5434/2013fa/readings.html



Cornell University  
Department of Computer Science

**CS 5434 - Defending Computer Networks - Fall 2013**

Summary   Lectures   **Readings**   Homework

#	Author	Title/Link
1	Aleph1	<a href="#"><i>Smashing the Stack for Fun and Profit</i></a>
2	Matt Conover	<a href="#"><i>w00w00 on heap overflows</i></a>
3	Scut	<a href="#"><i>Exploiting Format String Vulnerabilities</i></a>
4	Blexim	<a href="#"><i>Basic Integer Overflows</i></a>
5	Mitre	<a href="#"><i>Common Weakness Enumeration &gt;/i&gt;</i></a>
6	Steve Christey	<a href="#"><i>Unforgivable Vulnerabilities</i></a>
7	Christey et al	<a href="#"><i>Structured CWE Descriptions</i></a>
8	Cowan et al	<a href="#"><i>StackGuard: Automatic Adaptive Detection and Prevention of</i></a>
9	Shacham et al	<a href="#"><i>On the Effectiveness of Address-Space Randomization</i></a>

# Main Goals for Today

- Understand format string vulnerabilities
- Introduction to CVE/CWE - enumerating and classifying vulnerabilities
- The economic/social big picture: why is the Internet riddled with vulnerabilities?
- Understand stack canary defenses against buffer overflows

# Interesting News This Time



## Analysis: Syria, aided by Iran, could strike back at U.S. in cyberspace

### US likely to wage cyber attacks against Syria

By Brendan Sasso - 09/04/13 05:45 AM ET

Tweet 36 Like 8 Send +1 2

COMMENTS  
EMAIL  
PRINT  
SHARE

Recommend 250 people recommend this.

The United States is likely to make cyber attacks part of any military action against Syria, experts say.

"I think that's a certainty," said Jim Lewis, a senior fellow with the Center for Strategic and International Studies and the director of the Technology and Public Policy Program.



By Joseph Menn  
SAN FRANCISCO | Thu Aug 29, 2013 7:07am EDT

(Reuters) - If the United States attacks Syria, it will be the first time it strikes a country that is capable of waging retaliatory cyberspace attacks on American targets.

The risk is heightened by Syria's alliance with Iran, which has built up its cyber capability in the past three years, and already gives the country technical and other support. If Iran stood with Syria in any fray with the United States that would significantly increase the cyber threat, security experts said.

Tweet 94  
Share 19  
Share this  
+1 5  
Email  
Print

Factbox  
Factbox: If U.S. strikes Syria, destroyers likely to deliver the blow  
Wed, Aug 28 2013

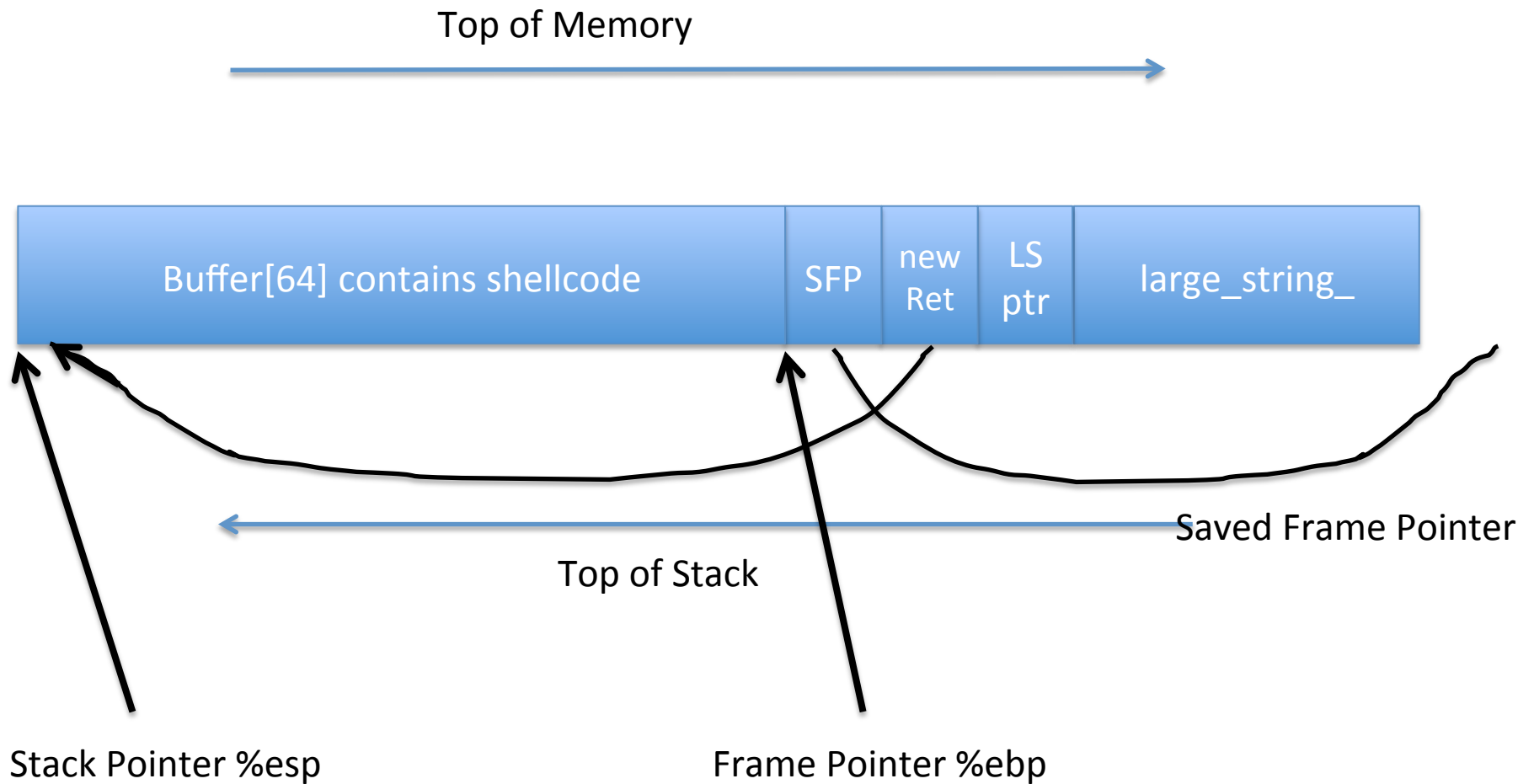
- Related News
- Obama makes case for Syria strike, British house votes no  
Thu, Aug 29 2013
  - Obama makes case for punishing Syria over gas attack  
Wed, Aug 28 2013
  - New York Times, Twitter hacked by

# Refresh: Example 2

```
void myFunc(char *str)
{
    char buffer[64];
    strcpy(buffer, str);
}

int main(int argc, char* argv[])
{
    char large_string[256];
    int i;
    for( i = 0; i < 255; i++)
        large_string[i] = 'A';
    myFunc (large_string);
}
```

# Refresh: Stack Before Strcpy



# Format String Vulnerabilities

- Class of vulnerabilities in C printf/sprintf/etc
  - Discovered at end of 1990s
  - Almost thirty years after C invented
- Also can affect syslog(), warn(), err()
- Core issue is when the format string is (partially) user supplied, not static
  - printf(userData,...) is bad
  - printf(“%s”, userData,...) doesn't have the issue
- Note that '...' arguments are on the stack
- And format string controls powerful functionality to do stuff with the printf arguments (ie the stack)
- Bypass any compile time checks when user supplied



# printfVulnerability.c

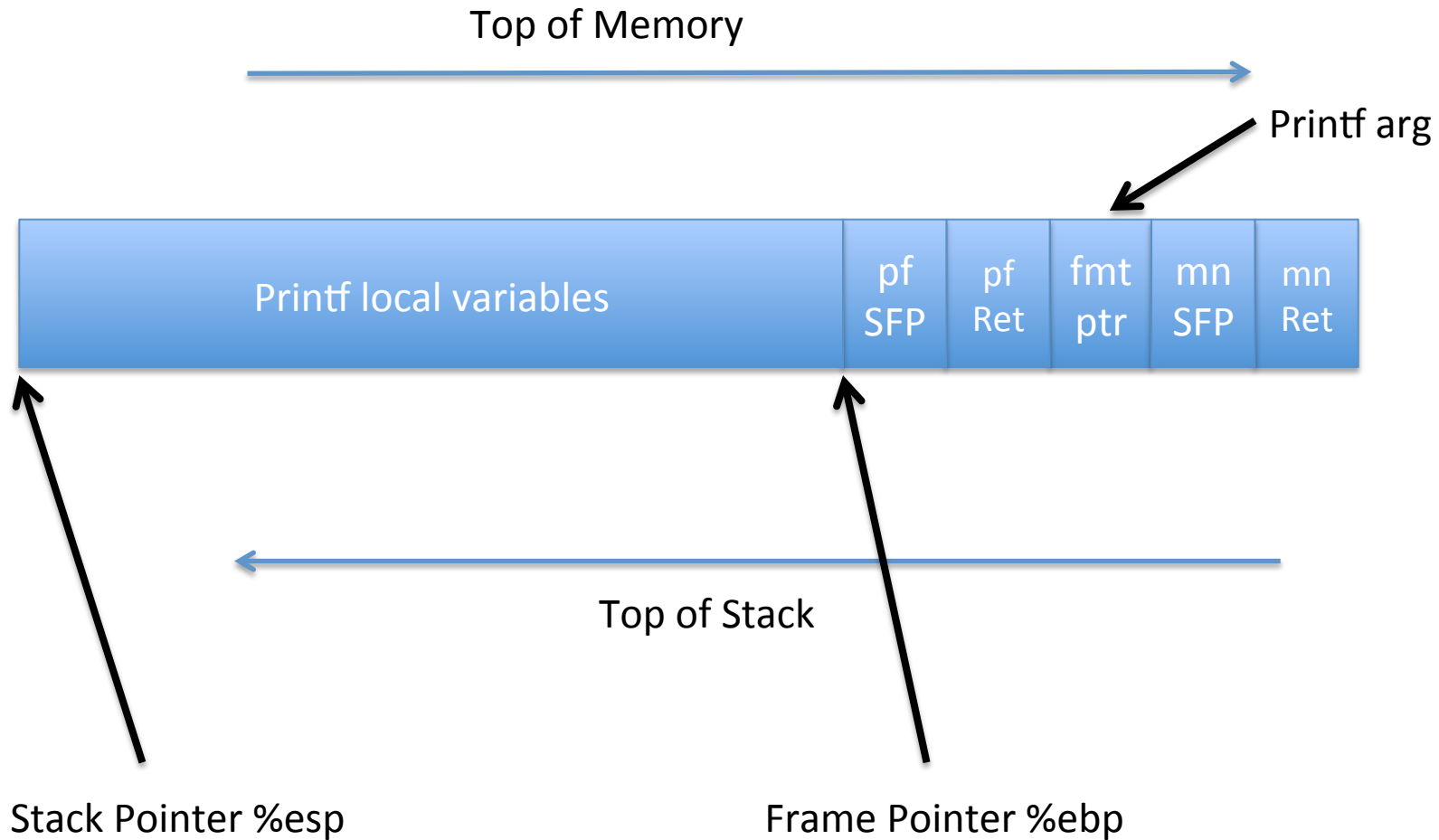
```
#include <stdio.h>
```

```
int main(int argc, char* argv[])  
{  
    printf(argv[1]);  
    return 0;  
}
```

## So let's try...

- `./printfVulnerability foo`
- `./printfVulnerability "foo \n"`
- `./printfVulnerability "%08x.%08x.%08x.%08x "`

# Stack in printf Vulnerability



Still in the no-defense, old-style, slightly fictionalized view of the world

# The Really Big Problem

The ‘%n’ format directive. From ‘man 3 printf’:

```
n      The number of characters written so far is stored into the integer indicated by the int * (or variant) pointer argument. No argument is converted.
```

This allows us to write on the stack at a location we can control! by doing `./printfVulnerability "%08x.%08x... %n"` we can move around the stack position where the %n will write to.

By doing `"aaaaa...%08x.%08x... %n"` we can control what number is written into that stack address.

By doing this repeatedly with small width fields, we can write one byte at a time, and overwrite an address (eg a saved IP)

# Common Vulnerabilities and Exposures List (CVE)

- Initiative by Mitre Corp to create a common dictionary of known vulnerabilities
  - US govt funded
- Now an industry standard
- Guided by an editorial board from across industry/academia/government
- Excellent Place to Start Looking for Publicly Known Vulnerabilities in something
  - <http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=excel>

# 'Excel' Vulnerabilities

## Search Results

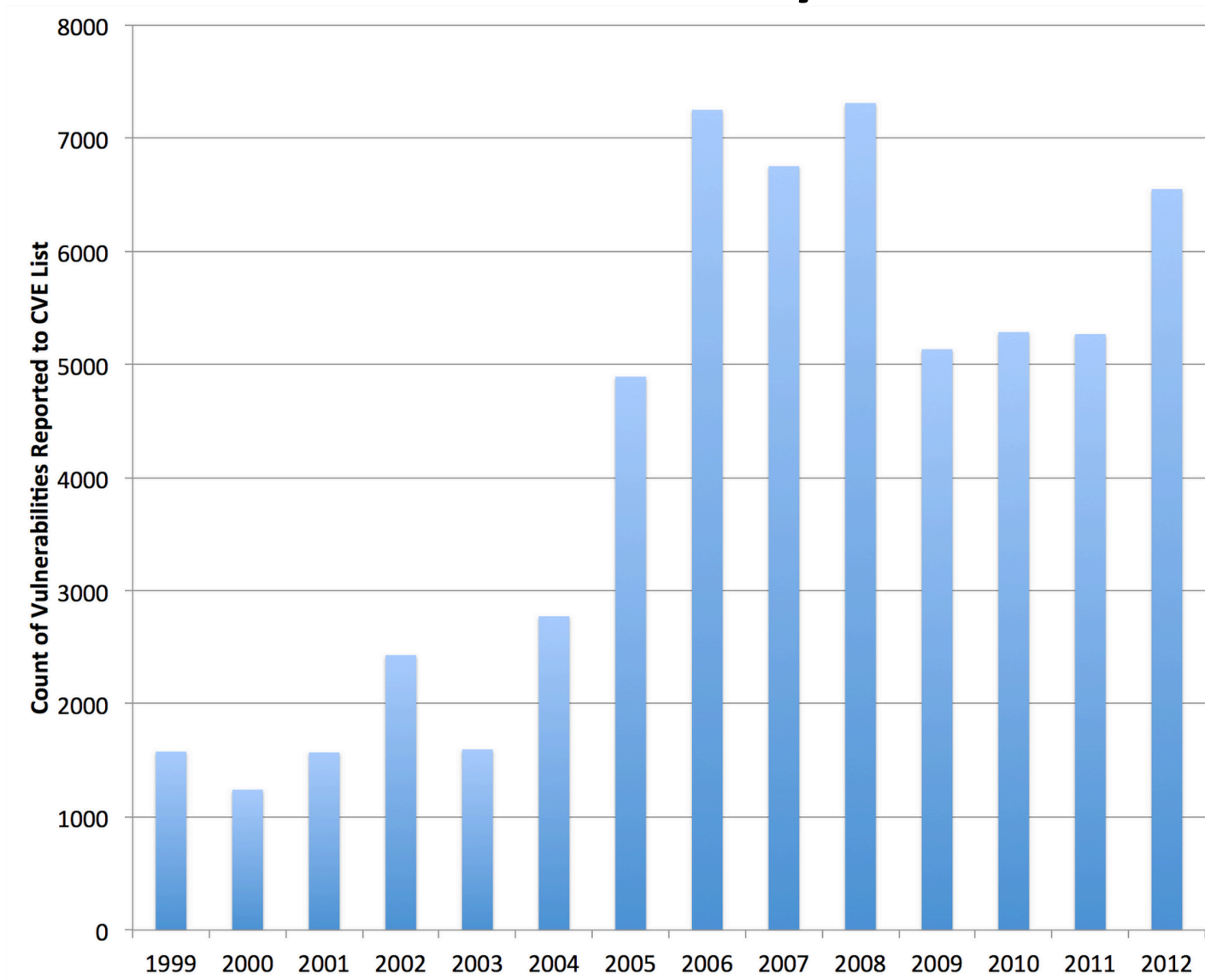
There are **201** CVE entries that match your search.

Name	Description
<a href="#">CVE-2012-5672</a>	Microsoft Excel Viewer (aka Xlview.exe) and Excel in Microsoft Office 2007 (aka Office 12) allow remote attackers to cause a denial of service (read access violation and application crash) via a crafted spreadsheet file, as demonstrated by a .xls file with battery voltage data.
<a href="#">CVE-2012-4233</a>	LibreOffice 3.5.x before 3.5.7.2 and 3.6.x before 3.6.1, and OpenOffice.org (OOo), allows remote attackers to cause a denial of service (NULL pointer dereference) via a crafted (1) odt file to vclo.dll, (2) ODG (Drawing document) file to svxcorelo.dll, (3) PolyPolygon record in a .wmf (Window Meta File) file embedded in a ppt (PowerPoint) file to tlo.dll, or (4) xls (Excel) file to scfiltlo.dll.
<a href="#">CVE-2012-2543</a>	Stack-based buffer overflow in Microsoft Excel 2007 SP2 and SP3 and 2010 SP1; Office 2011 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel Stack Overflow Vulnerability."
<a href="#">CVE-2012-1887</a>	Use-after-free vulnerability in Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 SP1, and Office 2008 and 2011 for Mac, allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel SST Invalid Length Use After Free Vulnerability."
<a href="#">CVE-2012-1886</a>	Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 SP1; Excel Viewer; and Office Compatibility Pack SP2 and SP3 allow remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted spreadsheet, aka "Excel Memory Corruption Vulnerability."
<a href="#">CVE-2012-1885</a>	Heap-based buffer overflow in Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 SP1; Office 2008 and 2011 for Mac; and Office Compatibility Pack SP2 and SP3 allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel SerAuxErrBar Heap Overflow Vulnerability."
<a href="#">CVE-2012-1847</a>	Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 Gold and SP1; Office 2008 and 2011 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 do not properly handle memory during the opening of files, which allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel Series Record Parsing Type Mismatch Could Result in Remote Code Execution Vulnerability."
<a href="#">CVE-2012-0185</a>	Heap-based buffer overflow in Microsoft Excel 2007 SP2 and SP3 and 2010 Gold and SP1, Excel Viewer, and Office Compatibility Pack SP2 and SP3 allows remote attackers to execute arbitrary code via a crafted spreadsheet that triggers incorrect handling of memory during opening, aka "Excel MergeCells Record Heap Overflow Vulnerability."
<a href="#">CVE-2012-0184</a>	Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 Gold and SP1; Office 2008 and 2011 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 do not properly handle memory during the opening of files, which allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel SXLI Record Memory Corruption Vulnerability."
<a href="#">CVE-2012-0143</a>	Microsoft Excel 2003 SP3 and Office 2008 for Mac do not properly handle memory during the opening of files, which allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel Memory Corruption Using Various Modified Bytes Vulnerability."
<a href="#">CVE-2012-0142</a>	Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 Gold and SP1; Office 2008 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 do not properly handle memory during the opening of files, which allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel File Format Memory Corruption in OBJECTLINK Record Vulnerability."
<a href="#">CVE-2012-0141</a>	Microsoft Excel 2003 SP3, 2007 SP2 and SP3, and 2010 Gold and SP1; Office 2011 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 do not properly handle memory during the opening of files, which allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel File Format Memory Corruption Vulnerability."

# Example Entry

CVE-ID	
<b>CVE-2012-2543</b>	<a href="#">Learn more at National Vulnerability Database (NVD)</a> <ul style="list-style-type: none"><li>Severity Rating</li><li>Fix Information</li><li>Vulnerable Software Versions</li><li>SCAP Mappings</li></ul>
Description	
Stack-based buffer overflow in Microsoft Excel 2007 SP2 and SP3 and 2010 SP1; Office 2011 for Mac; Excel Viewer; and Office Compatibility Pack SP2 and SP3 allows remote attackers to execute arbitrary code via a crafted spreadsheet, aka "Excel Stack Overflow Vulnerability."	
References	
<b>Note:</b> <a href="#">References</a> are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.	
<ul style="list-style-type: none"><li>MS:MS12-076</li><li><a href="http://technet.microsoft.com/security/bulletin/MS12-076">URL:http://technet.microsoft.com/security/bulletin/MS12-076</a></li><li>CERT:TA12-318A</li><li><a href="http://www.us-cert.gov/cas/techalerts/TA12-318A.html">URL:http://www.us-cert.gov/cas/techalerts/TA12-318A.html</a></li><li>BID:56431</li><li><a href="http://www.securityfocus.com/bid/56431">URL:http://www.securityfocus.com/bid/56431</a></li><li>SECTrack:1027752</li><li><a href="http://www.securitytracker.com/id?1027752">URL:http://www.securitytracker.com/id?1027752</a></li></ul>	

# CVE Counts By Year





# Is That All The Vulnerabilities?

- No!
  - Anecdote/single datapoint (8000 vs 122)
- Basically, we have no idea how many software vulnerabilities exist in total
  - Probably millions at least,
  - Probably not billions

# Common Weakness Enumeration(CWE)

- More recent effort by Mitre
  - [cwe.mitre.org](http://cwe.mitre.org)
- Idea is to classify vulnerabilities into general types
  - See the recurring patterns
  - Prioritize what's most important
  - Learn not to generate more and more of these things
- Excellent place to look for general issues when you get into a new domain

# CWE Top 25

Rank	Score	ID	Name
[1]	93.8	<a href="#">CWE-89</a>	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
[2]	83.3	<a href="#">CWE-78</a>	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
[3]	79.0	<a href="#">CWE-120</a>	Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')
[4]	77.7	<a href="#">CWE-79</a>	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
[5]	76.9	<a href="#">CWE-306</a>	Missing Authentication for Critical Function
[6]	76.8	<a href="#">CWE-862</a>	Missing Authorization
[7]	75.0	<a href="#">CWE-798</a>	Use of Hard-coded Credentials
[8]	75.0	<a href="#">CWE-311</a>	Missing Encryption of Sensitive Data
[9]	74.0	<a href="#">CWE-434</a>	Unrestricted Upload of File with Dangerous Type
[10]	73.8	<a href="#">CWE-807</a>	Reliance on Untrusted Inputs in a Security Decision
[11]	73.1	<a href="#">CWE-250</a>	Execution with Unnecessary Privileges
[12]	70.1	<a href="#">CWE-352</a>	Cross-Site Request Forgery (CSRF)
[13]	69.3	<a href="#">CWE-22</a>	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
[14]	68.5	<a href="#">CWE-494</a>	Download of Code Without Integrity Check
[15]	67.8	<a href="#">CWE-863</a>	Incorrect Authorization
[16]	66.0	<a href="#">CWE-829</a>	Inclusion of Functionality from Untrusted Control Sphere
[17]	65.5	<a href="#">CWE-732</a>	Incorrect Permission Assignment for Critical Resource
[18]	64.6	<a href="#">CWE-676</a>	Use of Potentially Dangerous Function
[19]	64.1	<a href="#">CWE-327</a>	Use of a Broken or Risky Cryptographic Algorithm
[20]	62.4	<a href="#">CWE-131</a>	Incorrect Calculation of Buffer Size
[21]	61.5	<a href="#">CWE-307</a>	Improper Restriction of Excessive Authentication Attempts
[22]	61.1	<a href="#">CWE-601</a>	URL Redirection to Untrusted Site ('Open Redirect')
[23]	61.0	<a href="#">CWE-134</a>	Uncontrolled Format String
[24]	60.3	<a href="#">CWE-190</a>	Integer Overflow or Wraparound
[25]	59.9	<a href="#">CWE-759</a>	Use of a One-Way Hash without a Salt

940 CWE's in total – another 37 pages like this

# Example From a Different Domain

J2EE Bad Practices: Direct Management of Connections	
<b>Weakness ID:</b> 245 ( <i>Weakness Variant</i> )	<b>Status:</b> Draft
<b>Description</b>	
<b>Description Summary</b>	
The J2EE application directly manages connections, instead of using the container's connection management facilities.	
<b>Time of Introduction</b>	
<ul style="list-style-type: none"><li>• Architecture and Design</li><li>• Implementation</li></ul>	
<b>Applicable Platforms</b>	
<b>Languages</b>	
Java	
<b>Common Consequences</b>	
<b>Scope</b>	<b>Effect</b>
Other	<b>Technical Impact:</b> <i>Quality degradation</i>
<b>Demonstrative Examples</b>	
<b>Example 1</b>	
In the following example, the class DatabaseConnection opens and manages a connection to a database for a J2EE application. The method openDatabaseConnection opens a connection to the database using a DriverManager to create the Connection object conn to the database specified in the string constant CONNECT_STRING.	
<i>Example Language: Java</i> <span style="float: right;"><i>(Bad Code)</i></span>	
<pre>public class DatabaseConnection {     private static final String CONNECT_STRING = "jdbc:mysql://localhost:3306/mysqlpdb";     private Connection conn = null;      public DatabaseConnection() {     }      public void openDatabaseConnection() {         try {             conn = DriverManager.getConnection(CONNECT_STRING);         } catch (SQLException ex) {...}     }      // Member functions for retrieving database connection and accessing database     ... }</pre>	

The use of the DriverManager class to directly manage the connection to the database violates the J2EE restriction against the direct management of connections. The J2EE application should use the web application container's resource management facilities to obtain a connection to the database as shown in the following example.

# Why So Many Vulnerabilities?

1. Writing secure code is hard
2. Software engineers are poorly trained in security (but not you guys!)
3. Economic incentives at software companies do not prioritize doing things right

# Writing Secure Code is Hard

- 940 CWEs (and counting)
- Large numbers of different categories of things to do wrong
  - Hard to know about all of them
- Humans are error prone at best
  - Most of us write a noticeable number of bugs/kloc
  - Significant fraction of bugs turn out to be exploitable
  - Testing all code-paths is both theoretically and practically impossible

# Software Engineers Untrained in Security

Rank	School name	Score
#1	Carnegie Mellon University Pittsburgh, PA	5.0
#1	Massachusetts Institute of Technology Cambridge, MA	5.0
#1	Stanford University Stanford, CA	5.0
#1	University of California–Berkeley Berkeley, CA	5.0
#5	Cornell University Ithaca, NY	4.6
#5	University of Illinois–Urbana-Champaign Urbana, IL	4.6
#7	University of Washington Seattle, WA	4.5
#8	Princeton University Princeton, NJ	4.4
#8	University of Texas–Austin Austin, TX	4.4
#10	Georgia Institute of Technology Atlanta, GA	4.3

You are here →

Source: <http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools/top-science-schools/computer-science-rankings>

# Undergraduate Security Course @Top 10

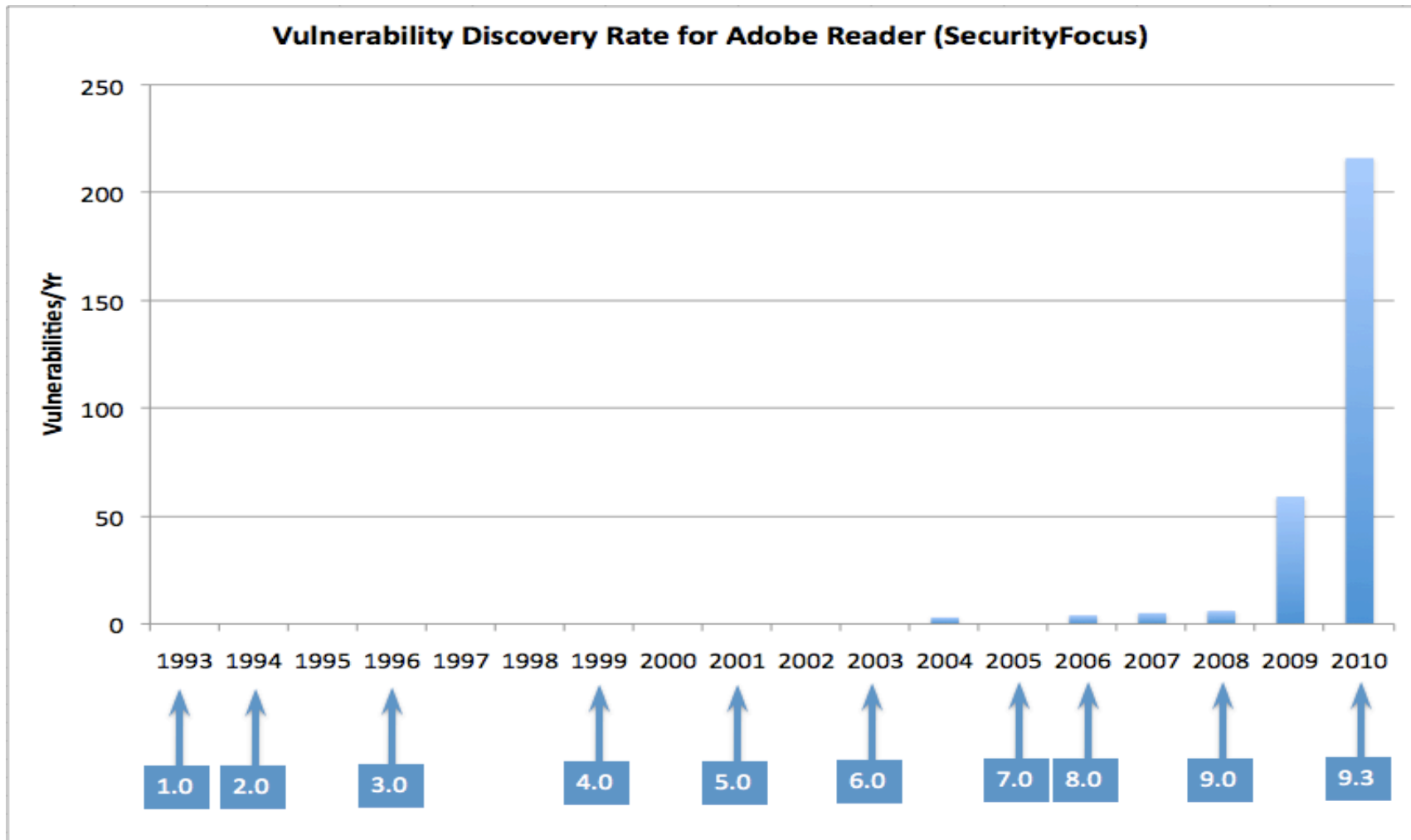
	<b>Available</b>	<b>Required</b>
Carnegie Mellon	Yes	No
MIT	Yes	No
Stanford	Yes	No
Berkeley	Yes	No
Cornell	Yes	No
Illinois/Urbana-Champaign	Yes	No
University of Washington	Yes	No
Princeton	Yes	No
UT Austin	Yes	No
Georgia Tech	Yes	No
Purdue*	Yes	No
UC Davis*	Yes	No



# Economic Incentives

- Most tech market categories have a winner-take-all structure
  - Microsoft in operating systems and office suites
  - Google in search
  - Apple/Google in smartphones
  - Oracle in databases
  - Cisco in routers/switches
- Company executives/VCs know this
  - So very strong pressure to ship code fast
    - Dominate the market, then solve problems later

# Typical Life of a Vulnerability



# Economic Incentives

- Not writing vulnerabilities in the first place is
  - Hard
  - Therefore expensive and slow
- Finding vulnerabilities takes lots of QA/test
  - Also, expensive and slow
- So companies have a strong incentive to not worry about it **too** much
  - Fix it later, when it becomes a PR problem

# In Today's New York Times...

## F.T.C. Says Webcam's Flaw Put Users' Lives on Display

By EDWARD WYATT

Published: September 4, 2013 | [Comment](#)

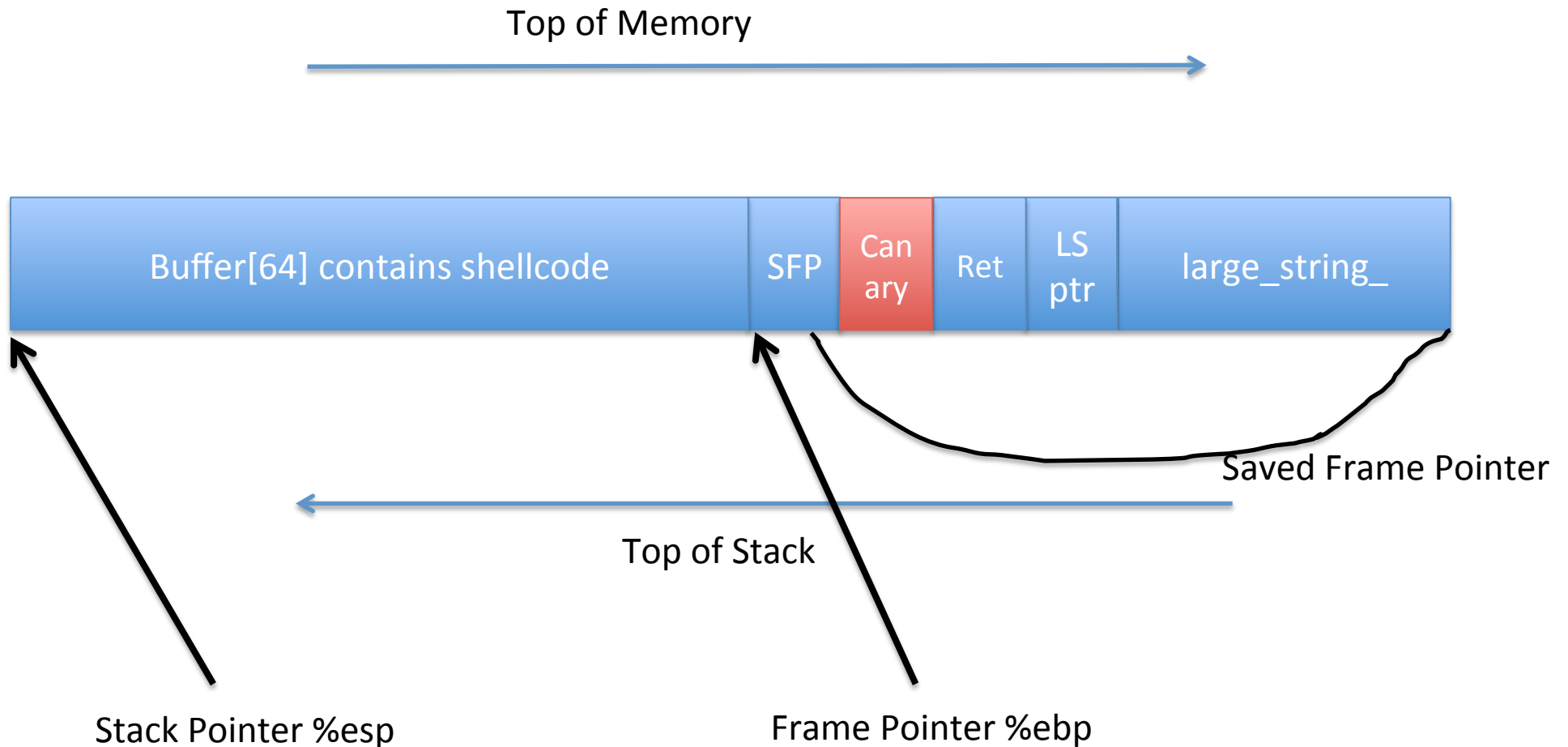
WASHINGTON — The so-called Internet of Things — digitally connected devices like appliances, cars and medical equipment — promises to make life easier for consumers. But regulators are worried that some products may be magnets for hackers.

According to the F.T.C., the company, TRENDnet, told customers that its products were “secure,” marketing its cameras for home security and baby monitoring. In fact, the devices were compromised. The commission said a hacker in January 2012 exploited a security flaw and posted links to the live feeds, which “displayed babies asleep in their cribs, young children playing and adults going about their daily lives.”

In detailing the security lapses, the commission said the company transmitted customers' login information over the Internet in clear, readable text rather than encrypting the data. It also said TRENDnet's mobile application, which allows customers to control the home camera from a smartphone, did not properly protect users' credentials. When the company became aware of the flaws, it uploaded a software patch to its Web site and tried to alert customers.

As part of the case, TRENDnet agreed to sanctions that include a 20-year security-compliance auditing program. The company also promised not to misrepresent the security of its cameras, the confidentiality of the activity that its devices transmit, or consumers' ability to control the security of the cameras or their recordings. The agency's four current commissioners voted unanimously for the sanctions.

# Canary-based Overflow Defense



Not invincible. Eg <http://phrack.org/issues.html?issue=56&id=5>