# OFFENSIVE COMPUTER SECURITY 2.0

**(aka “OCS 2.0” or “OCS2015_REVAMP”)**

**Free Opencourseware**

Introduction Lecture
W. Owen Redwood, Ph.D.
http://hackallthethings.com/

# This class

- Structured as a hands-on survey of topics
    - Topics hand picked from a variety of expert resources
    - Hands on through homework assignments
- Will transform n00bs into ninjas in **16 weeks**
    - If you get a decent grade
    - You are required to find a 0-day
        - (exploitable memory corruption bug)
            - i.e. one that can gain remote code execution (RCE) on a box

# This class

- 15 week, 3 credit hour graduate class
    - 26 Lectures (2 per week)
    - 10 Homeworks (Very hands on)
        - one is to find a 0-day
    - 2 Exams + Final Exam (3 total)
- Pre-Reqs (n00b friendly):
    - Familiar with C/C++
    - Familiar with Assembly
    - Basic grasp of security concepts is useful

# HISTORY

Originally I created it at Florida State university, under the advising of Prof Xiuwen Liu:

- Spring 2013: “Offensive Security”
  - https://www.cs.fsu.edu/~redwood/OffensiveSecurity/
- Spring 2014: “Offensive Computer Security”
  - https://www.cs.fsu.edu/~redwood/OffensiveComputerSecurity/
- Still taught at FSU by Prof Xiuwen Liu
  - https://www.cs.fsu.edu/~liux/
- OCS has spread to over a dozen other universities:
  - Sometimes lead by faculty
  - Mostly lead by students (Via clubs  / CTF teams)

# WHAT’S NEW in the REVAMP?

- Better exercises in homeworks
- Polished lecture videos and slides
- New content:
  - Windows internals (brought back)
  - more ROP (new HW on it)
  - more on the VLC 0-day hunt exercise

AND IT’S STILL **OPEN SOURCE!**

- Happy to help other universities with it
  - as well as other curriculum development

# The Instructors

## W. Owen Redwood (sk4ld)

- PhD FSU 2016
  - Cyber Physical Systems Vulnerability Research
- MS FSU 2010
- BS Georgia Tech 2008

## Specialties:

- Bug hunting, reverse engineering, exploit development/mitigation, and other vulnerability research on embedded and cyber physical systems;
- Computer Architecture & Virtualization research (desktop & embedded);
- Botnet design, analysis, and reverse engineering research;
- Honeypot research;
- etc …

DISCLAIMER:
MY OPINIONS ARE MY OWN AND NOT ANY OF  MY EMPLOYERS’

# The Instructors

## Hahna Kane Latonick (hakatak)

- Worked across several private industry companies and the U.S. government for the past 9 years.
- Alumna of Drexel University and Swarthmore College
  - BS/MS Computer Engineering
  - Minor Mathematics
- Subject matter expert in information systems security, vulnerability research and computer network operations.
- She has also competed as a DEFCON CTF finalist in 2014.

DISCLAIMER: MY OPINIONS ARE MY OWN AND NOT ANY OF  MY EMPLOYERS’

# The Website & other details

Hosted at:

http://hackallthethings.com/

Self Paced

- Originally designed for 2 lectures per week + 10 homeworks over 15 weeks. (usually each homework takes 3 weeks)

# CURRICULUM

| TOPIC | # of Lectures<br>*(including bonus lectures)* |
|---|---|
| C/C++ Bugs & Code Auditing | 3 |
| Linux / Windows Internals & Rootkits | 3 |
| Reverse Engineering x86 | 2 |
| General Vulnerability Research (static/dynamic analysis) | 2 |
| Exploit Development (shellcode -> ROP) | 7 |
| Web Application Hacking | 4 |
| TCP/IP network security & hacking | 2 |
| Memory Forensics & Incident Response | 1* (theme throughout many other lectures) |
| Social Engineering & Physical Security | 2 |

# Grade Breakdown

Homeworks 55%

- 10 Homeworks (5.5% each) are hands on exposure to topics, and are mini-project like

Midterms 30%

- Midterms 1 and 2 will cover the meat of the class

Final Exam 15%

# Grading Policy

Individual work only:

- On every homework, assignment, and project
- Do not share answers

Grading is based off of your:

1. Ability to utilize the required skills
2. Communicate what you did, what happened, and etc...

# Midterm 1 and 2

Midterm 1 = Week 6

Midterm 2 = Week 13

Final = Week 15 or 16

# Extra Credit

Extra credit should be granted for:

- Participation in any capture the flag games
    - See your university’s CTF team or Cybersecurity club to get involved.
    - Should be weighed upon difficulty of problems solved, and your level of participation
        - 500 point problems should receive serious extra credit
    - see https://ctftime.org/

# Don’t Satisfy the Prereqs?

Pre-Reqs (n00b friendly):

- Familiar with C/C++
- Familiar with Assembly
- Basic grasp of security concepts is useful

...

*No experience with Assembly or C?* Go through the Programming From The Ground Up (FREE) book:

https://savannah.nongnu.org/projects/pgubook/

# Who this class is for

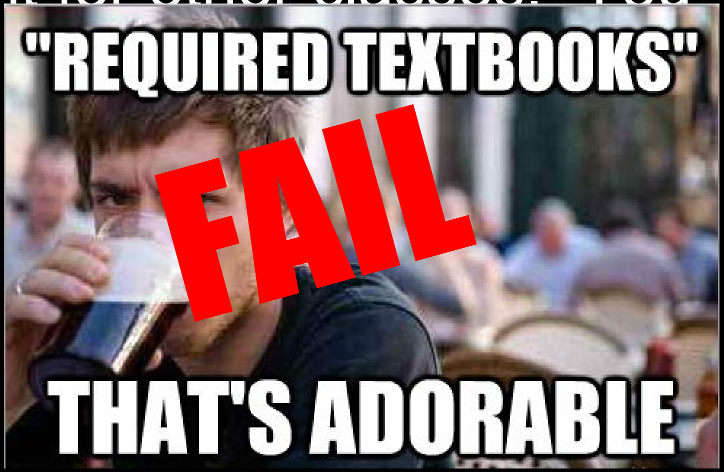
**Anyone** who wants to become:

- Better Defenders
- Incident Responders
- Penetration Testers (aka pentesting)
- Security Professionals
- Forensics Professionals
- Vulnerability Researchers (aka VR)
- and so on

We focus mainly on Pentesting and VR

# Who this class is NOT for

- Students who are completely new to information security
  - you will fail this class
- Lazy people who don't do the assigned reading or homework.
  - I don't care if you don't do it for other classes. You better do it for this one.
  - **Tests will cover reading material not covered in class**



# The books

**Hacking: The Art of Exploitation** 2nd edition- Jon Erickson (AKA “The AOE”)

- 2008 book (will be relevant for a very long time)
- HANDS ON approach to all the material, rich with source code, comes with CD
- *Is going to be our main textbook*

*The Web Application Hacker’s Handbook 2nd edition- Dafydd Stuttard*

- *2012 book*
- *2nd half of the class*

# Virtual Machines

**You should know how to use them.**

The Live CD that comes with Hacking the Art Of Exploitation is ideal for experimentation.

- Set up a VM (I suggest Virtual Box) with .iso of the live cd.
- You will use this VM to do some of the homeworks

# The books used to create this class

An incomplete list:

- Hacking: The Art of Exploitation
- The Web Application Hacker’s Handbook
- The Shellcoder's Handbook (2nd ed)
- Windows Internals series
- Metasploit: The Penetration Testers Guide
- Practical Malware Analysis
- The Art of Debugging with GDB, DDD, and Eclipse
- The Rootkit ARSENAL
- Secure Coding in C and C++
- Exploratory Software Testing
- Writing Security Tools & Exploits

# Motivations

1) **Security/Architecture Analogy:**

- “Teaching only defense is like teaching blind people to be architects”
  - It will all “feel secure” but not be secure.
  - You are unable to see cracks, weaknesses, and faults in the *foundation, building, and design*
  - Offense measures the defense in a very binary manner
    - Did it spawn a shell or DoS it? (Yes/No)

# Motivations

**2)** Most security education focuses heavily on Cryptography…

but...

"One of the most dangerous aspects of cryptology ..., is that you can almost measure it." -Matt Blaze (Afterword in Bruce Schneier's "Applied Cryptography")

***To break into most systems, you don't have to break crypto.***

- Cue OPM hack joke

# Motivations

**3)** “A Fool With a Tool is Still a Fool” - David A. Wheeler.

- “It’s a mistake to think that analysis tools (like flawfinder) are a substitute for security training and knowledge”
- *Too many pentests go like this:*
  - a. *Pentester uses script utility on target, finds no ways in.*
  - b. *Pentester reports to client “You are safe”.*
  - c. *Client gets hacked weeks later.*
- This class teaches the fundamentals and essential knowledge, and is agnostic of tools

# 4) The COST of "Reactive" Security

Consequences of exploitation 2013

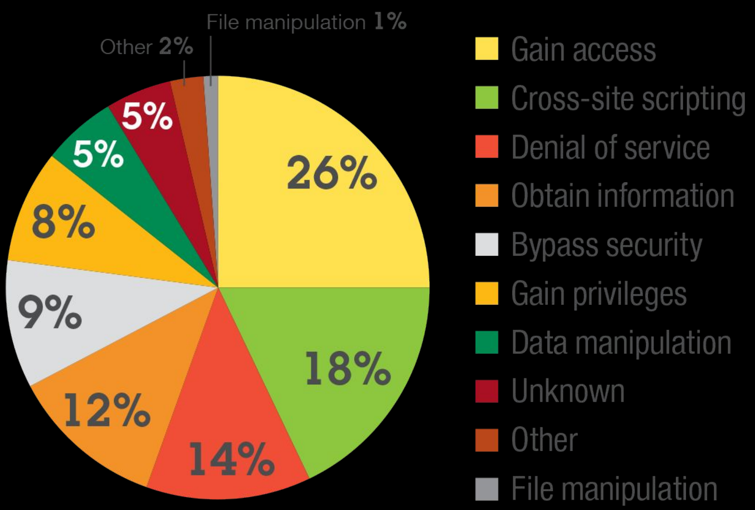
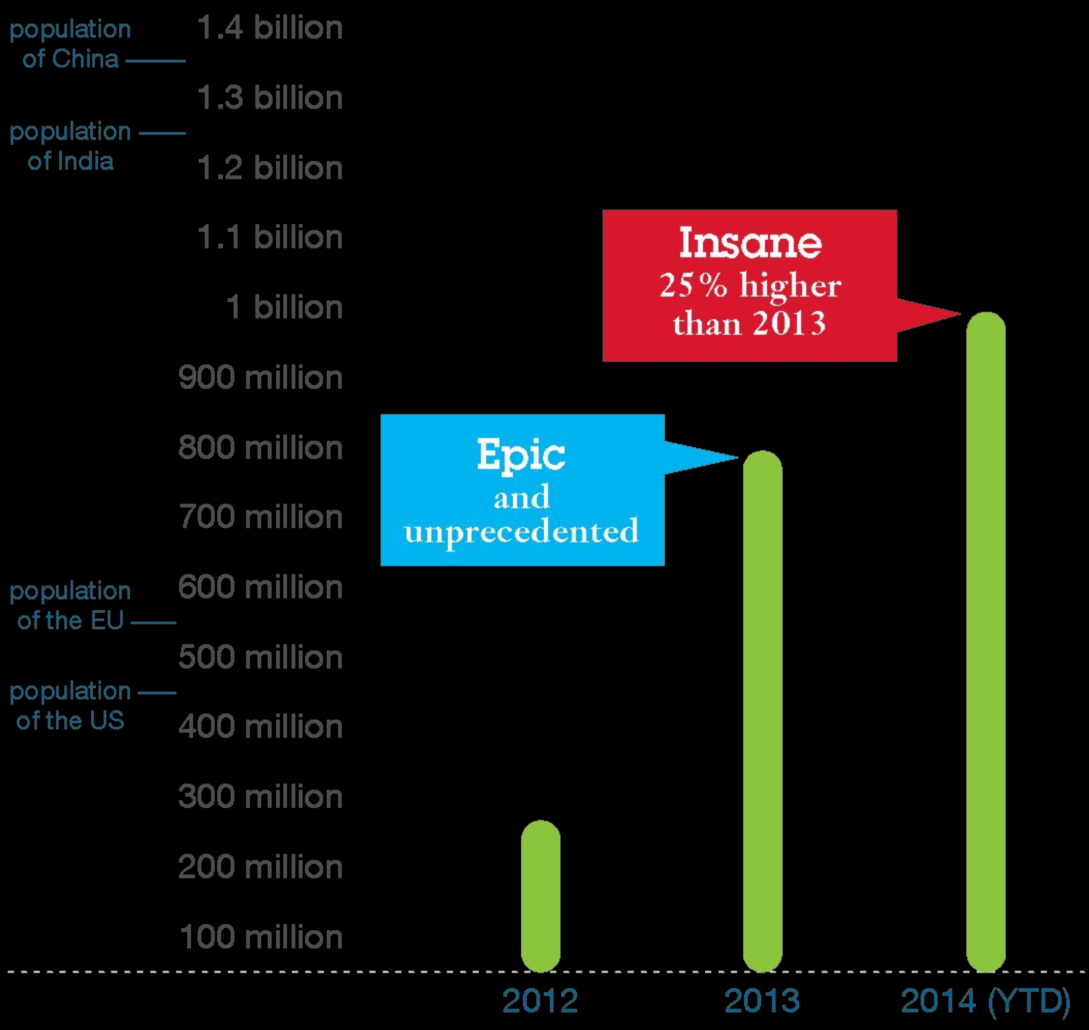


*Figure 12. Consequences of exploitation 2013*

Source: IBM X-Force® Research and Development

## Total records leaked by year

compared to estimated population sizes



Epic and unprecedented

Insane 25% higher than 2013

*Figure 1. Total records leaked by year, compared to estimated population sizes*

Source: IBM X-Force® Research and Development

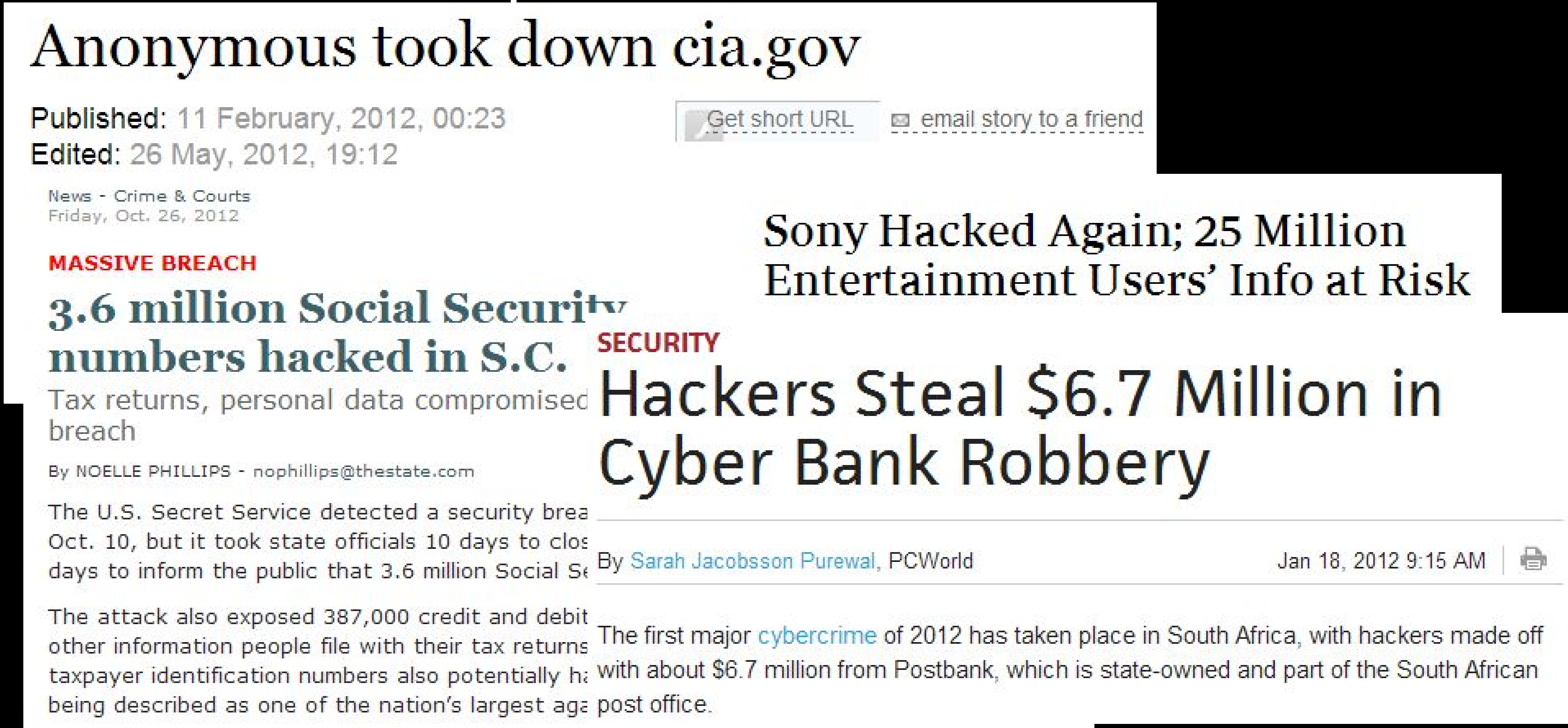
# Motivations (Pen testing)

- Pen testing is fun
- you get paid to hack
  - and think like a bad guy

And people look at you like ^

# Motivations (Incident Response)

- Networks get hacked
- Incident responders are in HIGH DEMAND



## Anonymous took down cia.gov

Published: 11 February, 2012, 00:23
Edited: 26 May, 2012, 19:12

Get short URL   email story to a friend

News - Crime & Courts
Friday, Oct. 26, 2012

MASSIVE BREACH

## 3.6 million Social Securi
## numbers hacked in S.C.

Tax returns, personal data compromised
breach

By NOELLE PHILLIPS - nophillips@thestate.com

The U.S. Secret Service detected a security brea
Oct. 10, but it took state officials 10 days to clos
days to inform the public that 3.6 million Social Se

The attack also exposed 387,000 credit and debit
other information people file with their tax returns
taxpayer identification numbers also potentially ha
being described as one of the nation's largest aga

## Sony Hacked Again; 25 Million Entertainment Users' Info at Risk

SECURITY

## Hackers Steal $6.7 Million in Cyber Bank Robbery

By Sarah Jacobsson Purewal, PCWorld   Jan 18, 2012 9:15 AM

The first major cybercrime of 2012 has taken place in South Africa, with hackers made off with about $6.7 million from Postbank, which is state-owned and part of the South African post office.

# Introduction

# What this class is about

1. Security Assessment
2. Risk Assessment

   RISK = THREAT x VULNERABILITY

   "Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization"

Source: http://pauldotcom.com/IntroToPenTesting.pdf

# This thing we call "Security"

Security is only appreciated when threats are visible, and are stopped

# About Security Employees IRL

- Only get negative press
  - attacks make them look bad
  - good security doesn’t get noticed, is only inconvenient
  - Often block development  work / projects
- Aren’t incentivized properly
  - Only objective is to respond to attacks and manage the attack surface
    - averse to expanding the attack surface
- *Largely reactive…*

# About Security Employees IRL

- *My opinion:*
    - Should be more proactive!
        - “Proactive Security”:
            - Looking for problems (without causing them)
            - penetration testing
            - vulnerability assessment
            - code review / audit
            - red teaming
            - sharing latest and greatest security research / news
        - Security engineers should be encouraged and rewarded for proactive security measures.
            - Because it’s 2015 and everyone gets hacked.

# It is time to wake up

http://www.digitalattackmap.com/

But that ^ is just DDoS and **NOT “hacking”**

- We are going to thoroughly explore the art of exploitation
  - art of gaining unauthorized access
    - So we can prevent it

# Pen Testing & Incident Response

Both require a great deal of offensive knowledge

"Dark Arts"

But Pen Testing = proactive (hopefully)

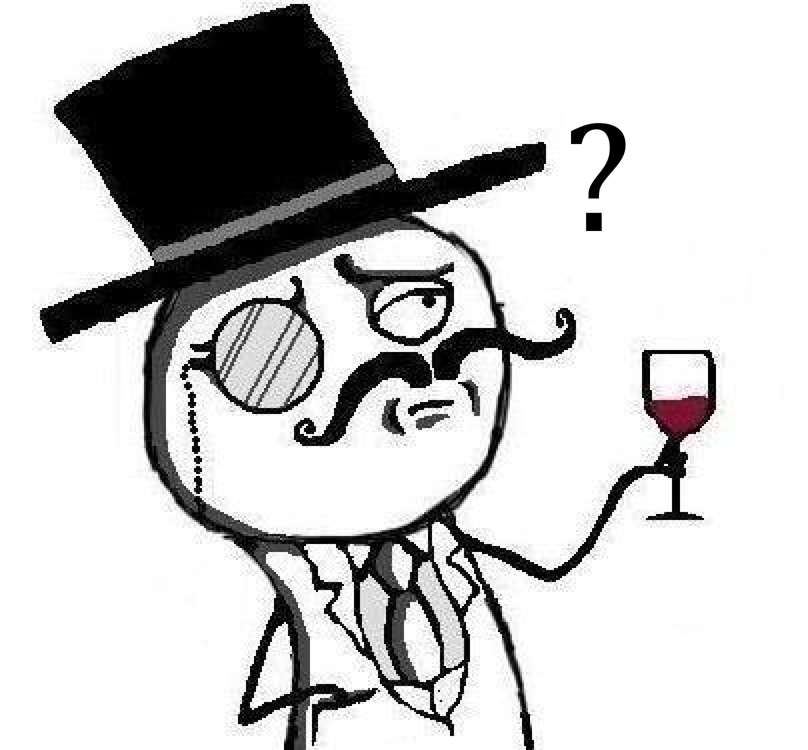
and Incident Response = reactive

# Hacking versus Penetration Testing

Hacking, *AKA cracking, etc..*

Penetration Testing, *AKA red teaming, security assessment, etc..*

What's the difference?



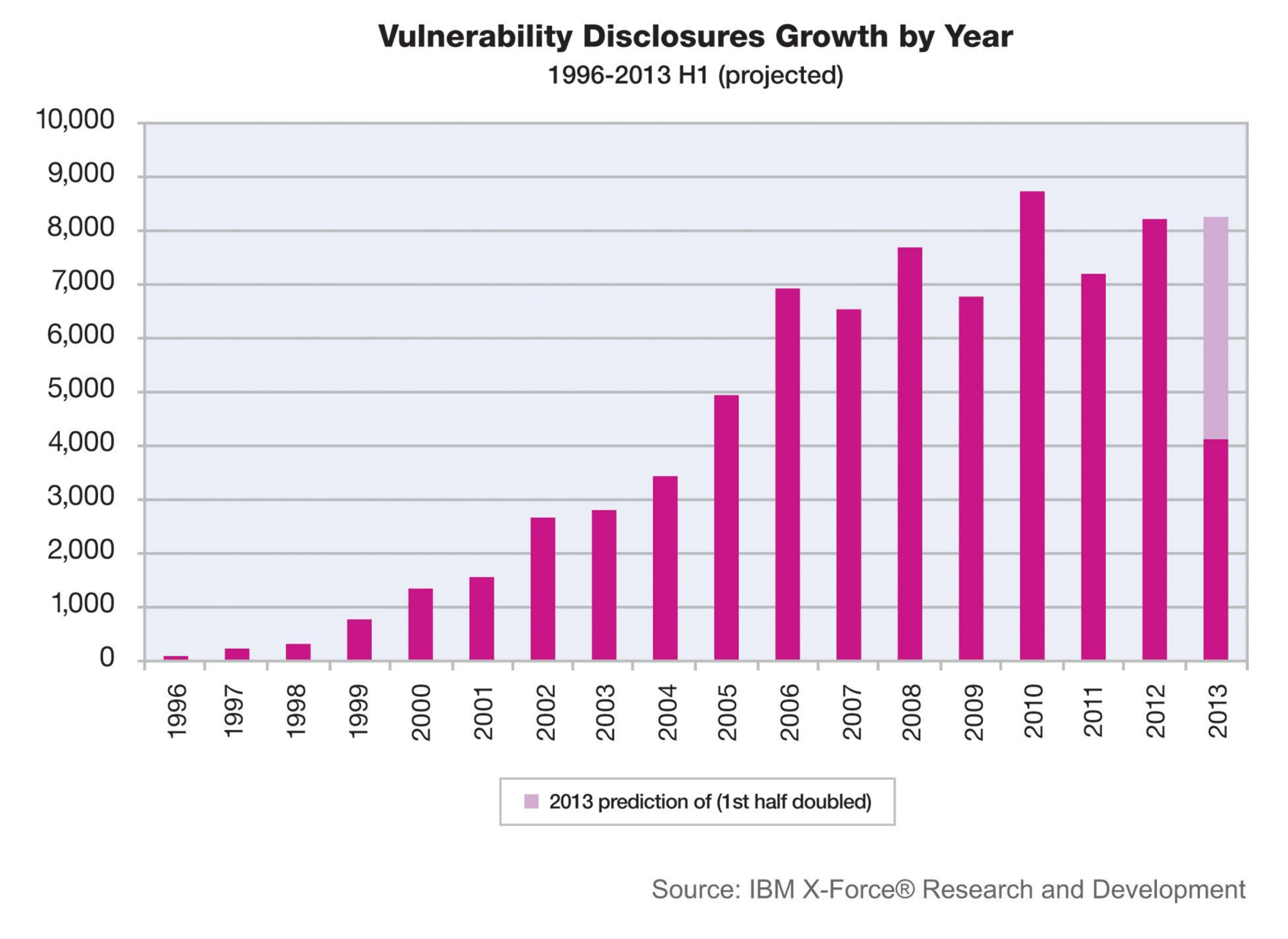
# PERMISSION

really thats it.

Without permission, its ILLEGAL

# Lets talk Vulnerabilities

# Total Vulnerabilities Disclosed



Source: IBM X-Force® Research and Development

# Top 10 vs everyone else

Vulnerability disclosures by large enterprise software vendors

2013 and 1H 2014

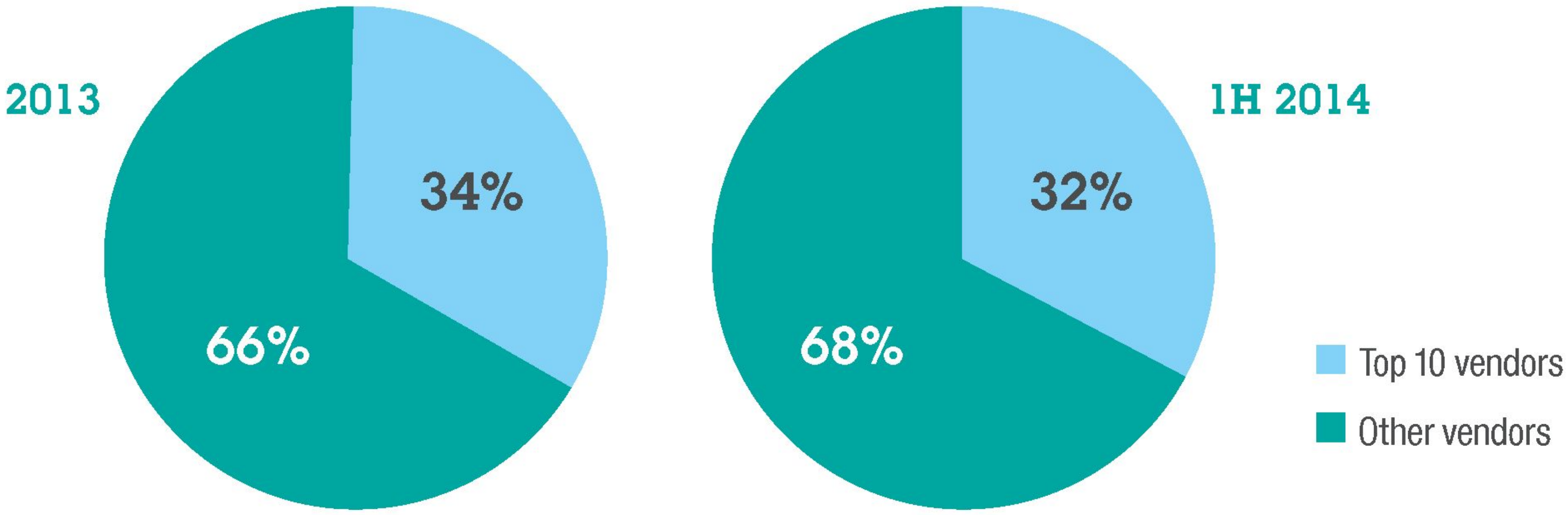
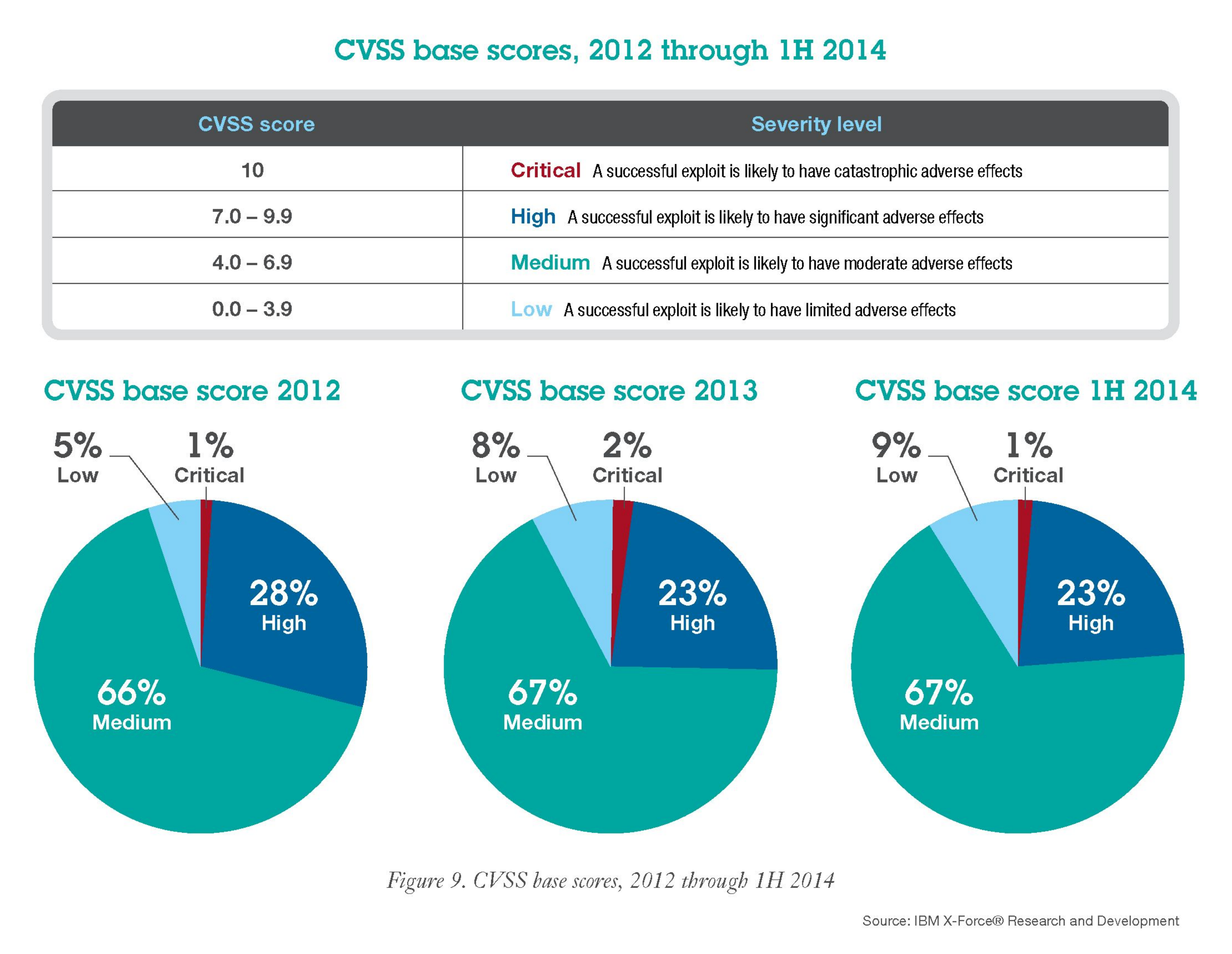


*Figure 7. Vulnerability disclosures by large enterprise software vendors, 2013 and 1H 2014*

Source: IBM X-Force® Research and Development

# Vulnerability Severity?



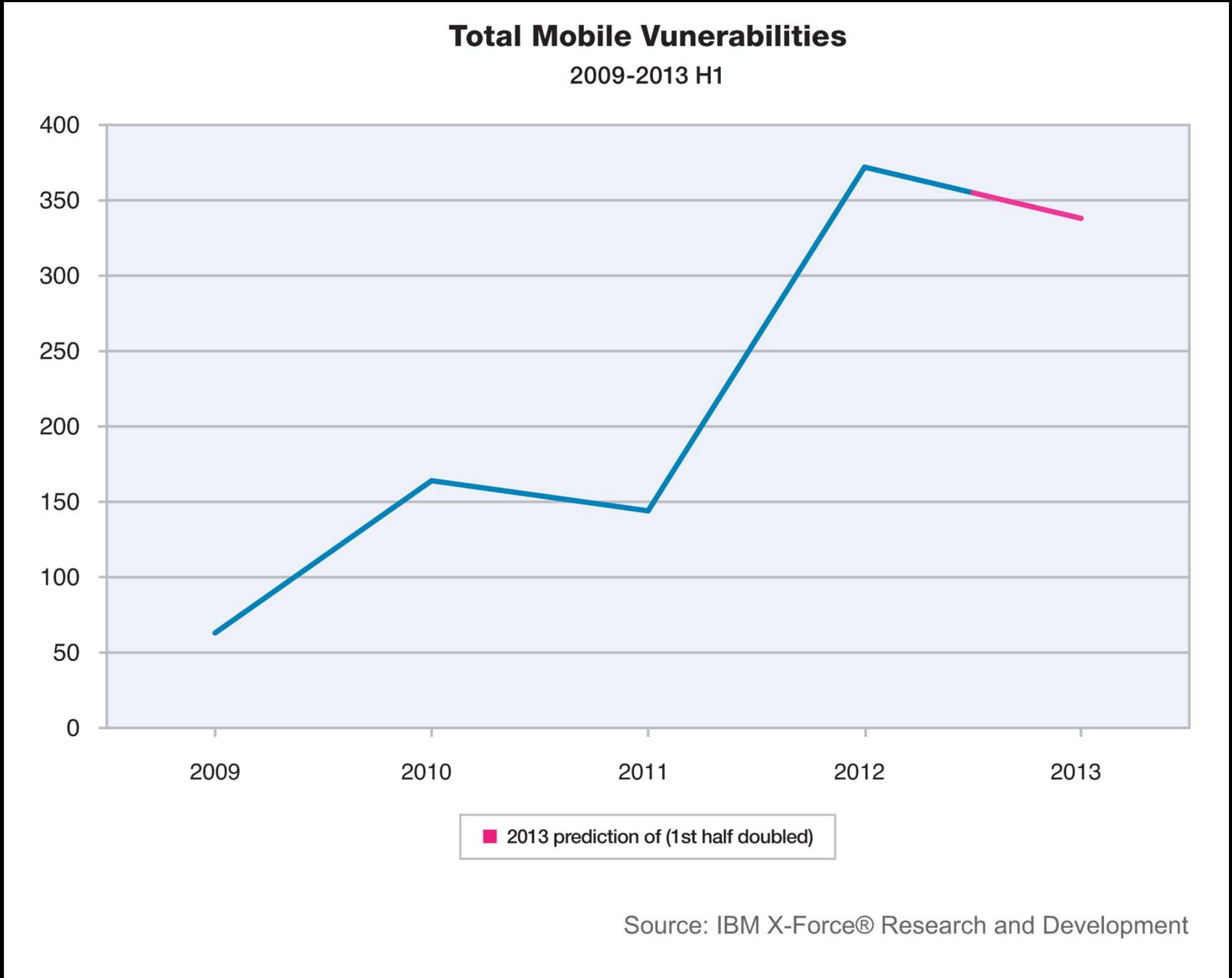
CVSS base scores, 2012 through 1H 2014

| CVSS score | Severity level |
|---|---|
| 10 | **Critical** A successful exploit is likely to have catastrophic adverse effects |
| 7.0 – 9.9 | **High** A successful exploit is likely to have significant adverse effects |
| 4.0 – 6.9 | **Medium** A successful exploit is likely to have moderate adverse effects |
| 0.0 – 3.9 | **Low** A successful exploit is likely to have limited adverse effects |

*Figure 9. CVSS base scores, 2012 through 1H 2014*

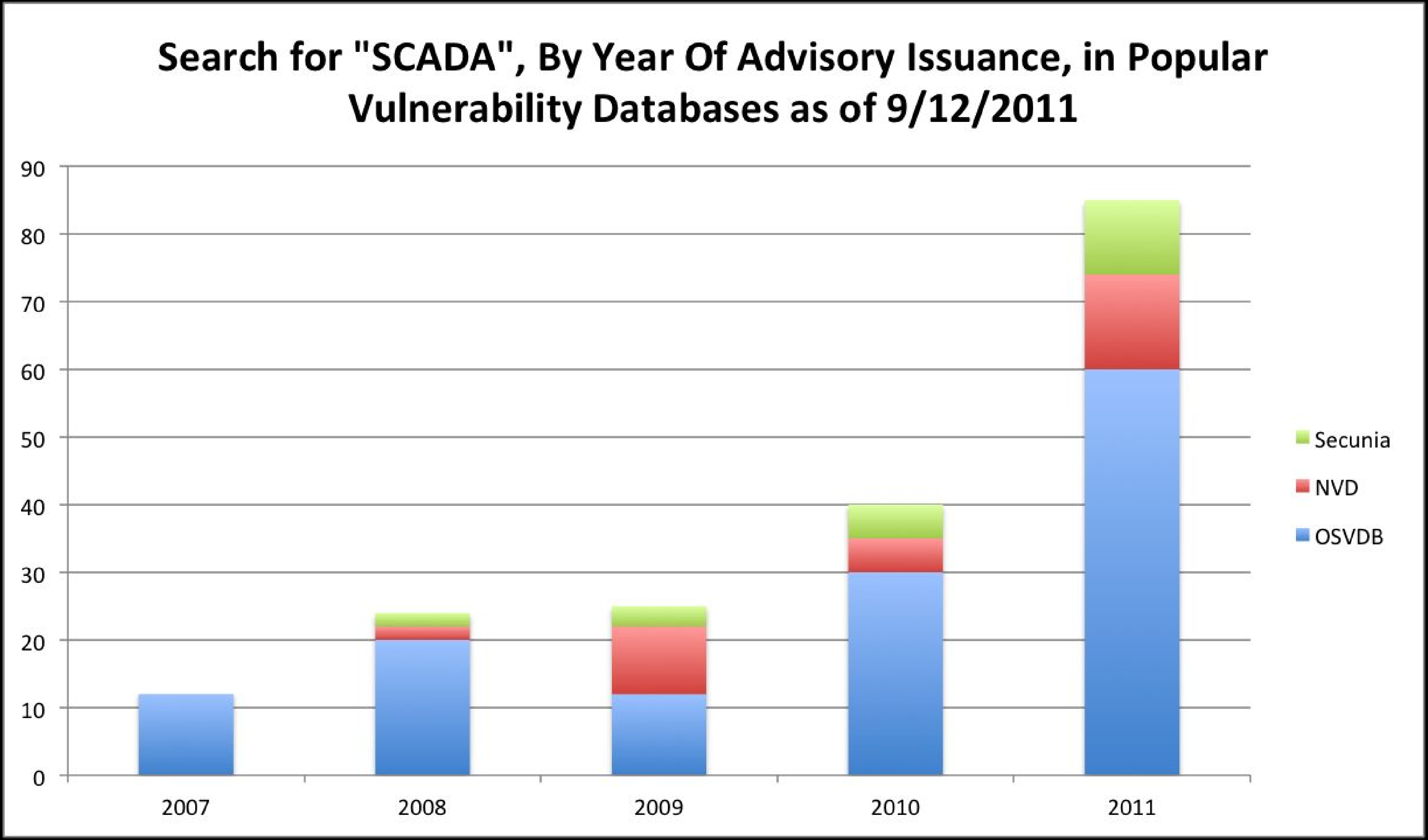
Source: IBM X-Force® Research and Development

# Vulnerabilities (Mobile)



Source: IBM X-Force® Research and Development

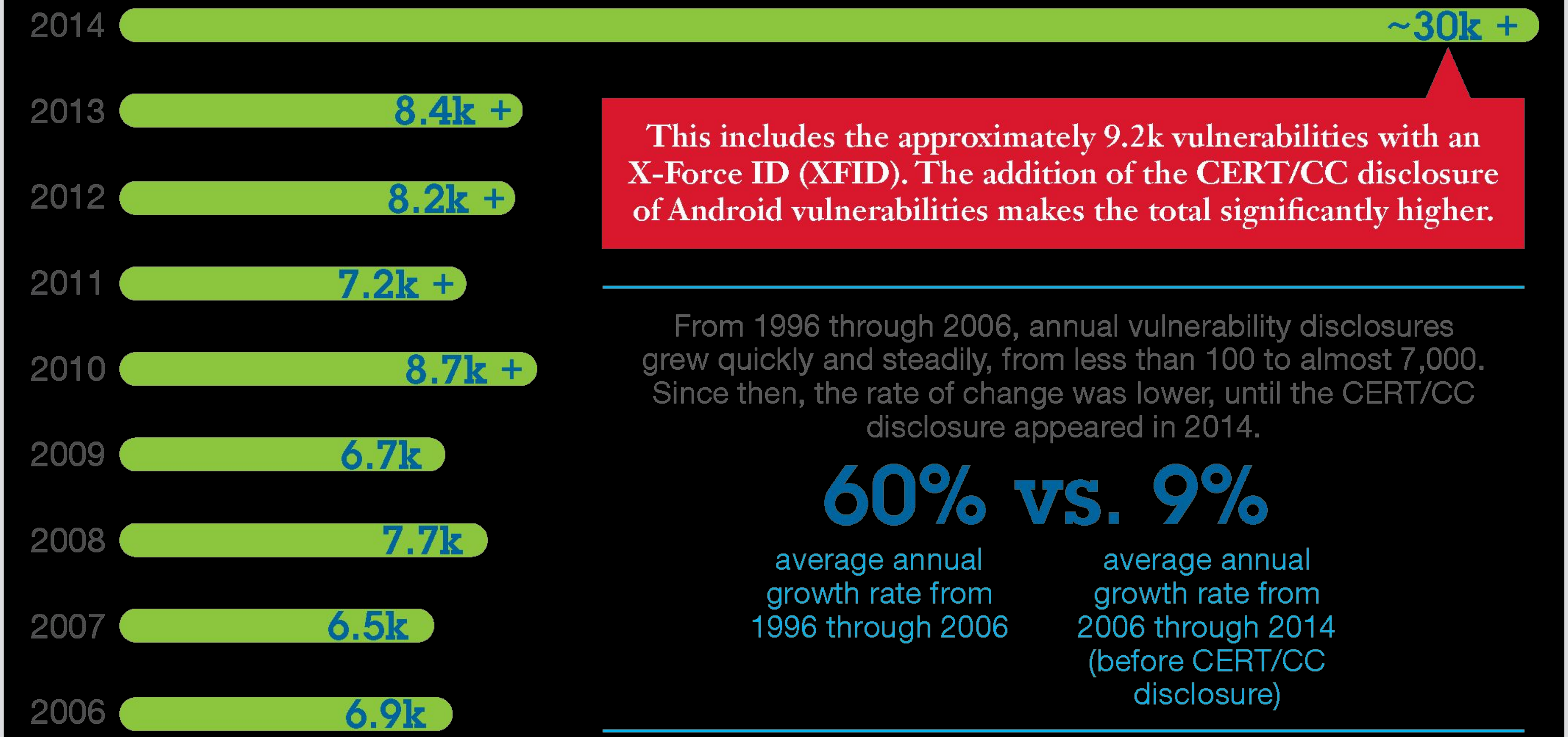
# Vulnerabilities (SCADA)



Source: http://www.energysec.org/blog/quick-and-dirty-vulnerability-trending/

## Vulnerability disclosures growth by year

1996 through 2014



This includes the approximately 9.2k vulnerabilities with an X-Force ID (XFID). The addition of the CERT/CC disclosure of Android vulnerabilities makes the total significantly higher.

From 1996 through 2006, annual vulnerability disclosures grew quickly and steadily, from less than 100 to almost 7,000. Since then, the rate of change was lower, until the CERT/CC disclosure appeared in 2014.

**60% vs. 9%**

average annual growth rate from 1996 through 2006

average annual growth rate from 2006 through 2014 (before CERT/CC disclosure)

*Figure 5. Vulnerability disclosures growth by year, 1996 through 2014*

Source: IBM X-Force® Research and Development

# Post Exploitation Trends?

## Consequences of exploitation 2013

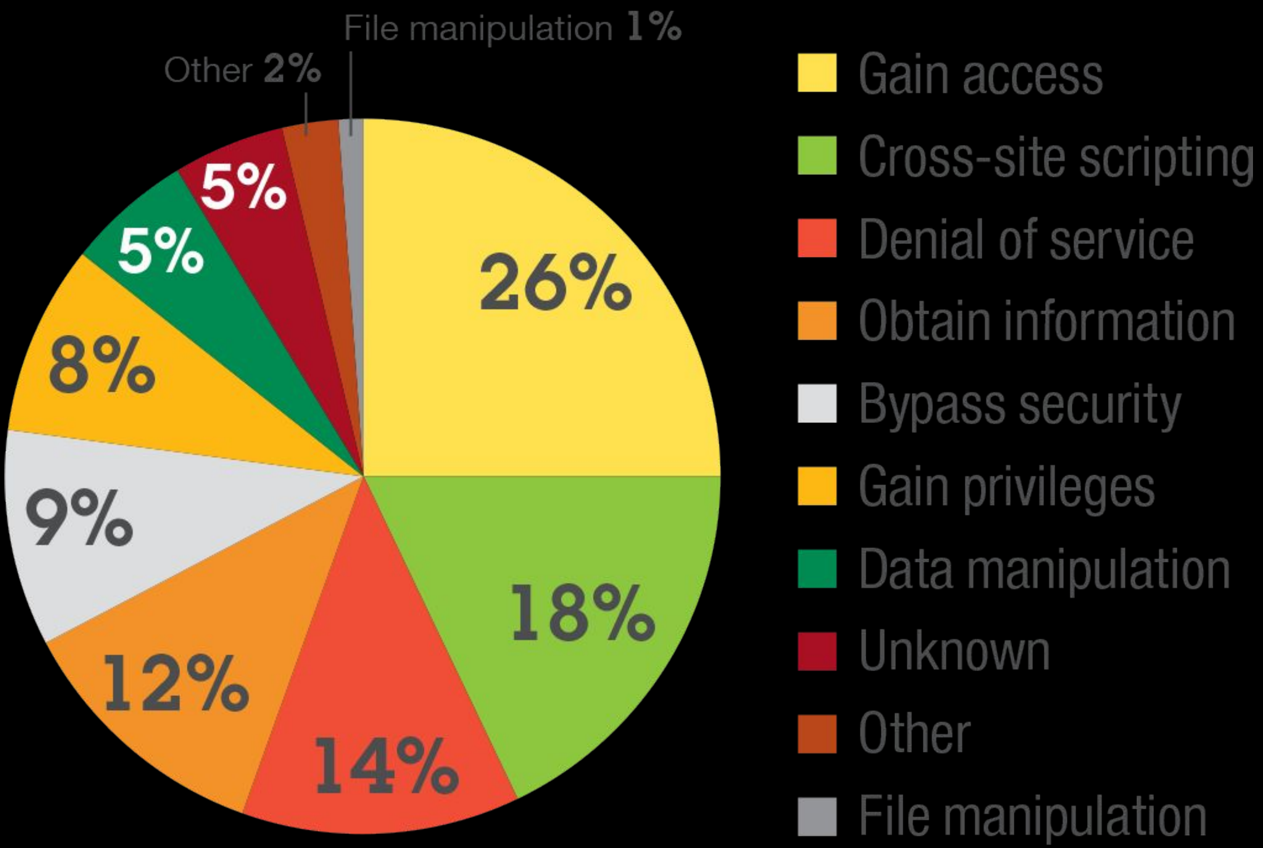


*Figure 12. Consequences of exploitation 2013*

Source: IBM X-Force® Research and Development

# Post Exploitation Trends…

Not including 2015

- OPM hack
  - largest on record…

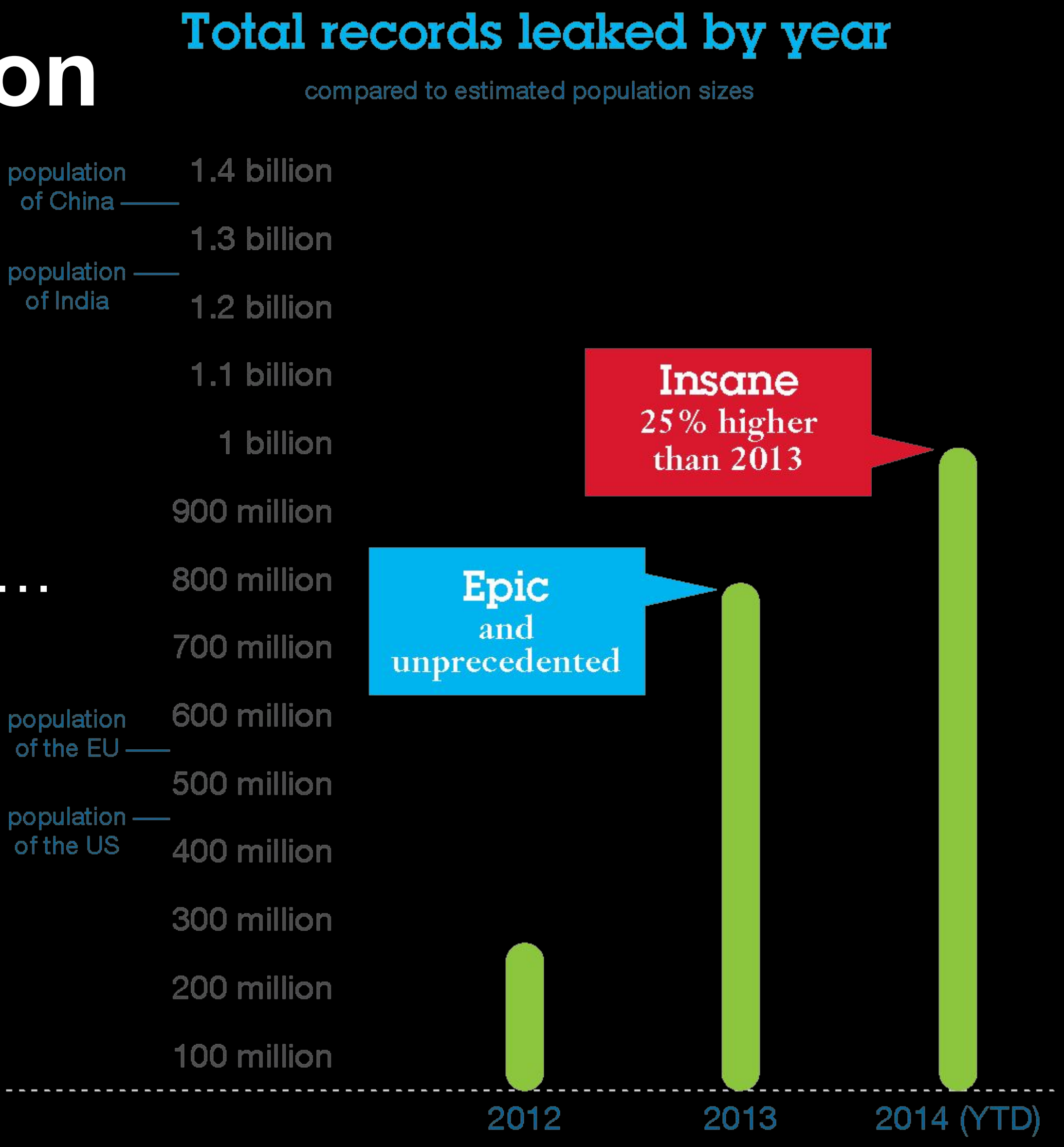
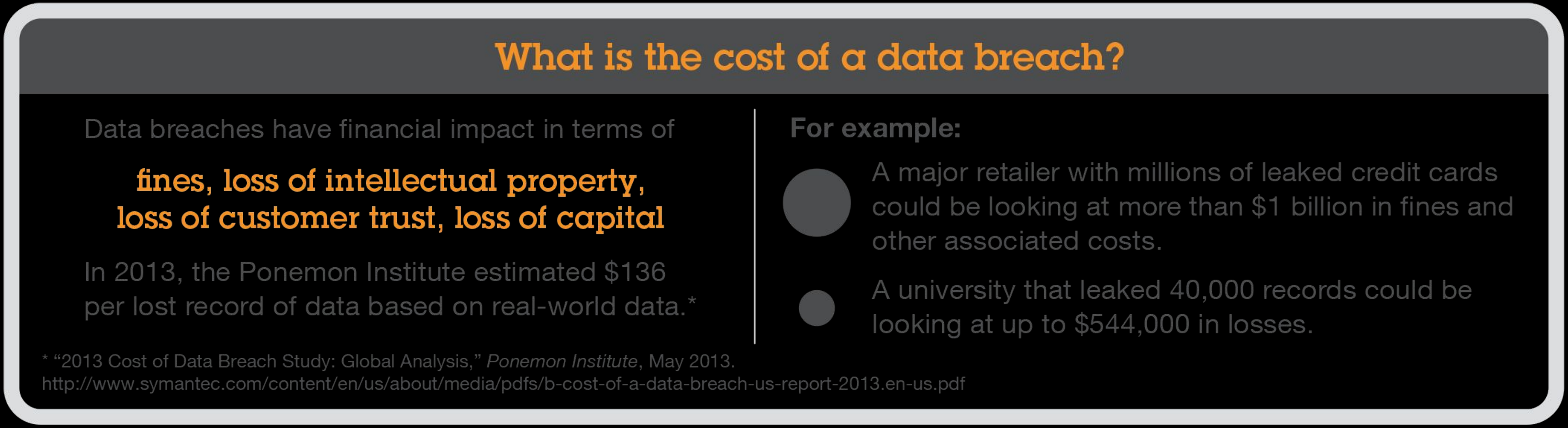


*Figure 1. Total records leaked by year, compared to estimated population sizes*

Source: IBM X-Force® Research and Development

# The Cost of a Data Breach



What is the cost of a data breach?

Data breaches have financial impact in terms of

**fines, loss of intellectual property, loss of customer trust, loss of capital**

In 2013, the Ponemon Institute estimated $136 per lost record of data based on real-world data.*

**For example:**

● A major retailer with millions of leaked credit cards could be looking at more than $1 billion in fines and other associated costs.

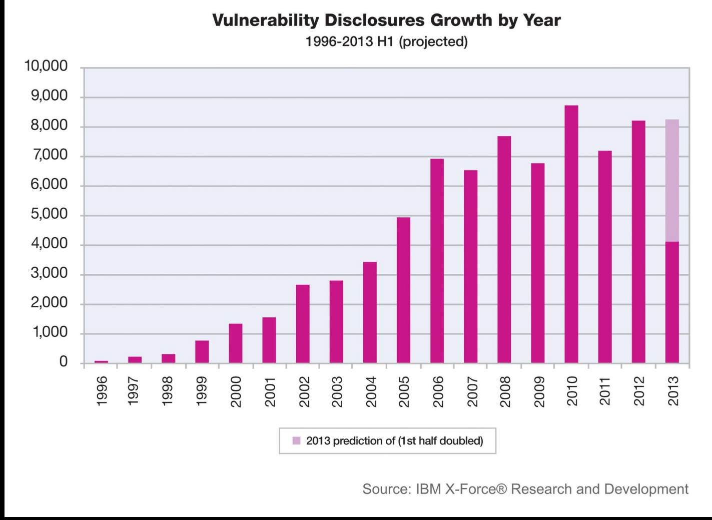
● A university that leaked 40,000 records could be looking at up to $544,000 in losses.

* “2013 Cost of Data Breach Study: Global Analysis,” *Ponemon Institute*, May 2013. http://www.symantec.com/content/en/us/about/media/pdfs/b-cost-of-a-data-breach-us-report-2013.en-us.pdf

*Figure 2b. Sampling of 2013 security incidents by attack type, time and impact*

Source: IBM X-Force® Research and Development

# Trends, Perspective, & Reality....



Vulnerability Disclosures Growth by Year

1996-2013 H1 (projected)

Source: IBM X-Force® Research and Development

Graphs show us getting worse each year!!!!!!

- More vulnerabilities each year!

But…… think:

- were we great at catching attackers / reporting bugs 5, 10, 15 years ago?
- How has technology scaled over these years? Aren’t there way more targets now?

# Ethics and Vulnerability Disclosure

Say you find a security problem

Who do you tell?  And how?

- How would they react?
- Would they sue you? patch it? or ignore it?
- What if you worked hard to find it?
  - should you be rewarded?
- What if they threaten legal action?!?!?!
- What if they do nothing?

# How We Got Here

# History time!  Early on...

- Security mailing lists
- Phrack [http://phrack.org/]
  - 1985-now
  - attacker focused
  - Still has great content
- 99% of people didn't know about security
  - wasn't a real problem

Perception: Vulnerability "Researchers" were evil people, practicing dark magic

# Private Communities

Morris worm (1988)

- Woke people up
- invite only mailing lists rose
  - these also became targets

Main problems:

- Vendors would not acknowledge security problems
- "Buy at your own risk"
  - but mostly only the attackers knew the risks...

But this changed...

# Full Disclosure

Inform everyone, good and bad!

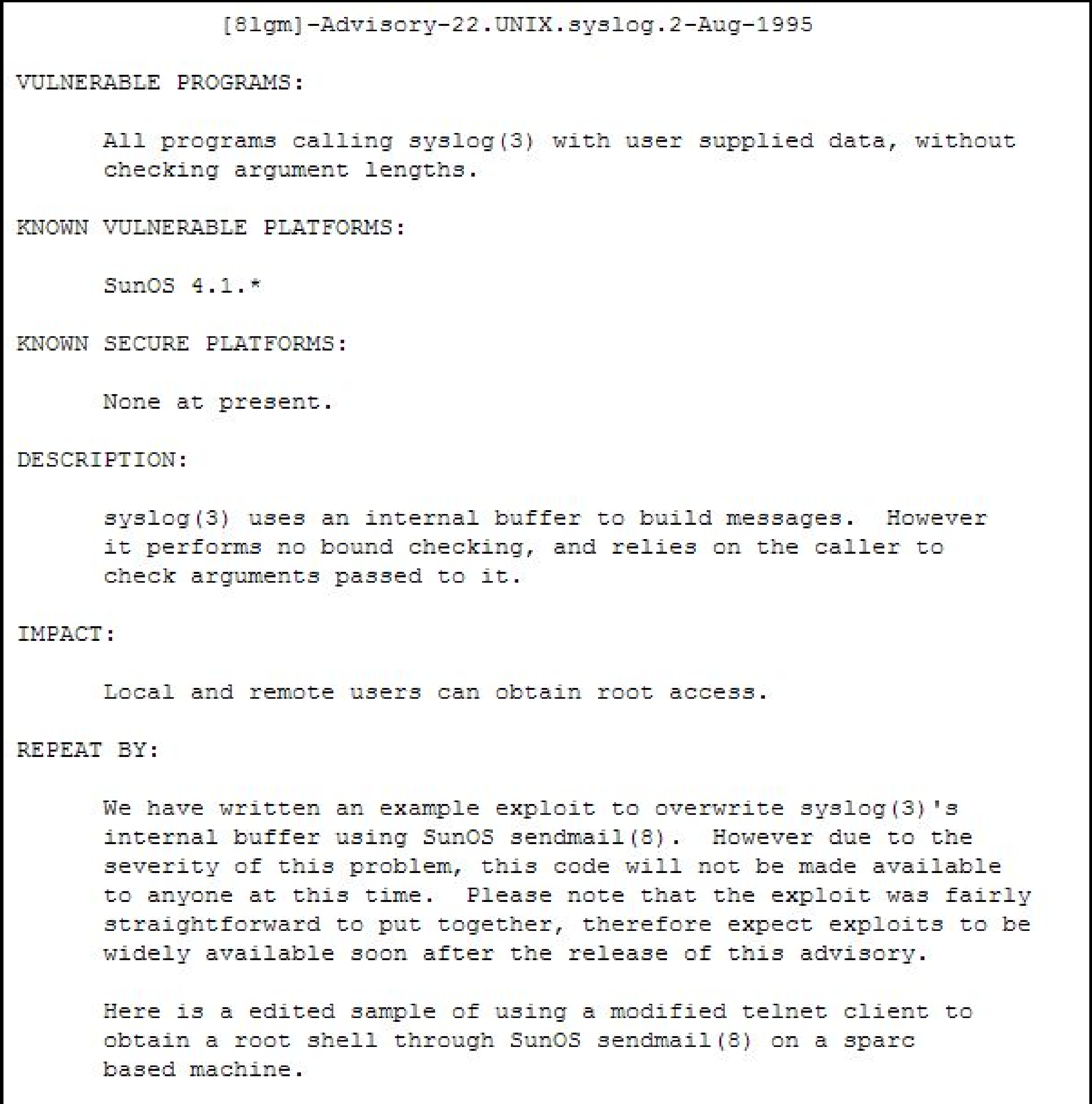
- 8lgm (8 legged groove machine)

Basic format, remains today:

- Affected software & OS's
- Description of Impact
- Fix and workaround info
- Reported to vendor and to the public

Extremely controversial at time!

- But in a sense necessary



[8lgm]-Advisory-22.UNIX.syslog.2-Aug-1995

VULNERABLE PROGRAMS:

All programs calling syslog(3) with user supplied data, without checking argument lengths.

KNOWN VULNERABLE PLATFORMS:

SunOS 4.1.*

KNOWN SECURE PLATFORMS:

None at present.

DESCRIPTION:

syslog(3) uses an internal buffer to build messages. However it performs no bound checking, and relies on the caller to check arguments passed to it.

IMPACT:

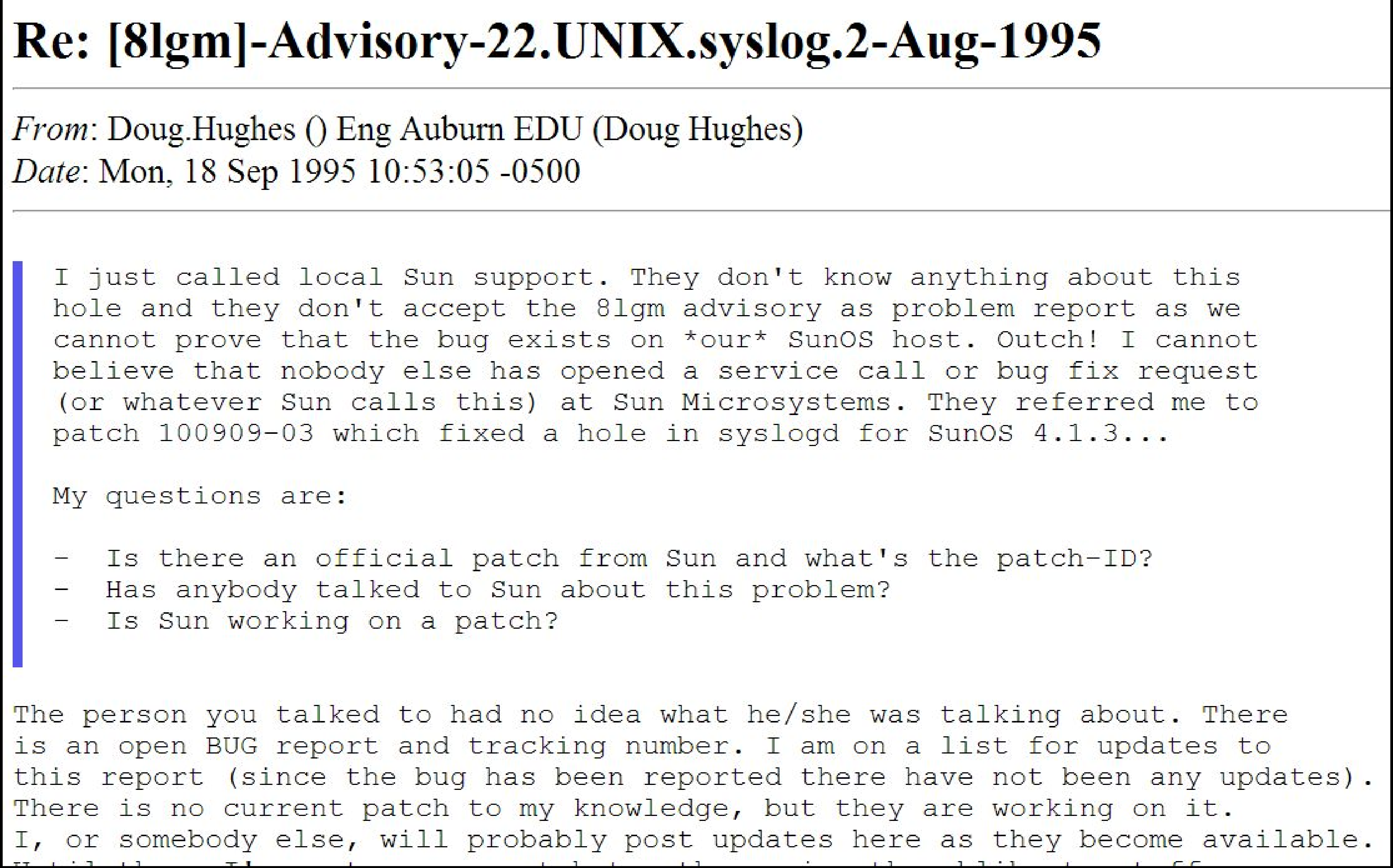
Local and remote users can obtain root access.

REPEAT BY:

We have written an example exploit to overwrite syslog(3)'s internal buffer using SunOS sendmail(8). However due to the severity of this problem, this code will not be made available to anyone at this time. Please note that the exploit was fairly straightforward to put together, therefore expect exploits to be widely available soon after the release of this advisory.

Here is a edited sample of using a modified telnet client to obtain a root shell through SunOS sendmail(8) on a sparc based machine.

# Full Disclosure common outcome...



## Re: [8lgm]-Advisory-22.UNIX.syslog.2-Aug-1995

*From*: Doug.Hughes () Eng Auburn EDU (Doug Hughes)
*Date*: Mon, 18 Sep 1995 10:53:05 -0500

> I just called local Sun support. They don't know anything about this hole and they don't accept the 8lgm advisory as problem report as we cannot prove that the bug exists on *our* SunOS host. Outch! I cannot believe that nobody else has opened a service call or bug fix request (or whatever Sun calls this) at Sun Microsystems. They referred me to patch 100909-03 which fixed a hole in syslogd for SunOS 4.1.3...
>
> My questions are:
>
> - Is there an official patch from Sun and what's the patch-ID?
> - Has anybody talked to Sun about this problem?
> - Is Sun working on a patch?

The person you talked to had no idea what he/she was talking about. There is an open BUG report and tracking number. I am on a list for updates to this report (since the bug has been reported there have not been any updates). There is no current patch to my knowledge, but they are working on it.
I, or somebody else, will probably post updates here as they become available.

# Situational awareness was bad....

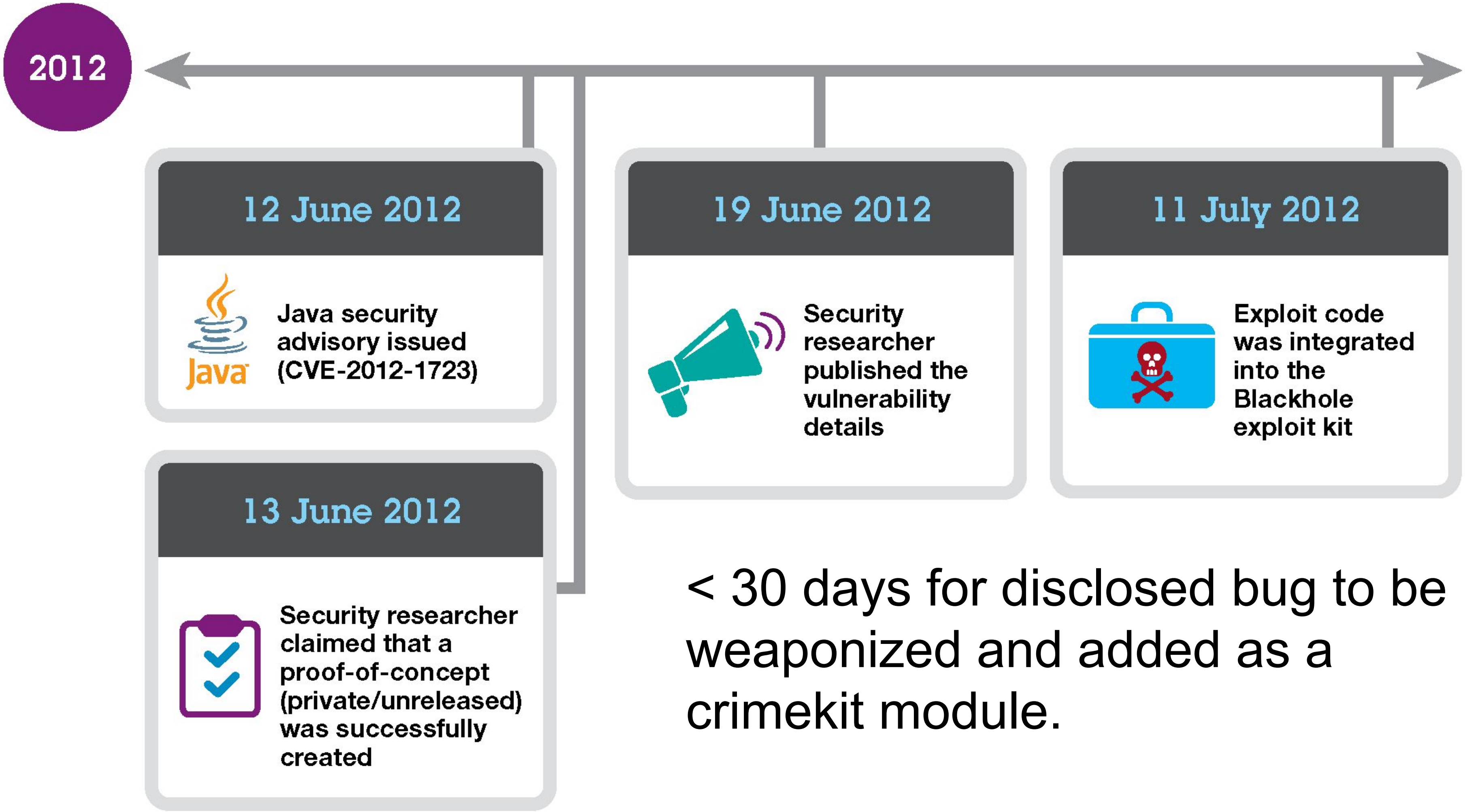
Vendors had poor communication:

- led to confusion/panic in customers
- lawyers involved
- slow patching / solutions
  - Attackers could exploit reported bug faster than it could be patched
    - and that still happens today

Still are problems @ startups/small companies

## Timeline of one-day attacks for 2012 Java vulnerability

12 June 2012 through 11 July 2012



2012

**12 June 2012**

Java security advisory issued (CVE-2012-1723)

**13 June 2012**

Security researcher claimed that a proof-of-concept (private/unreleased) was successfully created

**19 June 2012**

Security researcher published the vulnerability details

**11 July 2012**

Exploit code was integrated into the Blackhole exploit kit

< 30 days for disclosed bug to be weaponized and added as a crimekit module.

*Figure 5. Timeline of one-day attacks for 2012 Java vulnerability (CVE-2012-1723), 12 June 2012 through 11 July 2012*

Source: IBM X-Force® Research and Development

# Full Disclosure continues

The main problems:

1. Creates a problem to **force** vendors to act
2. Lack of clarity around vuln research and legal issues
   - Vendor's first reaction was to get lawyers involved
3. Underground industry evolved around all the new available info
   - mass malware rises from full disclosures
   - script kiddies got more skills

Bottom lines:

1. "Researchers" became famous from it
2. FD did not result in a reduction of attacks...

# Responsible Disclosure ~2002

Mass Malware & Worms made people reconsider FD in 2000's.

- ILOVEYOU, Code Red, Code Red II, Nimda, Blaster, Slammer, etc...
- Most worms reused FD researchers' code

"Responsible Vulnerability Disclosure Process"

- Submitted to IETF by Christey & Wysopal in 2002
- Responsible - researchers withhold info until vendor patch
- Responsibilities centered around researchers, not vendors  (problem???)
- Source:http://tools.ietf.org/html/draft-christey-wysopal-vuln-disclosure-00

# Bug Bounties ~2010

People came to realize:

- Vulnerability research is a valuable service that protects vendors and customers, and it should be rewarded.
- Linus's Law: "*given enough eyeballs, all bugs are shallow*" (Linus Torvalds)
- Thus bug bounties were formed
    - Bugs for $$$$$!

# Bug Bounties 2013

| Company | Scope | Bounty | URL |
|---|---|---|---|
| Google | Web & Apps | $500-$20,000 | http://www.google.com/about/appsecurity/reward-program/ |
| Facebook | Web | $500 + | https://www.facebook.com/whitehat/bounty/ |
| Mozilla | Web / Mobile/ Apps | $500 - $3,000 | http://www.mozilla.org/security/bug-bounty.html |
| Barracuda | Appliances | up to $3,133.70 | http://www.barracudalabs.com/bugbounty/ |
| Zero Day Initiative | Popular software / applications | Reward points, benefits, and $500-$5,000 | http://www.zerodayinitiative.com/about/ |

# Bug Bounties 2013

| Company | Scope | Bounty | URL |
|---|---|---|---|
| tarsnap | Web & Apps | $1-$2,000 | http://www.tarsnap.com/bugbounty.html |
| Wordpress | Web | $100-$1,000 | http://www.whitefirdesign.com/about/wordpress-security-bug-bounty-program.html |
| Hexrays | Software | $5,000 | http://www.hex-rays.com/bugbounty.shtml |
| Paypall | Web / Apps | unknown | https://cms.paypal.com/cgi-bin/marketingweb?cmd=_render-content&content_ID=security/reporting_security_issues |

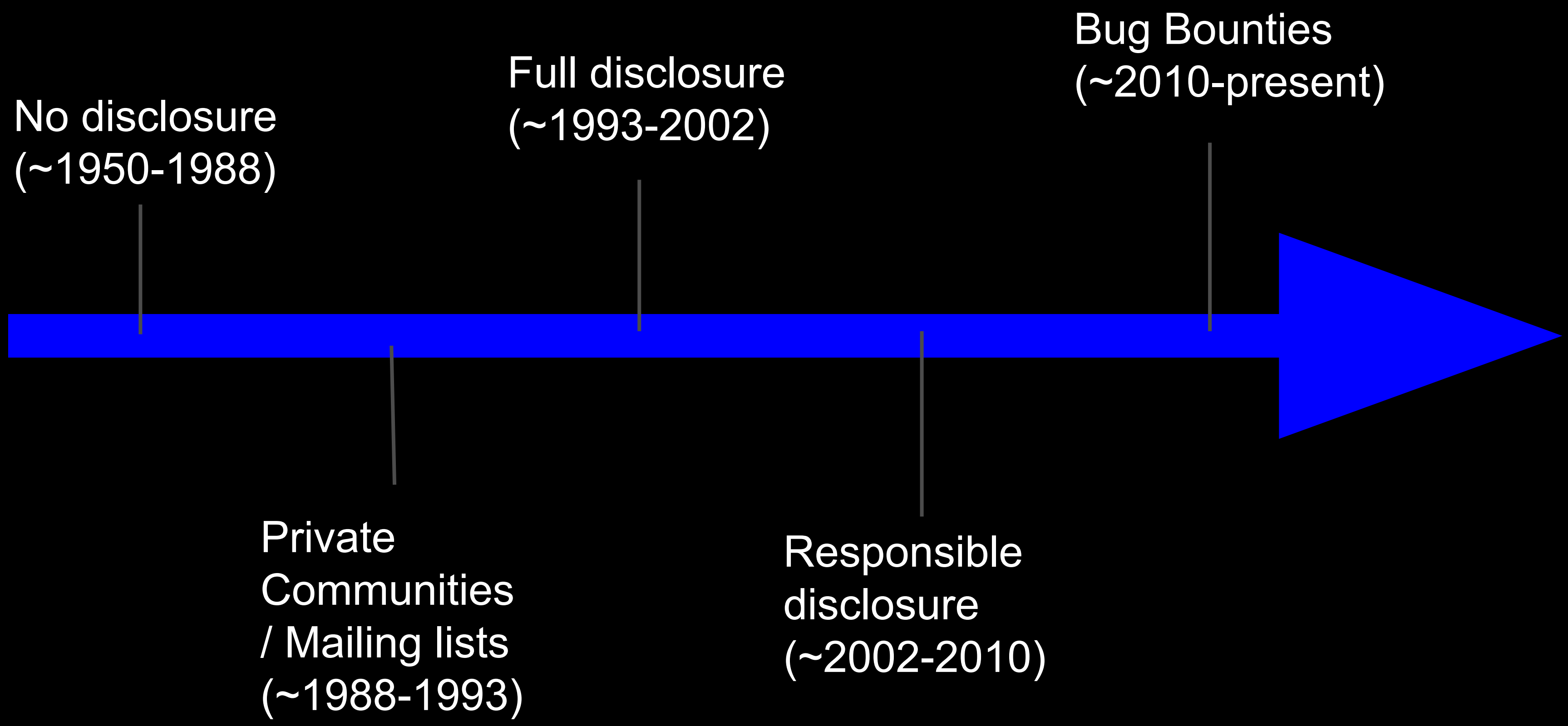
And many more.....

# Bug Bounties and Disclosure Websites

Huge list here:

http://computersecuritywithethicalhacking.blogspot.com/2012/09/web-product-vulnerabilty-bug-bounty.html

# Timeline



No disclosure (~1950-1988)

Private Communities / Mailing lists (~1988-1993)

Full disclosure (~1993-2002)

Responsible disclosure (~2002-2010)

Bug Bounties (~2010-present)

# Vendor's Patching Trends got better

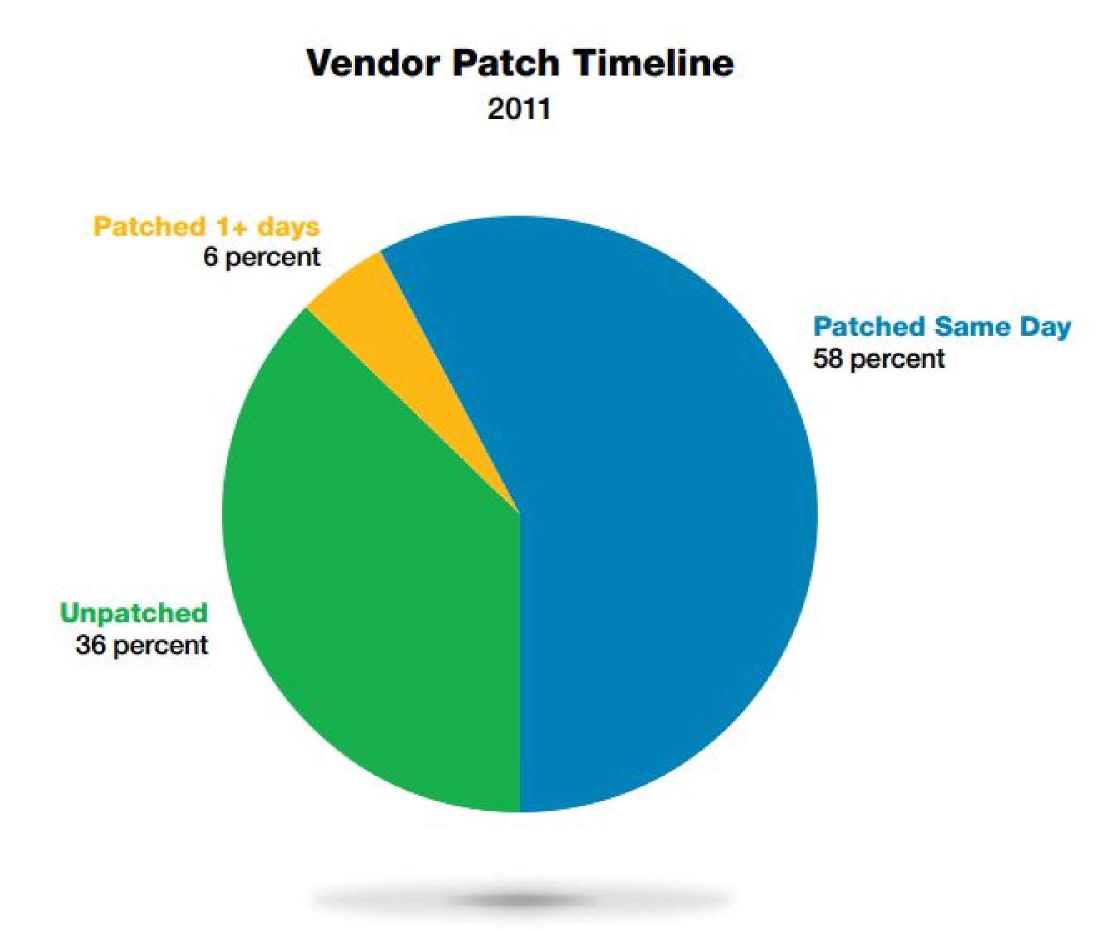


*Figure 46: Vendor Patch Timeline – 2011*

Source: IBM's X-Force 2011 Trend and Risk report

# Vendor's Patching Trends got better

## Unpatched vulnerabilities

The total amount of unpatched vulnerabilities recorded **dropped by 15%** in 2013.

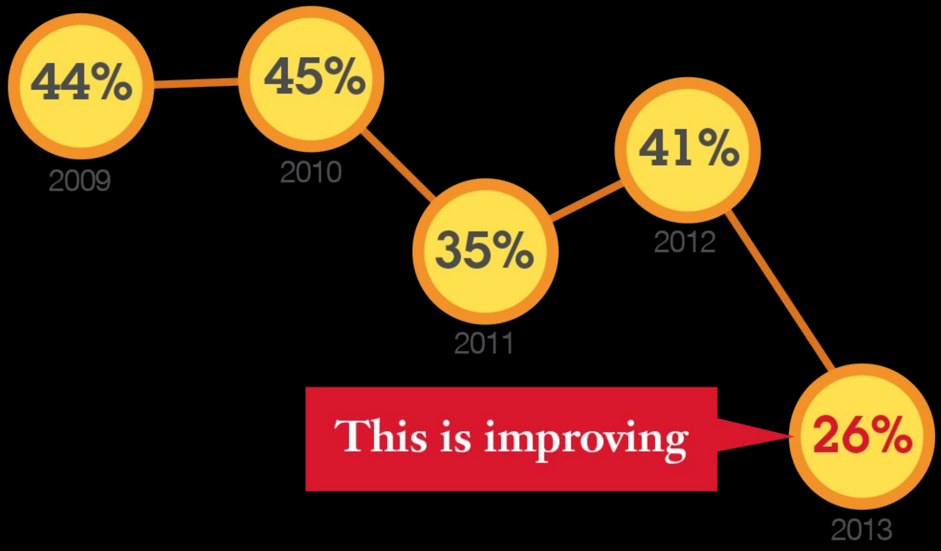


*Figure 10. Vendor patch rates of publicly disclosed vulnerabilities, 2009 to 2013*

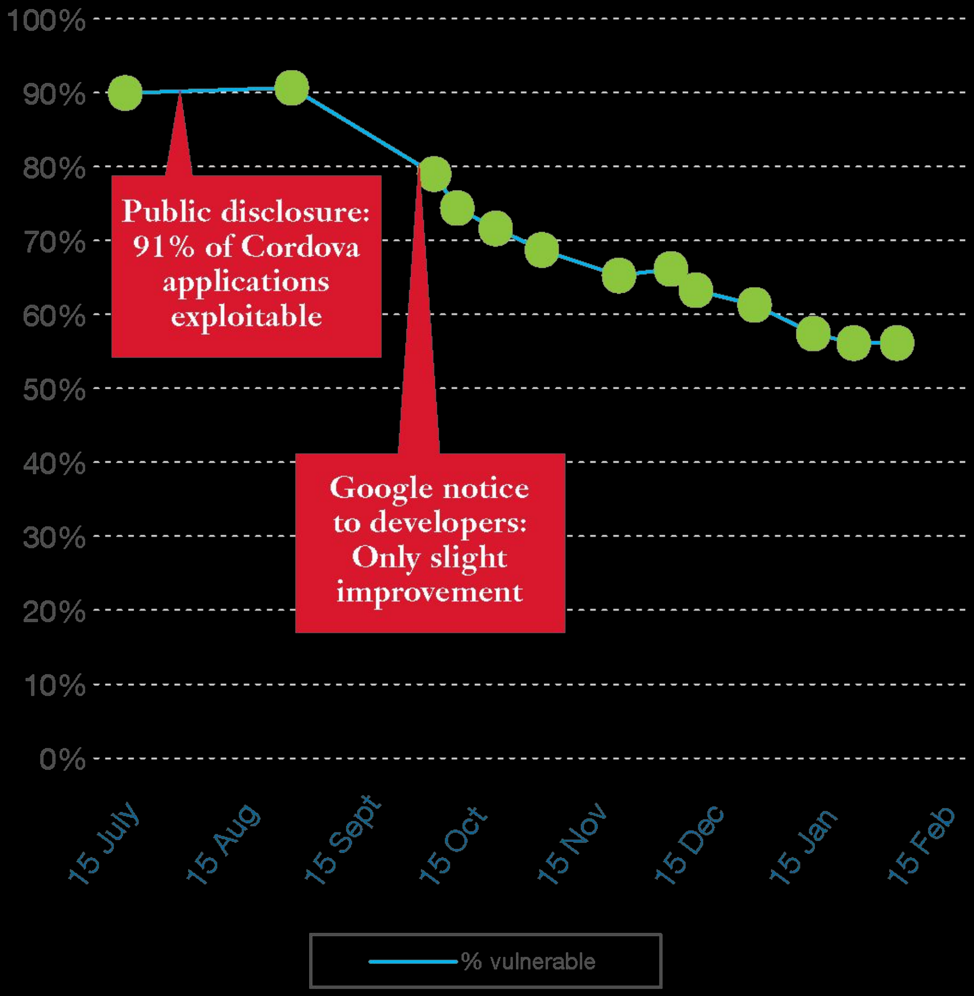
Source: IBM X-Force® Research and Development

# But not good in other areas

- Mobile
  - Stagefright
  - [http://androidvulnerabilities.org/](http://androidvulnerabilities.org/)
- SCADA
- Embedded

Developer response to Cordova vulnerability disclosures

15 July 2014 through 02 February 2015



Public disclosure: 91% of Cordova applications exploitable

Google notice to developers: Only slight improvement

*Figure 4. Developer response to vulnerabilities after disclosures, 15 July 2014 through 02 February 2015*

Source: IBM X-Force® Research and Development

# The Value of Offensive Security

More bugs are found

- ethically disclosed
- better patching

Less full-disclosures of weaponized bugs:

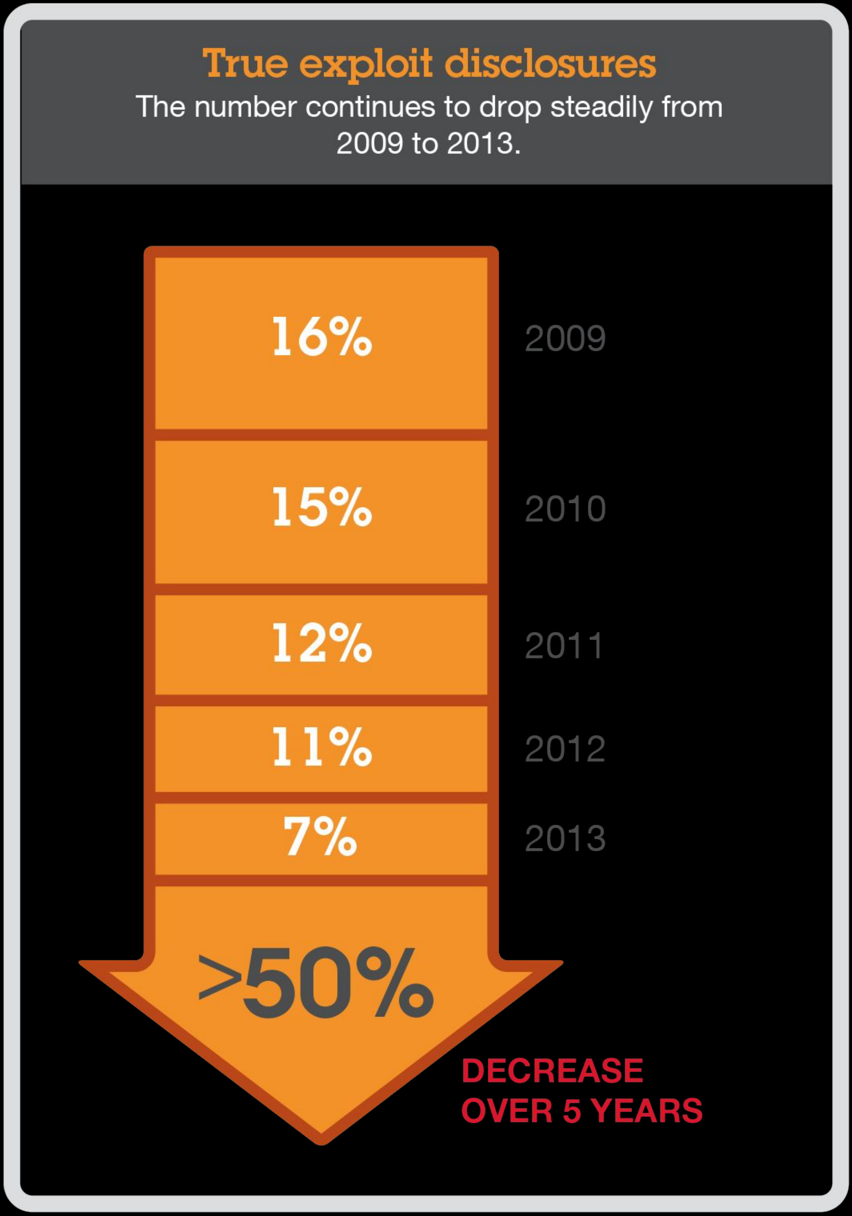


*Figure 13. True exploit disclosures, 2009 to 2013*

Source: IBM X-Force® Research and Development

## Are things getting worse?

Situational Awareness
is getting better

# Disclosure Debate

Still people are all about:

- Anti-disclosure
- Full-disclosure
- Responsible-disclosure
- Coordinated-disclosure
- Delayed-disclosure
- etc...

# How NOT to do disclosure:

RAGE-BLOG by ORACLE’s chief security officer complaining about all security bug submissions.

https://web.archive.org/web/20150811090106/https://blogs.oracle.com/maryanndavidson/entry/no_you_really_can_t

-STOP BREAKING THE LICENSE!

-ZERO credit for researchers

GTFO!!!!!1! STOP LOOKING AT OUR CODE FOR BUGS

# How NOT to do disclosure:

Video from the hacker who was behind the July 2013 Intrusion on Apple Developer's sites.

http://www.youtube.com/watch?v=q000_EOWy80

- Shows ACTUAL user's personally identifiable information (PII) in his video
  - "I am being accused of hacking but I have not given any harm to the system and i did notwanted to damage."
  - Likely a troll

# Finally: About Vendor Negligence

Vendor know about vuln but refuses to patch. = no consequence

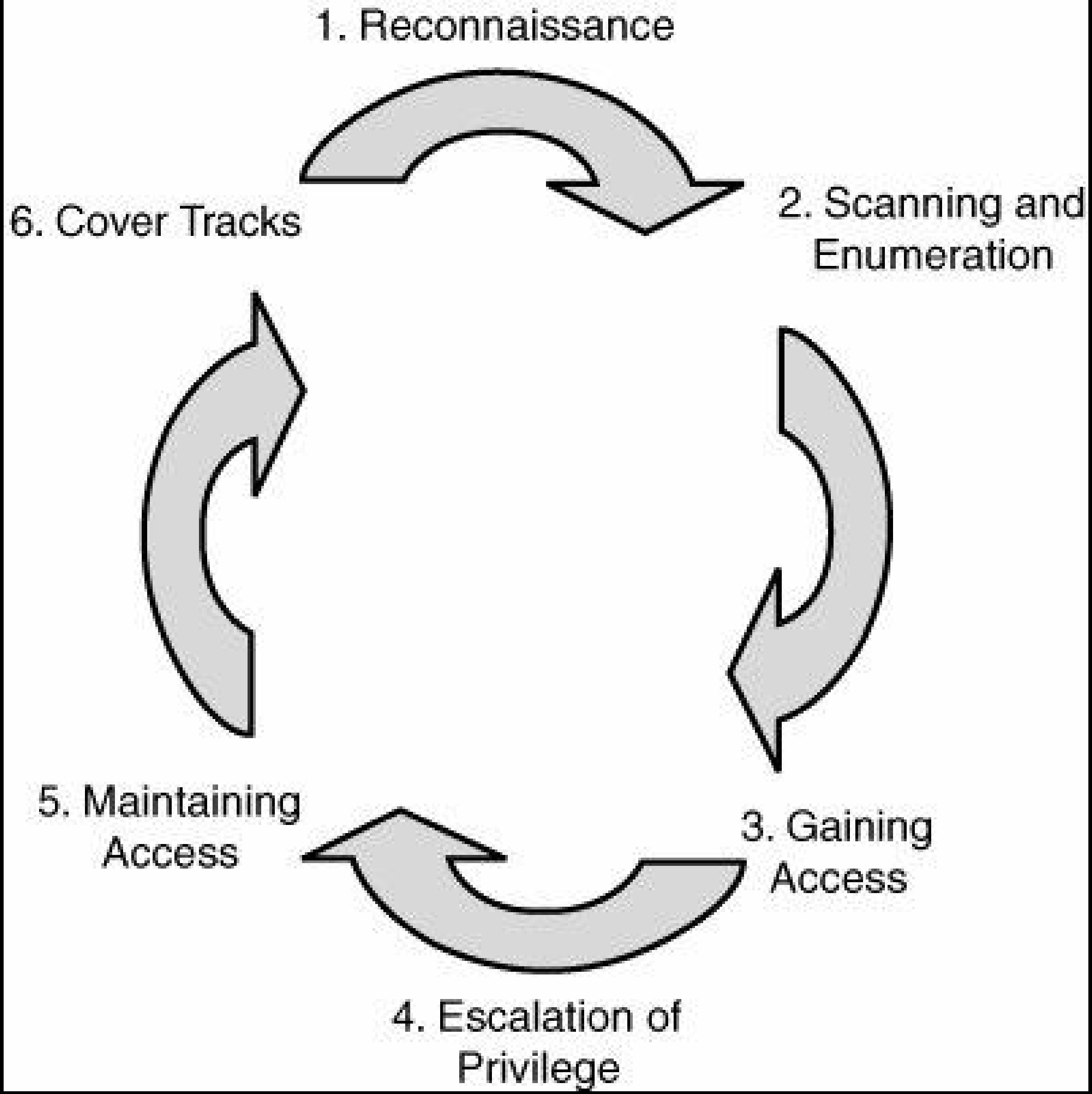
Person A knows Person B is dying but refuses to help (but can help) =  potentially Negligent Homicide.

No equivalent in Cyber.

- Negligent Robbery?
- Negligent Identity Theft?
- Negligent Abuse of Computer Systems (CFAA :P)

# On second thought, lets get back to..

# The Basics of Penetration Testing and Hacking



# Prior to a penetration test... getting permission

A discussion with the client establishes the following:

1. The type of penetration test
    1. physical access or just remote access?
    2. social engineering allowed?
    3. covert or overt
2. Rules of Engagement
    1. What is off limits
    2. Threat model (insider threat, ex-employee, outsider, etc)
    3. Specified targets
3. Timeline
4. What to expect from the report

# 1) Reconnaissance

- Internet searches
  - For URLs (google, yahoo, bing, etc)
  - For devices / access points (http://www.shodanhq.com/)
  - Company website
    - cached versions
  - of public records
  - social media
- Phone calls
  - to sales
  - to IT
  - to PR
- Visit in person...

# This = Intelligence Gathering

Identifying target and it's assets, and services, and gathering as much info as possible.

- Company Website, google
- Public Financial records / news
  - Recent / future mergers
- DNS records
- Social Media, employee blogs
- phone calls, visits

OSINT
(open source intelligence)

HUMINT, usually off limits

http://www.pentest-standard.org/index.php/Intelligence_Gathering

# 2) Scanning and Enumeration

This involves determining what applications/OSes are up and running, what versions they are, discovering accounts for them, and how to access the applications.

TONS of tools for automating this.

- nmap
- w3af
- sqlmap
- metasploit
- many many more

# Identifying Attack Surface

Depends on the entity (system, business, etc), and the components

For a single system: would be all ports running open, all user accounts and the strengths of their passwords, the filesystem permission model, all available programs (i.e. /bin/cp, /bin/ls, /bin/sh, /bin/bash), and *anything excluding physical access.*

# Discovering Vulnerabilities

- Perhaps a vulnerable CMS is used, or plugin?
  - plugins are attacked far more than the framework
- Perhaps an old network service is in use?
- Default credentials work anywhere?
  - routers, SCADA, PLC

etc...

# 3) Gaining access

Via:

- Brute force
- web hacking
- exploit development
- malware / mass-malware
- Social Engineering
- etc...

# Common ways attackers break into businesses

- Social Engineering (HUMINT)
  - easiest way in BY FAR
  - spear phishing: trick an employee to visit your malicious link, or execute your malicious attachment, or give over user/pass
- Web application exploitation
  - command injection: SQLi, CGI,
  - directory traversal: ....home.php?../../../../etc/passwd
- Pivoting from 3rd party partner systems
- Network application exploitation
- Malicious USB's, or gift gaming keyboards.
- and more

# 4) Privilege Escalation

Gaining access is just one step.

Attackers want root.

- Password cracking
- SUID program exploits
- sandbox escape
- keylogging
- More social engineering
- etc...

# 5) Maintaining Access & Post Exploitation

After attackers get *root* access to your systems:

- establish back doors (prefer open source applications, for ease)
- crack moar passwords, expand control
- erase logs
- go after your IP, data, and users
- steal $$$
- pivot into 3rd party systems

# What you will learn in this class

- Reverse engineering (x86) of binaries
- Exploit Development
    - Shellcode development
- Network hacking
- Web Application Hacking
    - SQLi, XSS
- Social Engineering
- Metasploit
- Post Exploitation techniques
- Lockpicking (Physical security is important too!!!) and more

# The most important thing you will learn

How to communicate system vulnerabilities to others.  So that they can fix them!

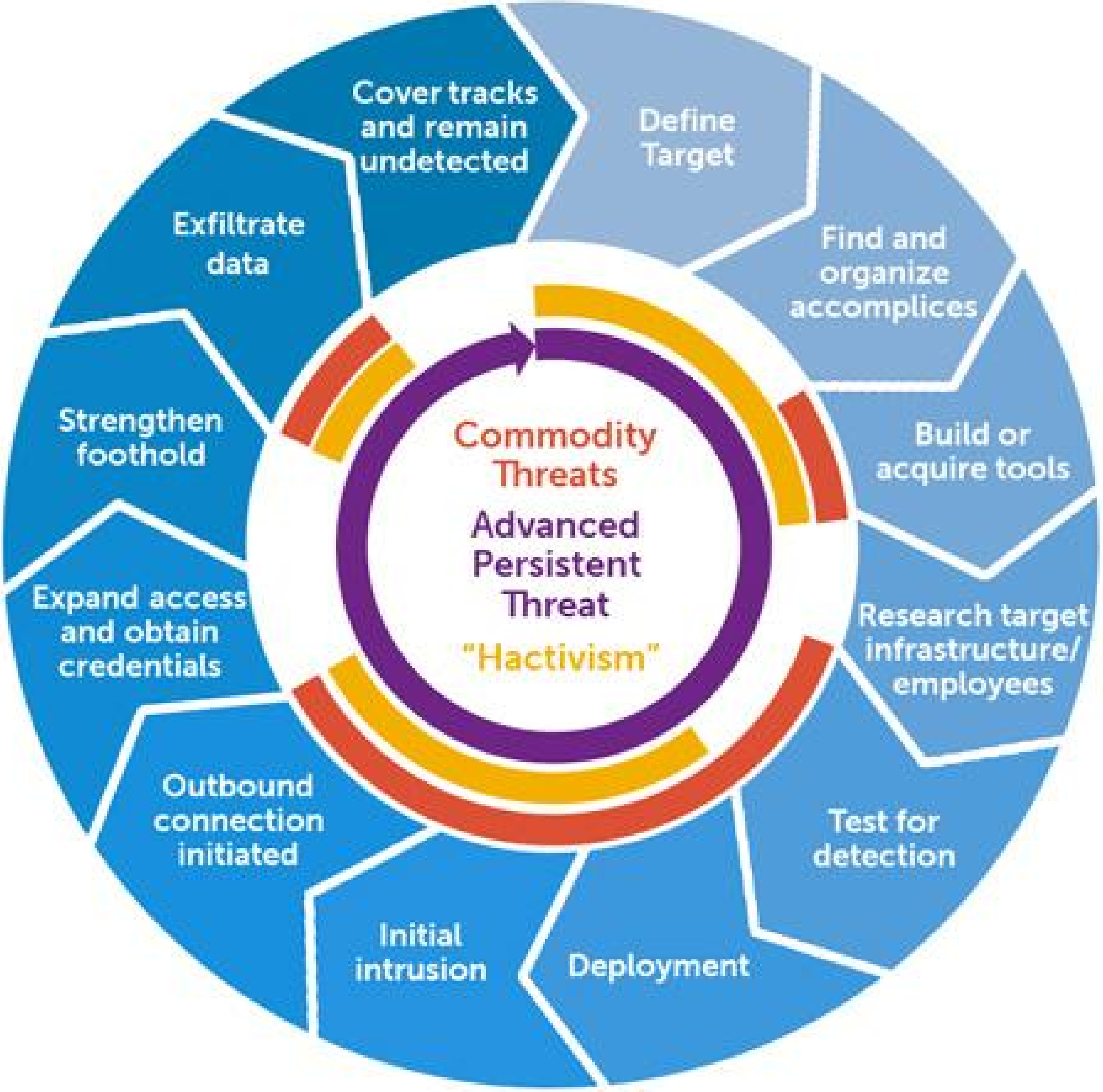
Hackers who cannot communicate are....

WORTHLESS

# Categorizing Threat

The key is understanding the capabilities posed by threats.

The number of threats is continually increasing.


Define Target
Find and organize accomplices
Build or acquire tools
Research target infrastructure/ employees
Test for detection
Deployment
Initial intrusion
Outbound connection initiated
Expand access and obtain credentials
Strengthen foothold
Exfiltrate data
Cover tracks and remain undetected
Commodity Threats
Advanced Persistent Threat
"Hactivism"

from wikipedia

# But why

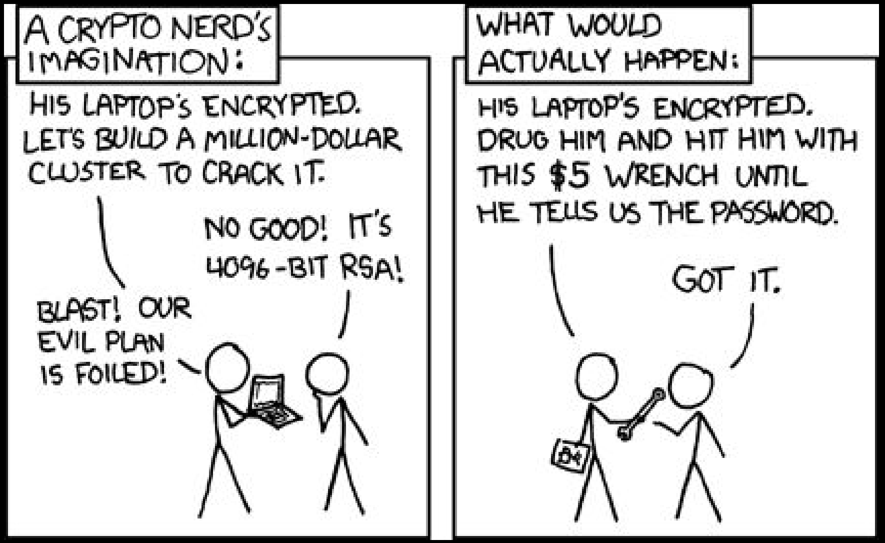
RISK = THREAT x VULNERABILITY

it is important to express the threat model when discussing vulnerabilities to help assess risks

# Real World

Bad guys have major advantage.  They can:

- use proxies, spoof IP, MAC address
  - attack anonymously
- utilize android/windows spyware apps
- attacking your partners
- blackmail/$5 wrench
- easily buy crimekits
  - zeus tr0jan
- can break many laws
  - impersonate police
    - social engineering


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

# Real World...

That’s why pen testing and incident responders are so important

# The COST of Reactive Security

## Consequences of exploitation 2013

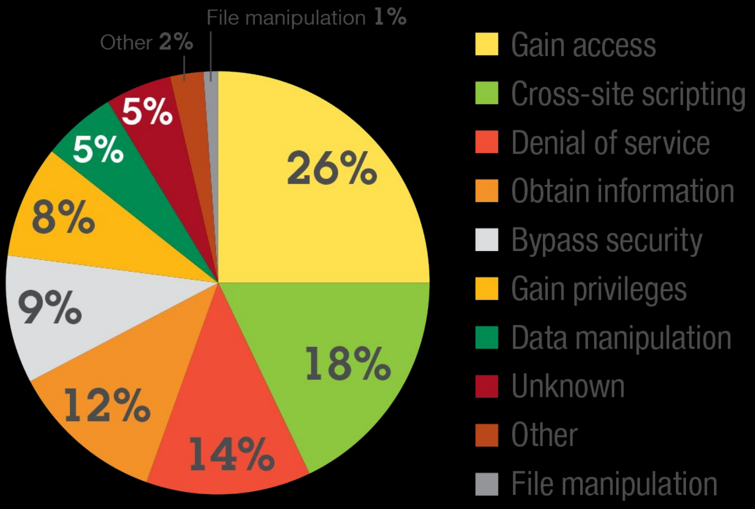
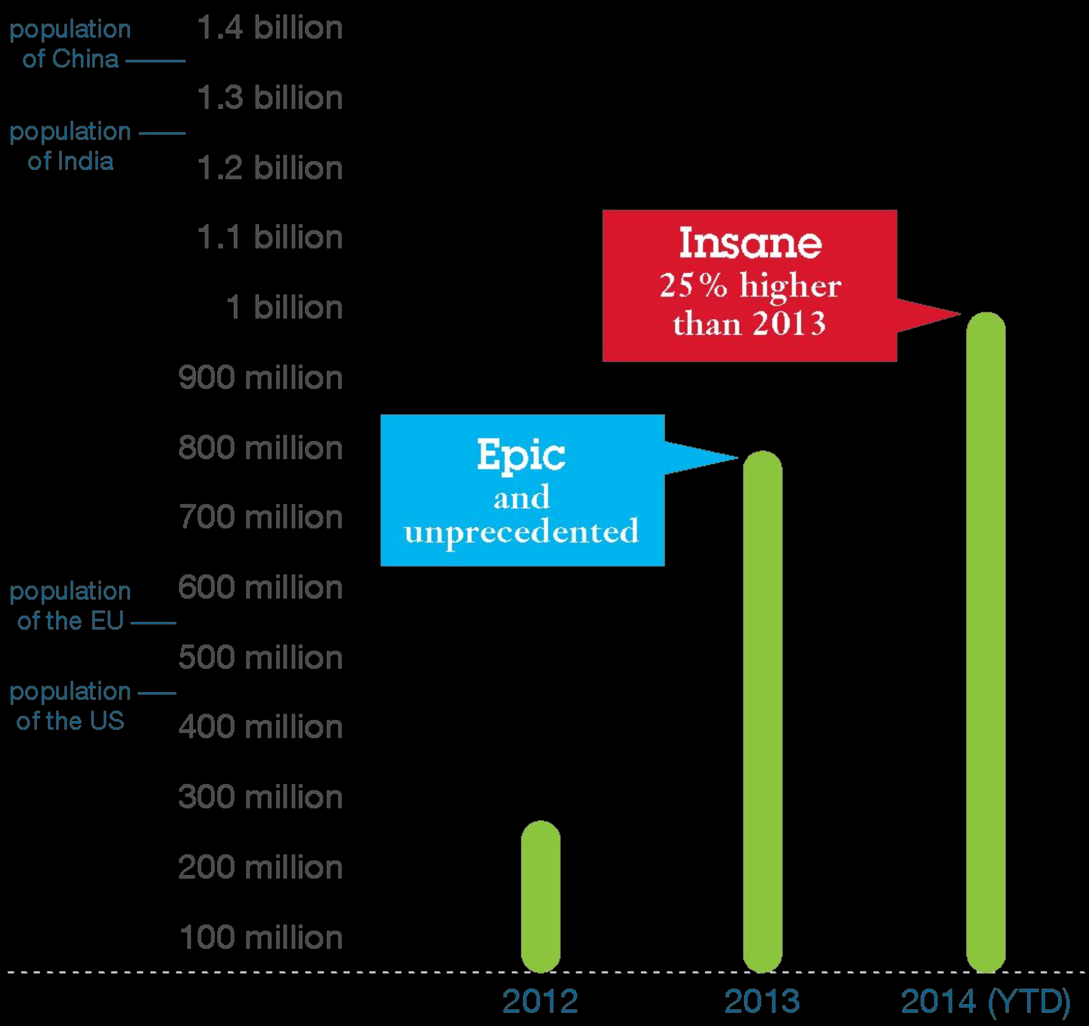


*Figure 12. Consequences of exploitation 2013*

Source: IBM X-Force® Research and Development

## Total records leaked by year

compared to estimated population sizes



**Epic** and unprecedented

**Insane** 25% higher than 2013

*Figure 1. Total records leaked by year, compared to estimated population sizes*

Source: IBM X-Force® Research and Development

# Doubts?

Can't we just fix this crap by:

- Using [better] security tools?
- everyone being smart (no more dumb users)
- everyone using strong passwords
- safe code
    - (no unsafe C functions)
    - safer languages like python
    - fix all the buffer overflows, SQLi vulns, etc!!

**its <CURRENT YEAR>!!!**

- keeping everything patched?
- etc...

**NO**

**:*(**

# Questions?

Reading: 0x200 up to 0x260 (HAOE)

# Sources

All the history slides:

- Dan Guido "Vulnerability Disclosure: Penetration Testing and Vulnerability Analysis", Fall 2011. pentest.cryptocity.net/files/intro/vuln_disclosure.pdf

All the IBM X-force Research Graphics:

- http://www-03.ibm.com/security/xforce/downloads.html