

Introduction

CIS 4930 / CIS 5930

Offensive Security

Prof Xiuwen Liu

W. Owen Redwood

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
 - Final project demands you do something impressive:
 - Make a difference on the security community
 - Expand existing tools
 - Design new tools
 - Explore cutting edge tools / techniques / skills

This class

Week 1: Intro

Week 2: Secure C & Code auditing

Week 3: Intro to Vulnerability Research

Week 4: Reverse Engineering Workshop

Week 5: Fuzzing and Automated Testing

Week 6: Exam #1 / Exploit Development

Week 7: Exploit Development week 2

Week 8: Network Hacking

Week 9: Web Application Hacking Week 1

Week 10: Spring Break

Week 11: Web Application Hacking Week 2

Week 12: Advanced Exploitation 1


Week 12: Advanced Exploitation 2 & Exam #2

Week 13: Penetration Testing & Incident Response Topics Week 1

Week 14: Penetration Testing & Incident Response Topics Week 2

Week 15: Physical Security and Social Engineering

Week 16: Final Exam (takehome)



*Taught with a
defense-centric
focus*

This class

Offered previously as “Offensive Security” in spring 2013

- CS, Comp Crim, EE, Biomath majors (grad / undergrad)
- Made headlines in the security field
 - During DEFCON :)
- Now with more guest lecturers / experts!

The Instructors

- Professor Xiuwen Liu (liux@cs.fsu.edu)
 - specialties: Computer Vision, Pattern Analysis, Computer Security, Cyber Physical Systems Security, etc...
- W. Owen Redwood (redwood@cs.fsu.edu)
 - The primary instructor
 - specialties: counter intelligence, system administration, exploit development, web application hacking, insider threats, and other bad stuff
 - don't call me "professor"

#whoami (Owen)

Exploit Developer, Reverse Engineer, Guitarist

PhD Student at FSU, under Mike Burmester

Research area: *Counterintelligence tools for critical infrastructure and insider threats*

Director of SAIT Research Lab 2010

Founded NOL3ptr CTF team 2011

Now FSU Cybersecurity Club

Sandia National Labs Internship in 2012:

Briefed Obama's Chief Science Advisor

Invited member of Sandia Summer Institute Think-tank 2012

Created from scratch and taught Offensive Computer Security at the graduate level:

<http://offsec.noleptr.com>



DISCLAIMER:

MY OPINIONS ARE MY OWN AND NOT ANY OF MY EMPLOYERS'

The Website

Hosted at: <http://www.cs.fsu.edu/~redwood/OffensiveComputerSecurity/>

We will try to video record (screencast) all the lectures and host the links on the website.

- Means you can save trees by not printing out all the lecture slides

Grade Breakdown

Homeworks 55%

10 Homeworks 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%

Required by FSU. (takehome)

No more term project

Grading Policy

Individual work only:

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

In all homeworks I grade based off of your:

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

SAIT Lab Access (room 010)

- Most homeworks will not require the lab, and can be done at home in a virtual machine, or by ssh into the lab.
- If you have a project idea, and would like to use the lab, contact us for access
 - We're happy to help!

Midterm 1 and 2

Midterm 1 = Feb 10

Midterm 2 = April 2

Extra Credit

Extra credit will be granted for:

- Participation in any capture the flag games
 - See the FSU CyberSecurity Club to get involved
 - Weighed upon difficulty of problems solved, and your level of participation
 - see <https://ctftime.org/>

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"

This thing we call "Security"

AND THEY CALLED

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

ME CRAZY



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
- *My opinion:*
 - Should be more involved in testing / fuzzing and evaluated on bugs found as well as security job.

Its time to wake up

<http://www.digitalattackmap.com/>

But this is just DDoS

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

Who this class is for

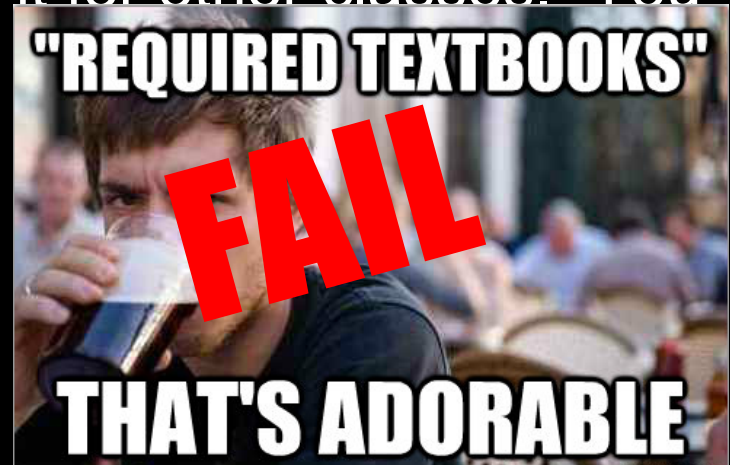
Seniors and Grads who want to become:

- Incident Responders
- Penetration Testers
- Security Professionals
- Forensics Professionals
 - i.e. FBI, law enforcement
- Vulnerability Researchers
- and so on

We will focus mainly on penetration testing and incident response

Who this class is **NOT** for

- Students who have not taken a security class before
 - 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

- 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

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 many of the homeworks

The books used to create this class

An incomplete list:

- Hacking: The Art of Exploitation
- Counter Hack Reloaded
- The Web Application Hacker's Handbook
- The Shellcoder's Handbook (2nd ed)
- Windows Internals 6 (1 & 2)
- 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

- Teaching only defense is like teaching people only to play goalie in soccer when you don't even know what the goal looks like.
 - people will be taking shots at you all day, and if you don't know how to attack, you won't know what to expect.
- *"One test is worth a thousand expert opinions"* - Anonymous dude
- Penetration testing is the best way to assess correct implementation of security controls and policies
 - And required for regulations Compliance (i.e. PCI...)

Motivations

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")

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

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 Security numbers hacked in S.C. SECURITY

Tax returns, personal data compromised in breach

By NOELLE PHILLIPS - nophillips@thestate.com

The U.S. Secret Service detected a security breach Oct. 10, but it took state officials 10 days to close the breach and 30 days to inform the public that 3.6 million Social Security

The attack also exposed 387,000 credit and debit card numbers and other information people file with their tax returns. Taxpayer identification numbers also potentially have been compromised. The breach is being described as one of the nation's largest agency

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

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.

Pen Testing & Incident Response

Both require a great deal of offensive knowledge

"Dark Arts"



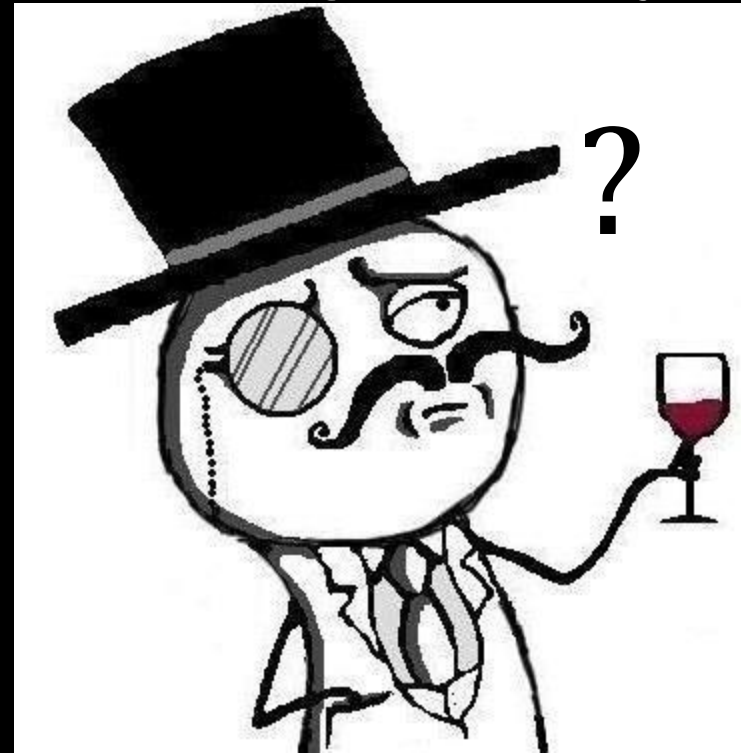
But Pen Testing = proactive (hopefully)
and Incident Response = reactive

Hacking versus Penetration Testing

Hacking, *AKA cracking, etc..*

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

What's the difference?



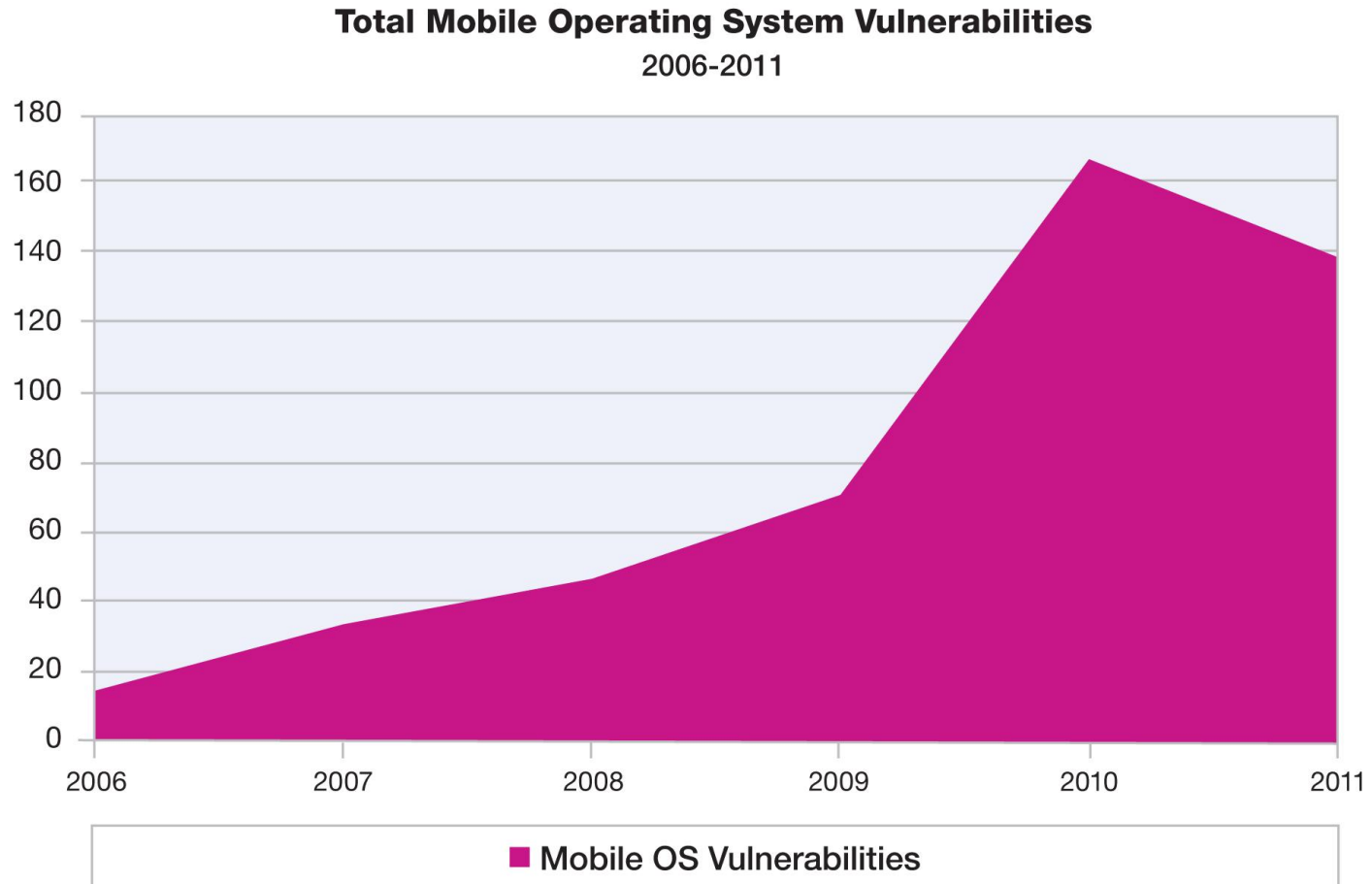
PERMISSION

really thats it.

Without permission, its
ILLEGAL

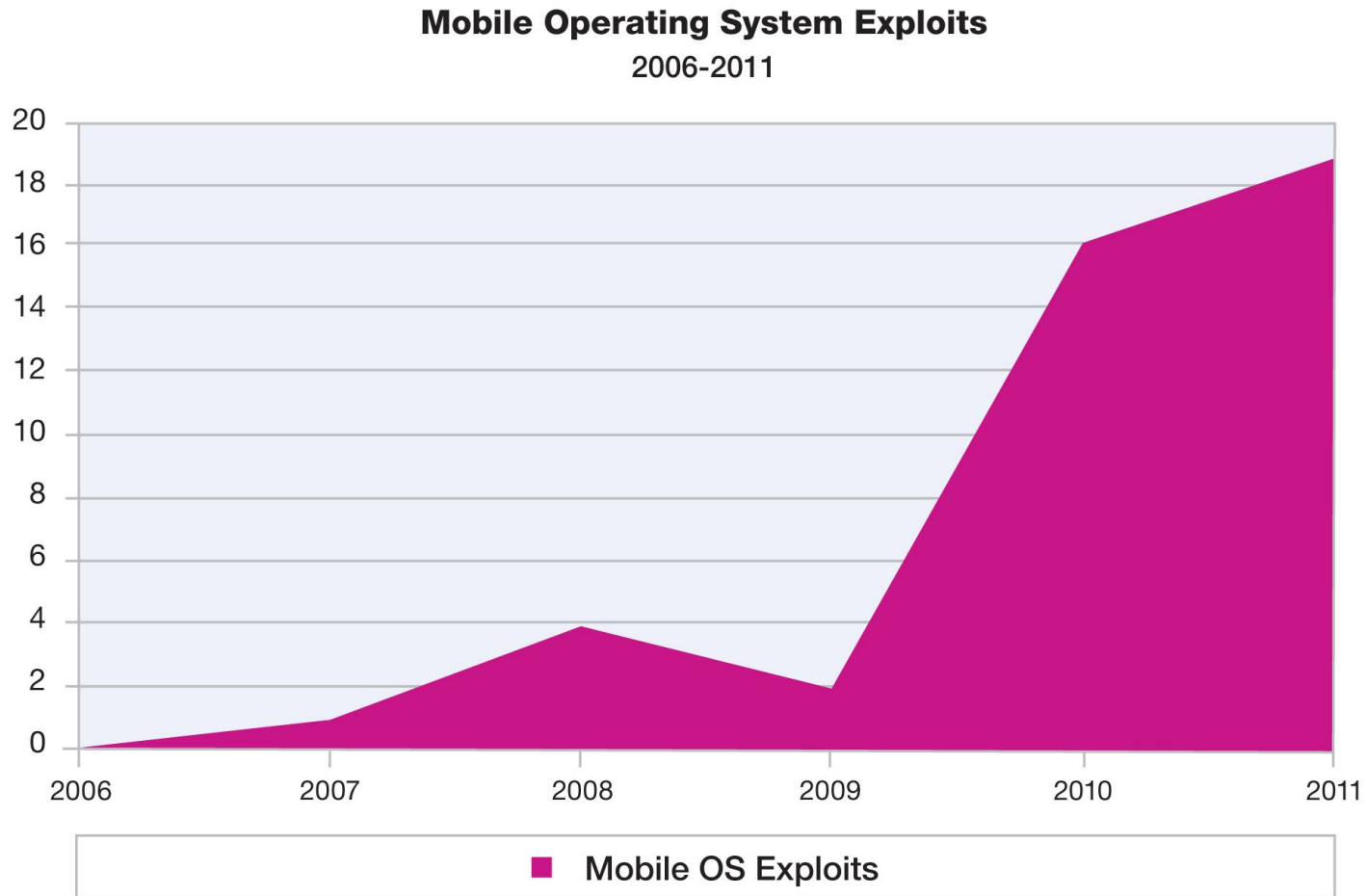
Lets talk Vulnerabilities

Vulnerabilities (Mobile)



Source: IBM X-Force® Research and Development

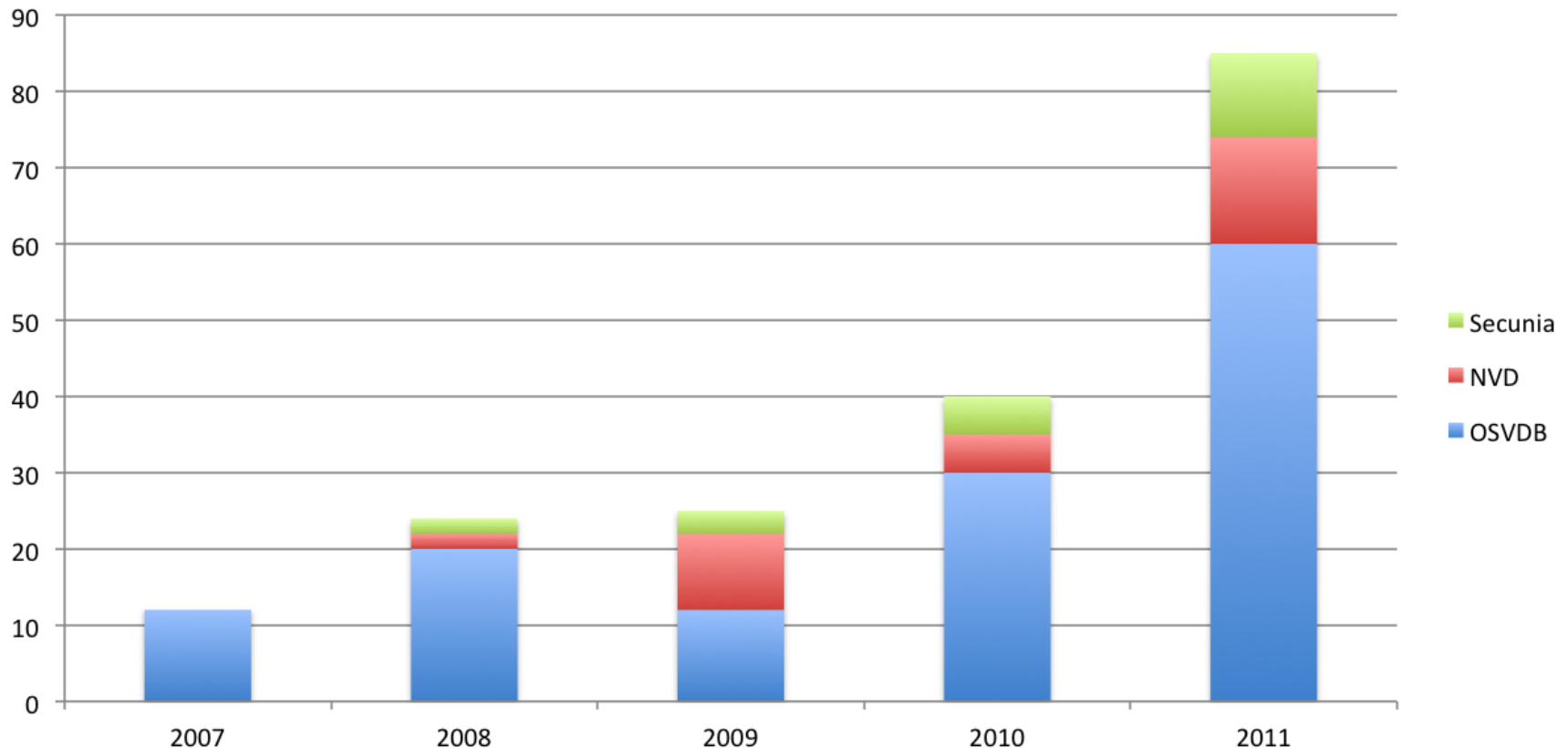
Exploits (Mobile)



Source: IBM X-Force® Research and Development

Vulnerabilities (SCADA)

Search for "SCADA", By Year Of Advisory Issuance, in Popular Vulnerability Databases as of 9/12/2011

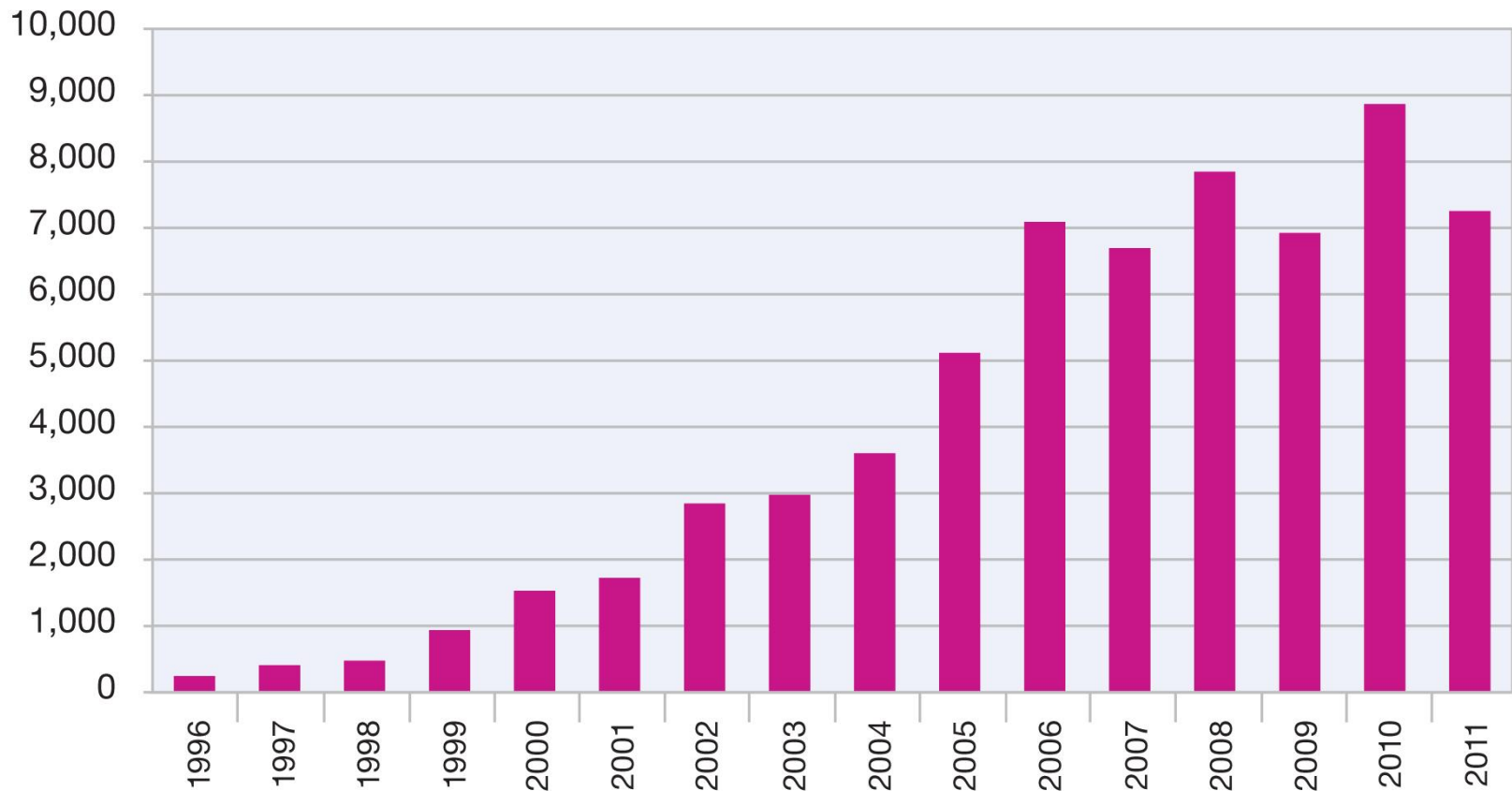


Source: <http://www.energysec.org/blog/quick-and-dirty-vulnerability-trending/>

Total Vulnerabilities Disclosed

Vulnerability Disclosures Growth by Year

1996-2011



Source: IBM X-Force® Research and Development

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?!?!?!?

How We Got Here



History time! Early on...

- Security mailing lists
- Phrack
 - 1985
 - attacker focused
- 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!

- 8lqm (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

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....

Poor communication on the inside of vendors

- led to confusion/panic in customers
- lawyers involved
- slow patching / solutions
 - sometimes attackers could exploit it quicker

Still a problem with small startups and small companies

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 (why stop?!?)
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>

Current Status of Industry

- Coordinated Vulnerability Disclosure
 - "We swear we won't sue you"
 - Vendor accepts responsibility for security issues :D
- vendorsec Mailing Lists
 - Invite-only mailing list for sharing vulnerability details and research (Bad idea??)
 - Compromised in 2011
- Delayed Disclosure
 - Issue PR release (vuln found in XYZ!)
 - Delay to disclose vuln details at major conference (Black Hat, Defcon, etc..) ... patch may not be out!

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

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

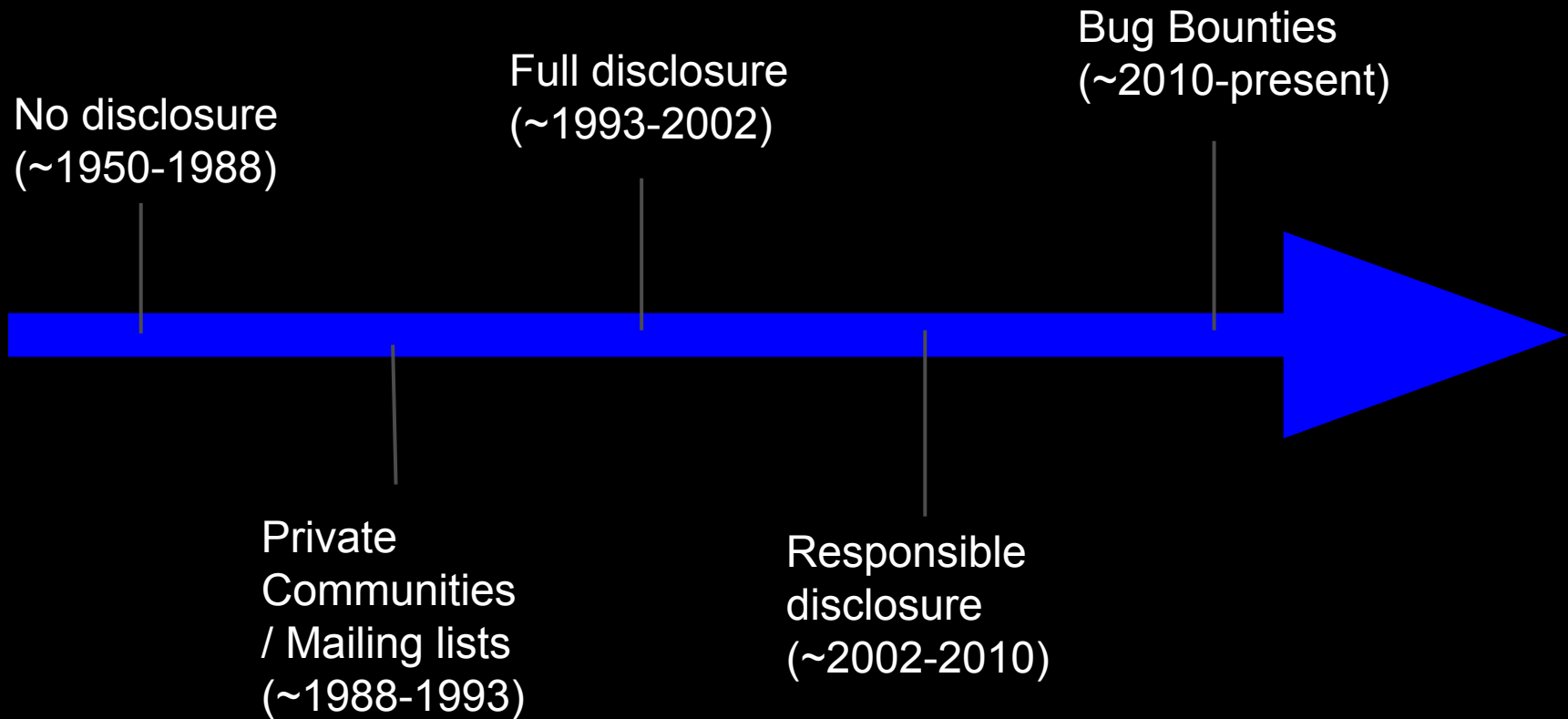
Company	Scope	Bounty	URL
tarsnap	Web & Apps	\$1-\$2,000	http://www.tarsnap.com/bugbounty.html
Wordpress	Web	\$100-\$1,000	http://www.whitefirdesign.com/about/wordpress-security-bug-bounty-program.html
Hexrays	Software	\$5,000	http://www.hex-rays.com/bugbounty.shtml
Payball	Web / Apps	unknown	https://cms.paypal.com/cgi-bin/marketingweb?cmd=_render-content&content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

Bug Bounties and Disclosure Websites

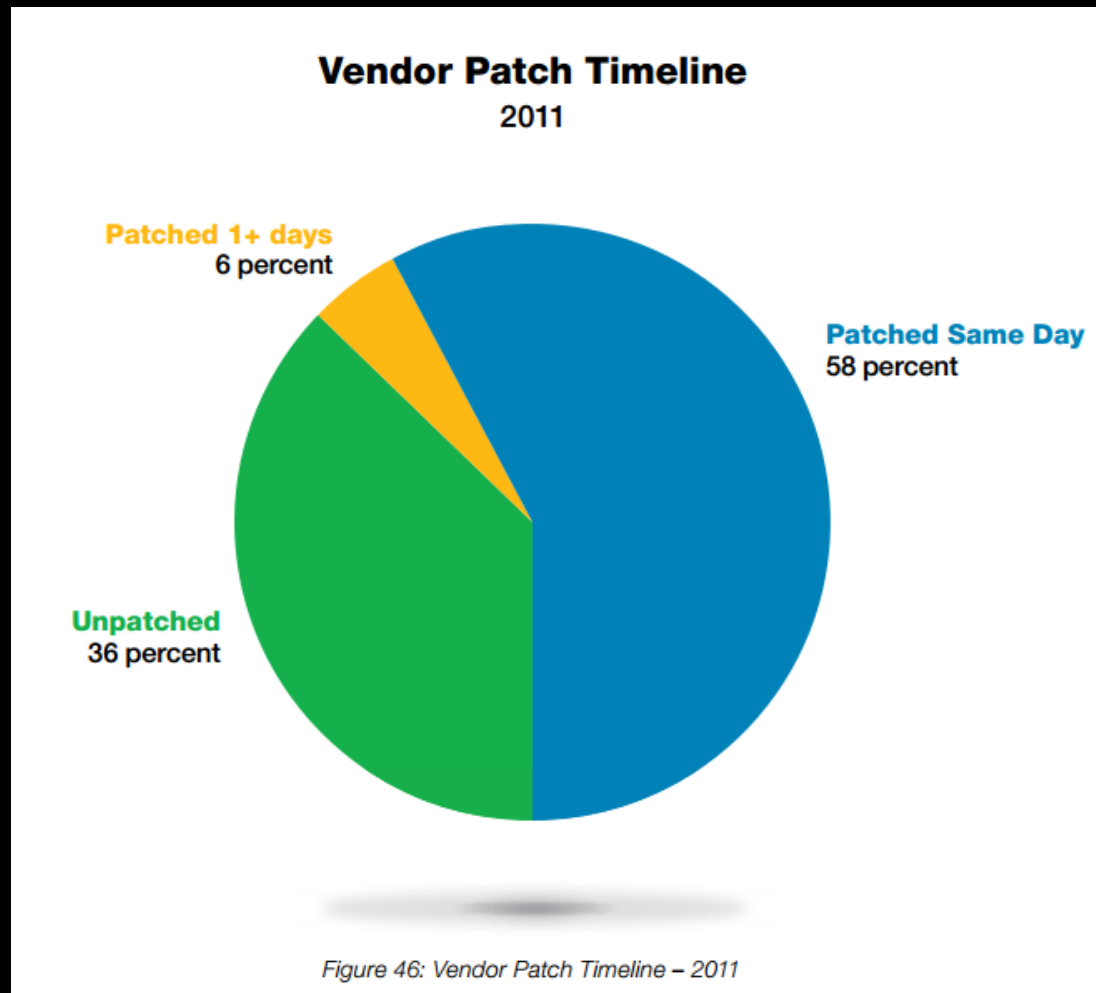
Huge list here:

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

Timeline

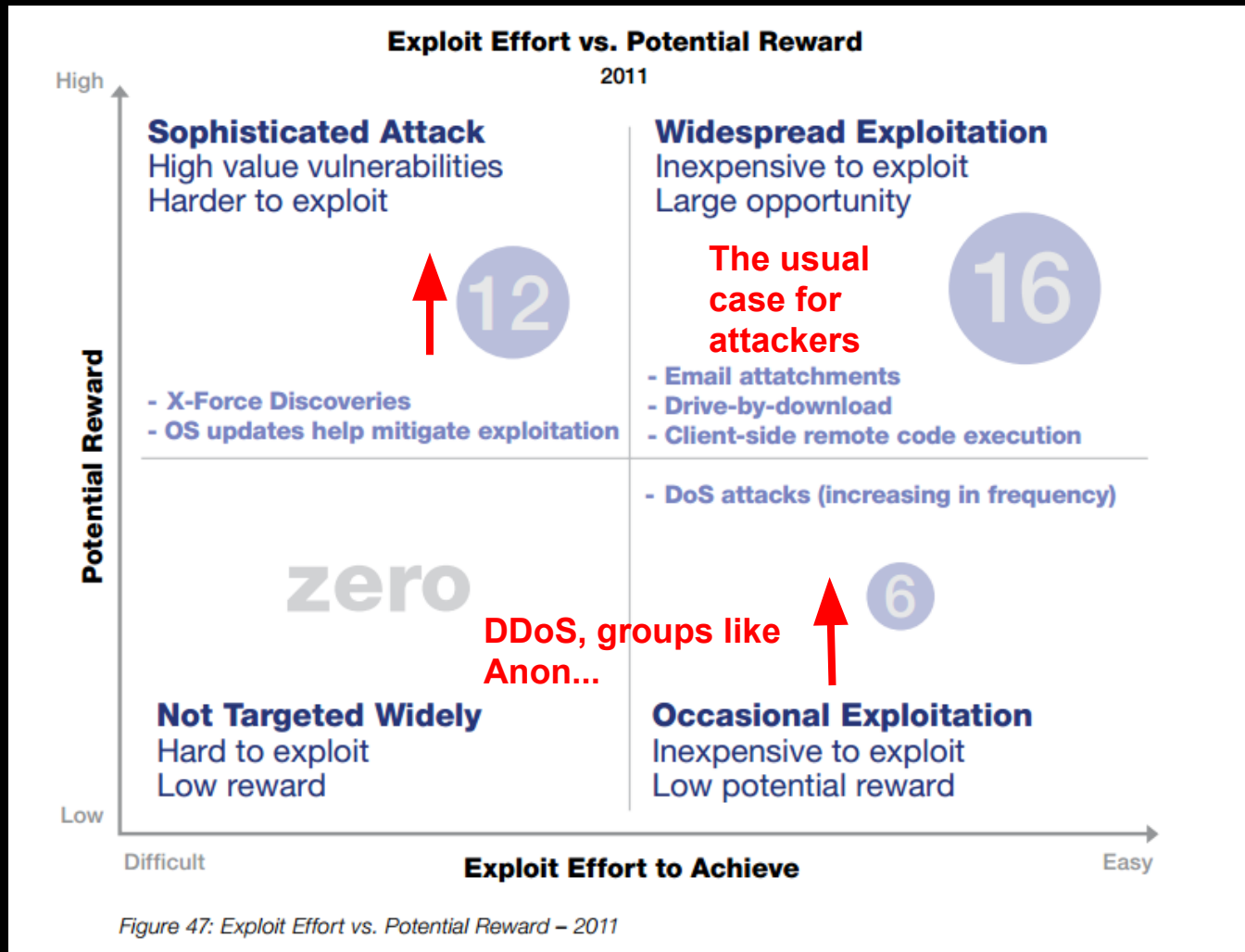


Vendor's Patching Trends got better



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

Bad Guy Trends

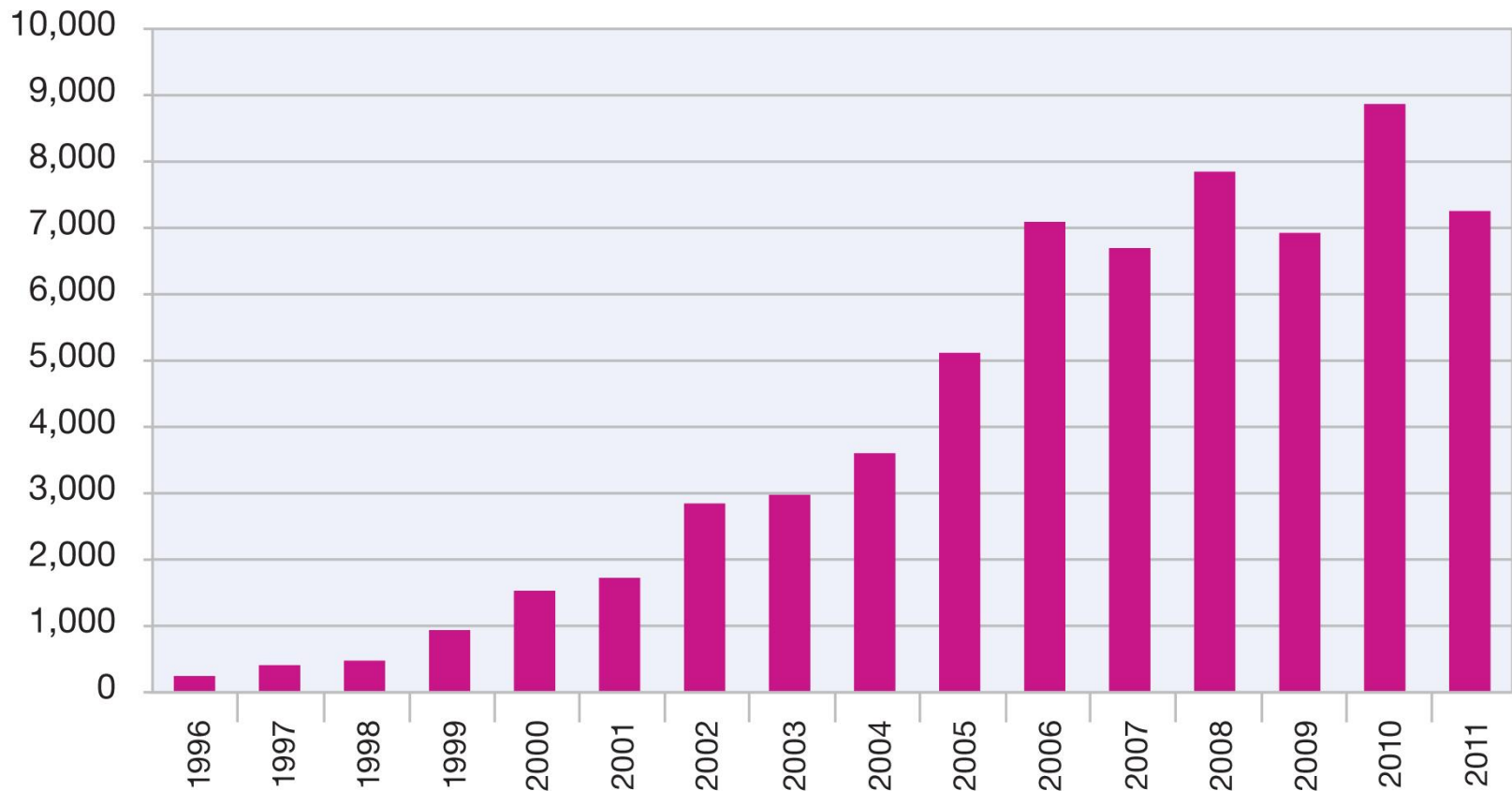


Source: IBM's X-Force 2011 Trend and Risk report

Total Vulnerabilities Disclosed

Vulnerability Disclosures Growth by Year

1996-2011



Source: IBM X-Force® Research and Development

Are things getting worse?

More and more vulnerabilities!!!

No

**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:

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

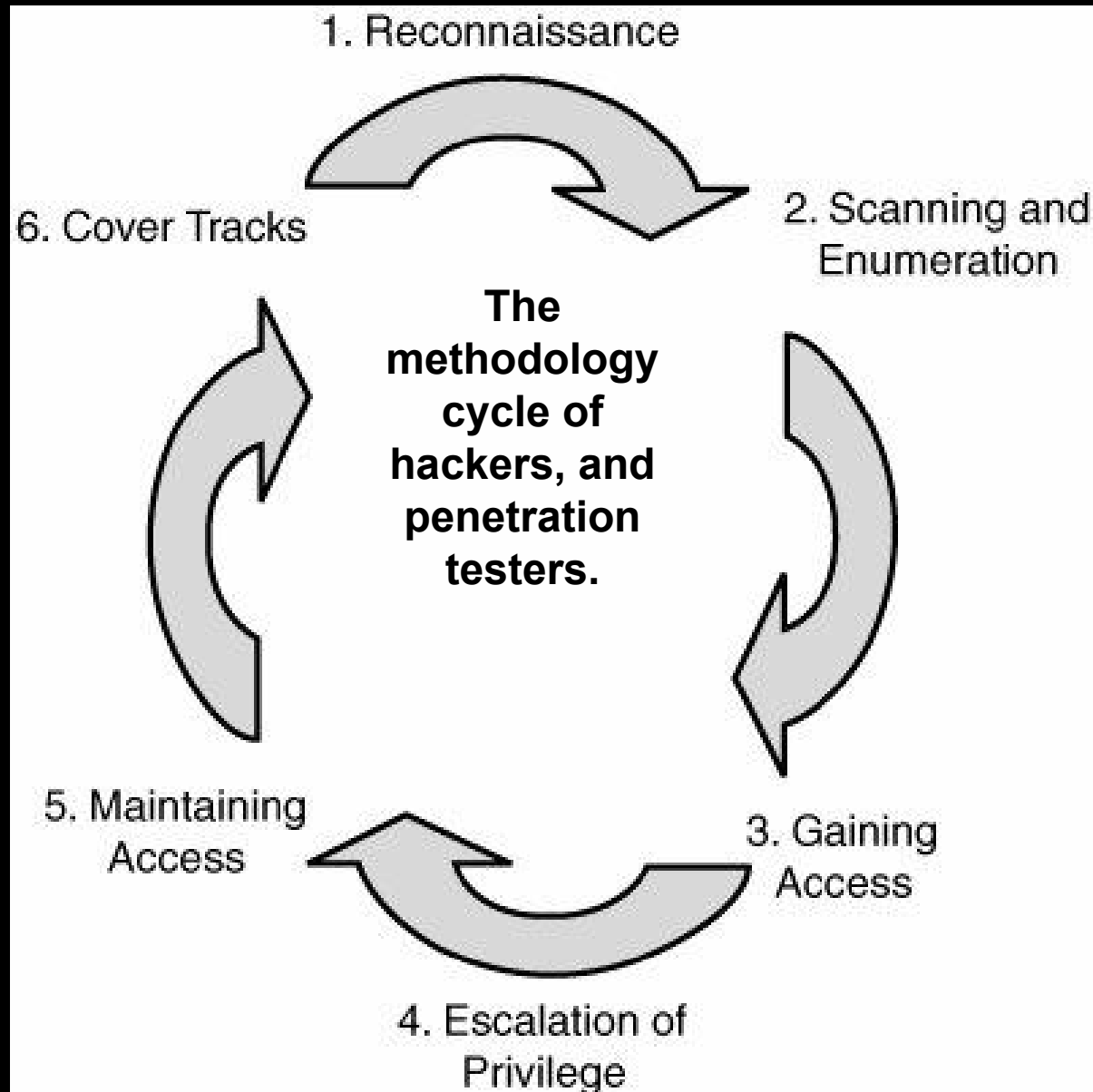
http://www.youtube.com/watch?v=q000_EOWy80

- 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 not wanted to damage."
 - Likely a troll

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
 - a. physical access or just remote access?
 - b. social engineering allowed?
 - c. covert or overt
2. Rules of Engagement
 - a. What is off limits
 - b. Threat model (insider threat, ex-employee, outsider, etc)
 - c. 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

Threat models

3 general model types

1. Attacker centric

- a. starts with an attacker and evaluates their goals and how they might achieve them

2. Software centric

- a. starts with the design of the system, and attempts to step through a model of it... looking for attacks against each aspect of it
- b. *i.e. Microsoft's Security Development Lifecycle*

3. Asset-centric

- a. starts from the assets in a trusted system.

Categorizing Threat

The key is understanding the capabilities posed by threats.

The number of threats is continually increasing.



from wikipedia

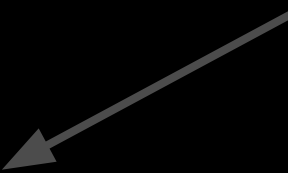
But why

$RISK = THREAT \times VULNERABILITY$

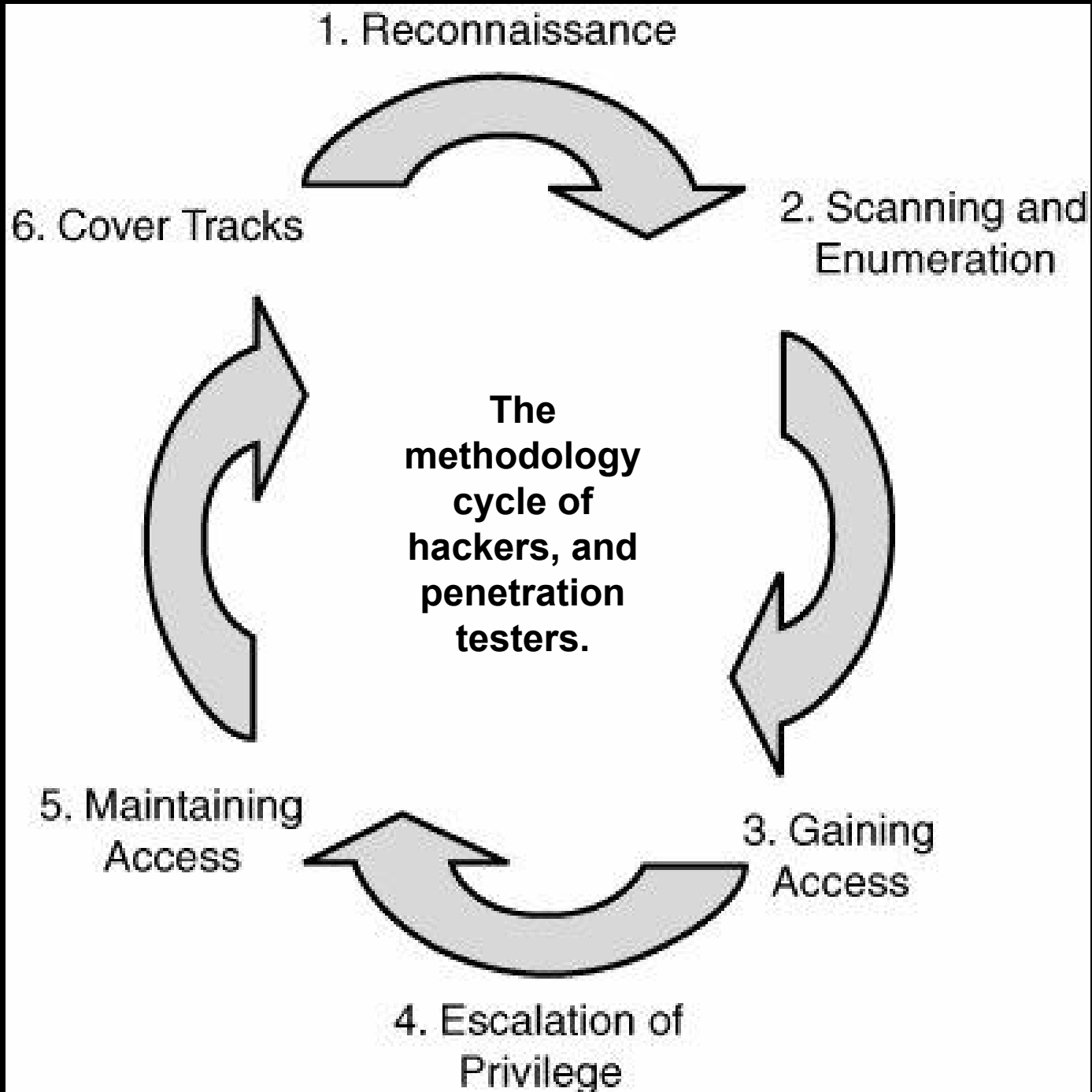
it is important to express the threat model when discussing vulnerabilities to help assess risks

Bad guy goals

On the
rise :(



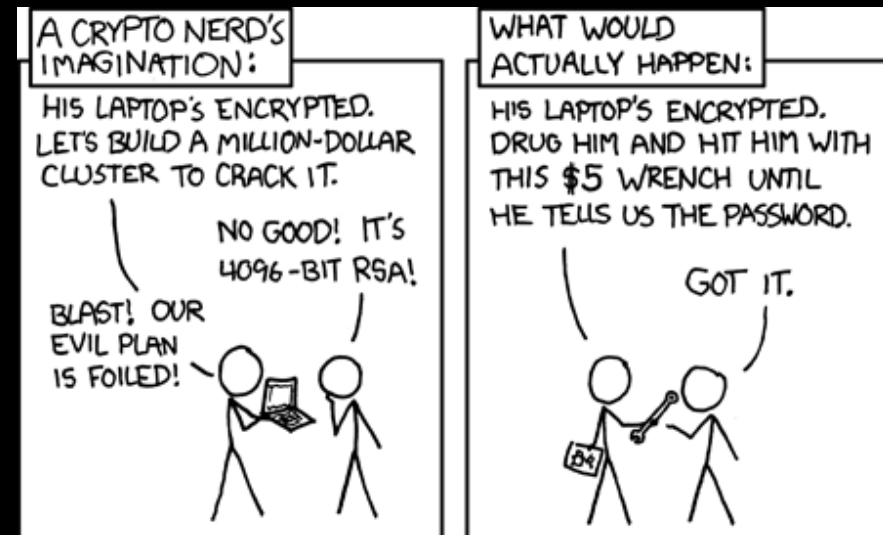
- \$\$\$
- Critical Infrastructure (i.e. proj Night Dragon)
- PII, enemies, political dissidents (operation Aurora)
- credit cards, financial data (Sony ps3 hack)
- passwords, password hashes (every hack)
- TOTAL Corporate Sabotage (HBGary hack)
- partner companies / 3rd parties (too many)
 - they will pivot from your systems to attack partners
- LONG TERM backdoors into your system
- intellectual property (most APT hacks)
- *and anything for the lulz*



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



Real World...

Thats why pen testing and incident responders are so important

Doubts?

Can't we just fix this crap by:

- 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!!

Come on already its 2014!!!

- keeping everything patched?
- etc... **I really wish, but its not likely!**
:(

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