CSCI 476

The Fundamentals of Software Security

Learning Objectives

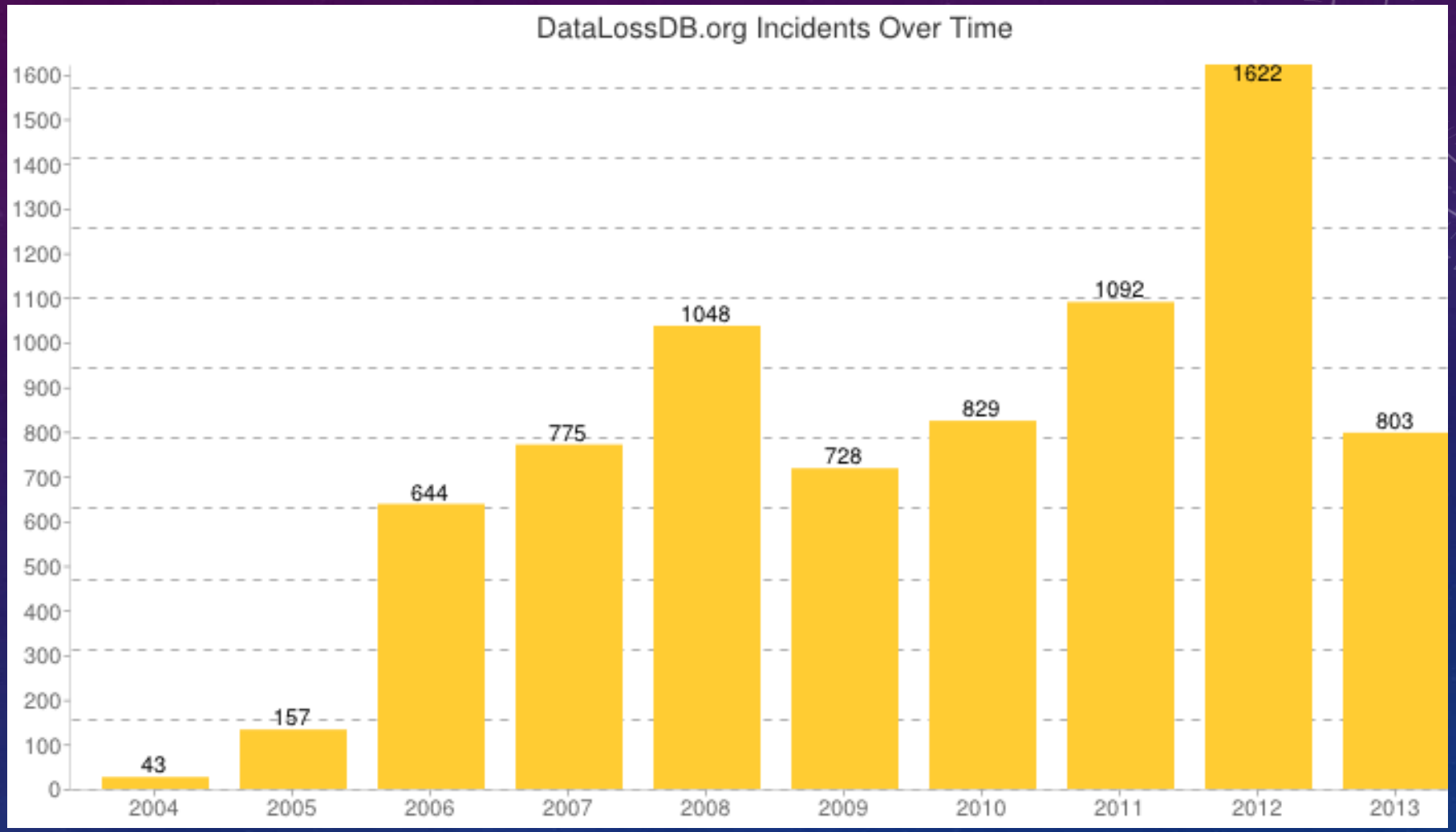
- Why application security is important to modern businesses
- Recent trends in software security
- Why software is not secure and what is needed to make it so
- The nature of application security
  - Vulnerabilities
  - Threats
  - Exploits
  - Risk

Introduction

- Businesses have always faced risks and threats
  - Before the adoption of IT and after
  - From external and internal agents
  - Accepted aspect of doing business
  - Security measures are introduced to mitigate threats and manage risk
- Businesses have rapidly adopted IT
  - Enabled newer ways of doing business
  - New threats and risks have emerged
  - Need to manage rapidly increasing threat spectrum

The Connected World Presents Challenges

Trends in Security Incidents



Recent Breaches

SC Magazine > Blogs > The Data Breach Blog

THE DATA BREACH BLOG

Unencrypted laptop stolen, 11,000 dialysis patients impacted

BY ADAM GREENBERG NOVEMBER 07, 2013

More than 11,000 patients and some employees of Colorado-based kidney care company DaVita are being alerted after an unencrypted laptop containing their personal data was stolen from a staffer's vehicle.

RELATED TOPICS
Breaches
Crime
Health Care

Cleveland hospital's unencrypted hard drive stolen, thousands affected

BY ADAM GREENBERG NOVEMBER 05, 2013

More than 7,100 patients who received care at University Hospitals of Cleveland are being notified that an unencrypted hard drive containing their data has been stolen.

RELATED TOPICS
Breaches
Crime
Vulnerabilities

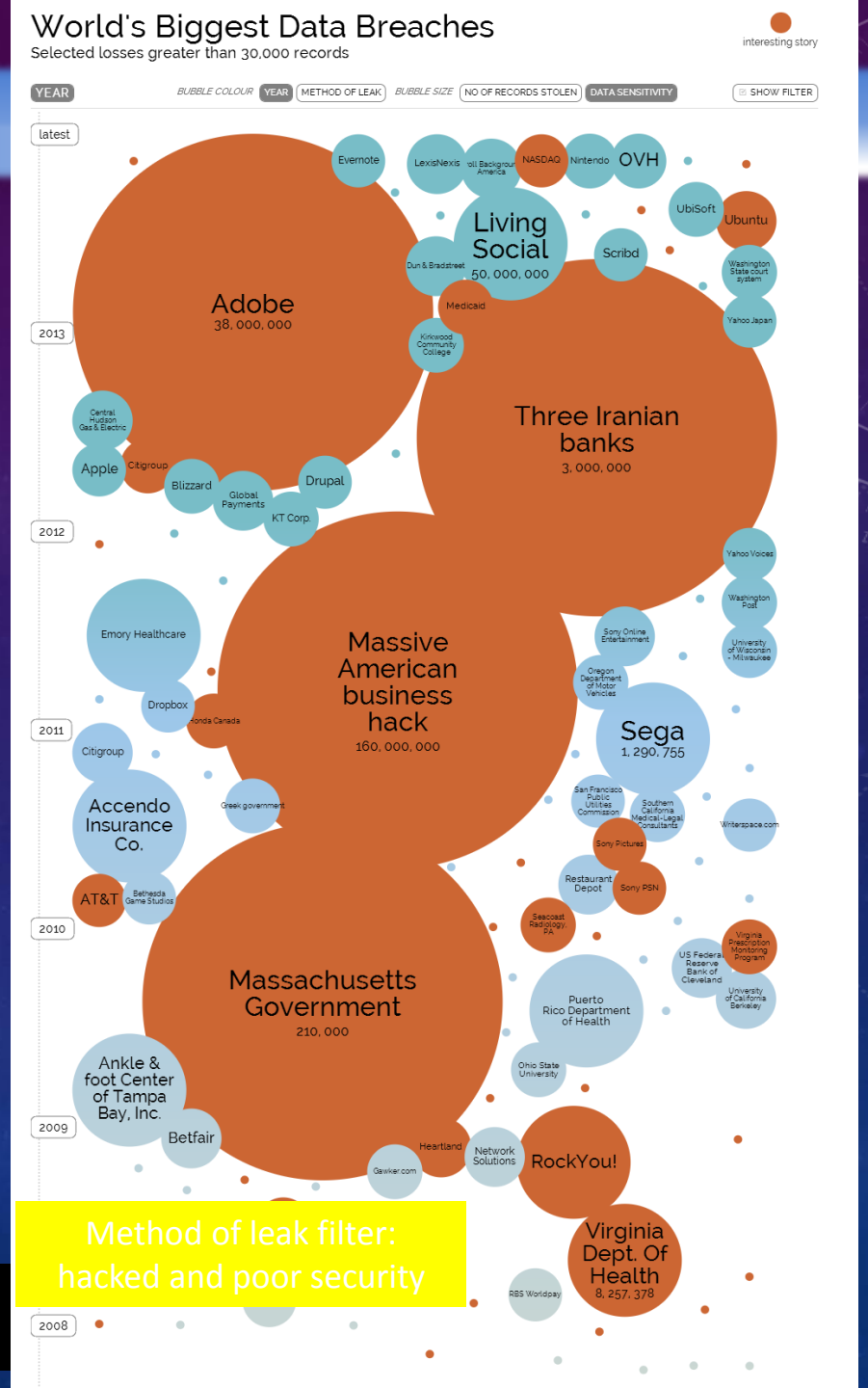
Baltimore County workers' personal information stolen

BY MARCOS COLON NOVEMBER 01, 2013

The personal information of current and past Baltimore County employees was stolen by a former employee of a county information technology contractor.

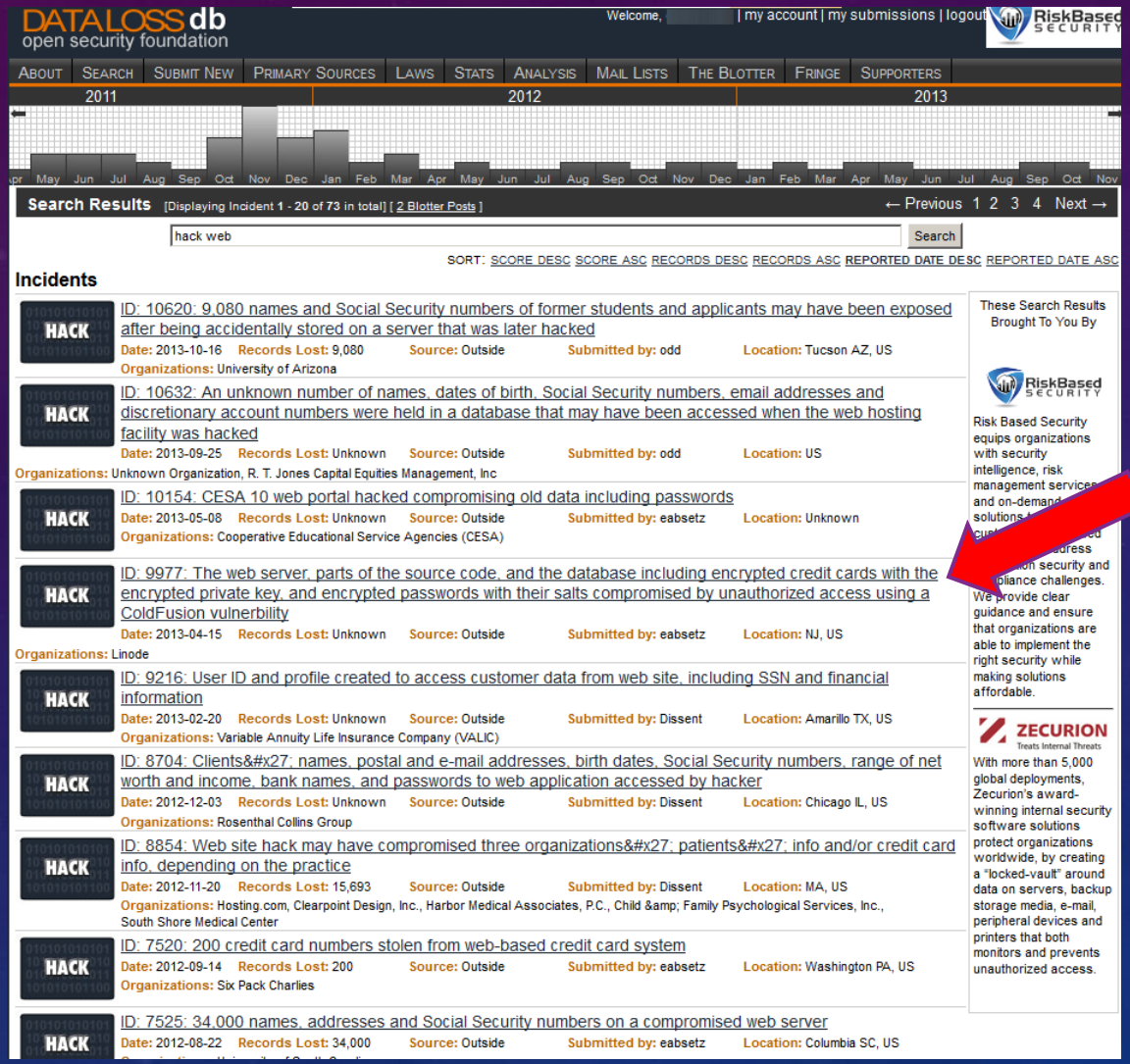
RELATED TOPICS
Breaches

http://www.scmagazine.com/the-data-breach-blog/section/1263/



http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

Dataloss DB



- This is a list of publically reported events related to hacking or web activity
- There are many more unreported incidents related to hacking and web activity

What Are The Threats To Your Business?

- Do you handle sensitive data?
  - If you lose some, what will it cost?
- Do you have web applications?
  - Why is that a problem?
- Are you certified for any compliance requirements like PCI-DSS, HIPAA or DISA?
- How much does it cost to patch your product in an emergency?
- Would your organization's reputation be hurt by a publically available security exploit?

Recent Incidents

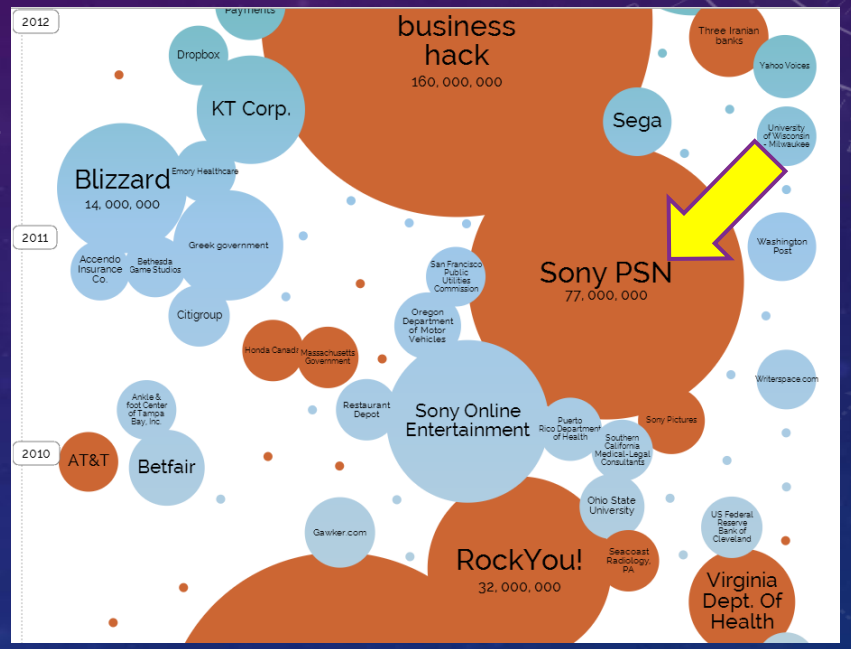
- RSA SecurID is used by companies globally for two-factor authentication
- Compromised on March 17th 2011
  - Phishing attack on two groups of RSA employees
  - MS Excel file which exploited Adobe Flash backdoor and allowed remote access to machines
  - RSA says it needs to replace all 40 million devices
  - RSA has spent $66 million till August 2011 to help its customers fix the problem

Recent Incidents

- April 2011
  - L-3 Communications attacked
  - Hackers used cloned SecurID devices
- May 2011
  - Lockheed Martin attacked
  - 45,000 SecurID tokens needed replacement
- June 2011
  - Northrop Grumman potentially compromised
  - Stopped all forms of remote access
- Banks worldwide under threat
  - Estimated cost of $100 million to fix

Recent Incidents

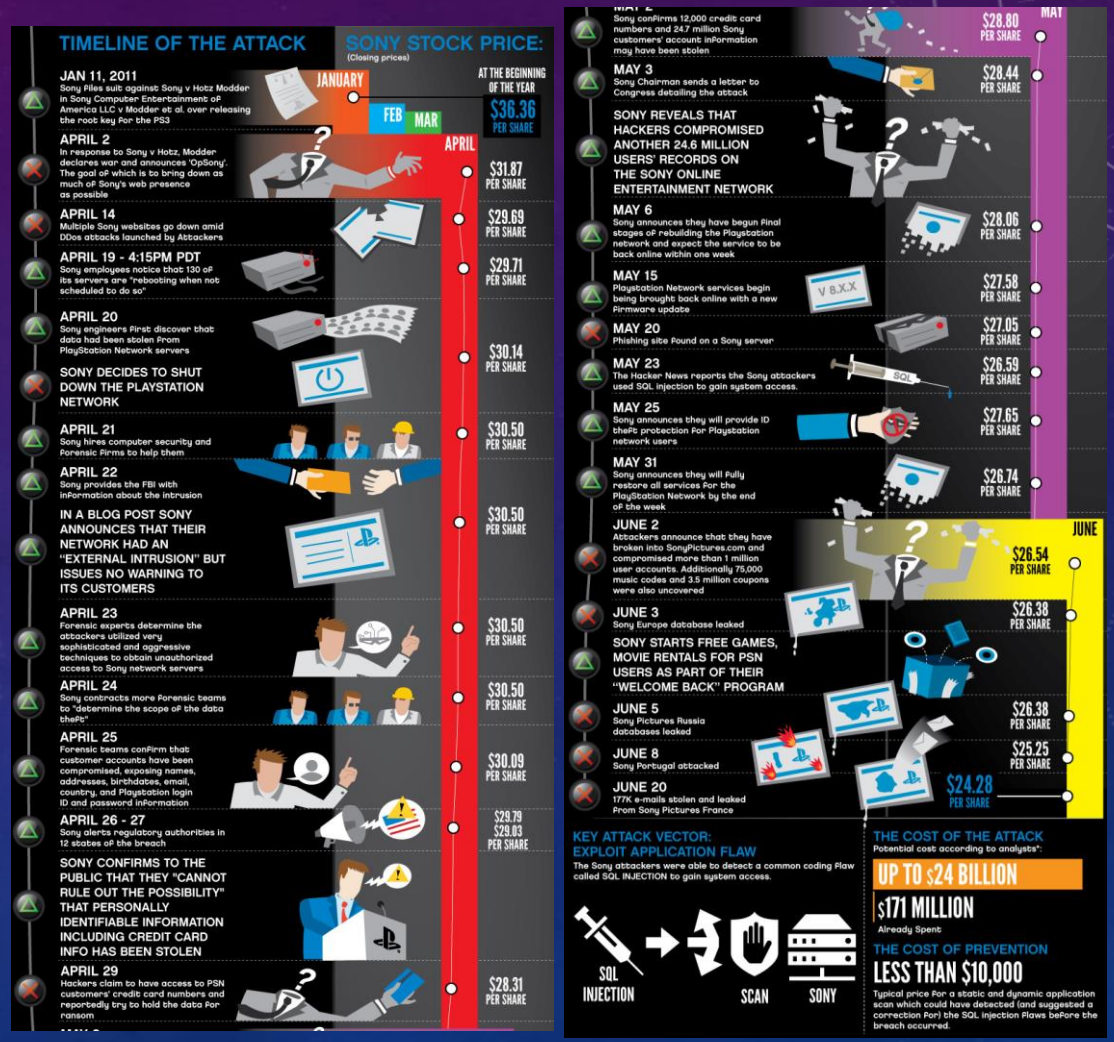
- Sony PSN
  - 77 million registered users
- Compromised on April 19th 2011
  - Application server behind a web-server and two firewalls was hacked via a known vulnerability
  - Servers rebooted randomly
  - Parts of personally identifiable information (PII) of all users stolen
  - Servers taken offline to protect further breach, outage lasted 23 days
  - Sony says breach cost them $171 million



http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

Recent Incidents

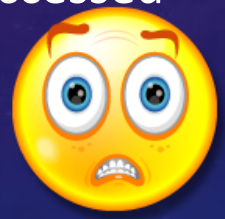
- April 26th 2011
  - Sony acknowledges that customer information was stolen
  - Governments and customers condemn delayed notification
- Sony is facing class action lawsuits in several countries
  - £250,000 fine from UK Information Commissioner's Office ("ICO")
- Simple Google searches found weaknesses in web pages
  - The Java security console was accessible from some web pages



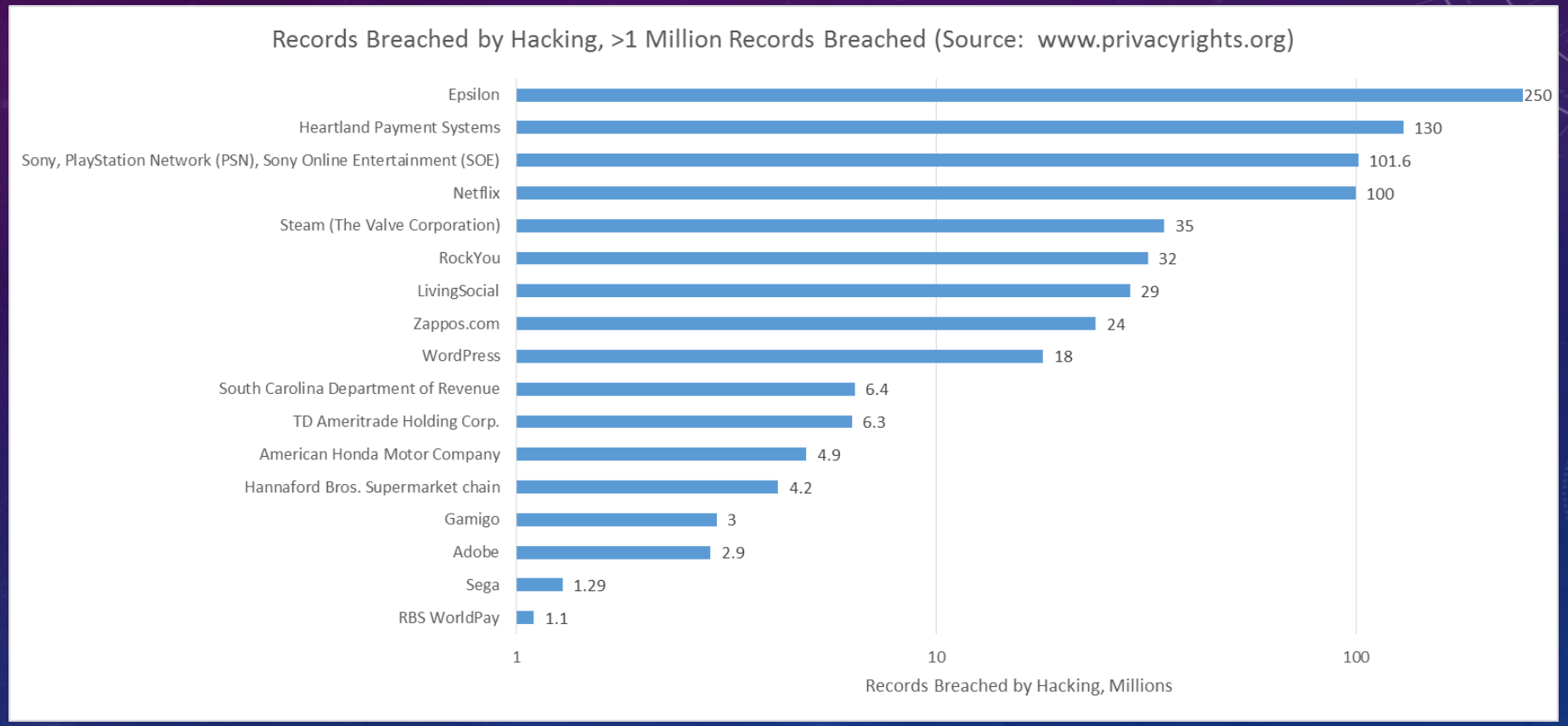
http://www.veracode.com/resources/sony-psn-infographic

Recent Incidents

- 2013
  - 110 million users were affected by a breach at Target
  - U.S. Federal Reserve Bank internal site hacked by Anonymous and data posted publically
  - Washington State Office of the Courts hacked and sensitive data accessed
  - LivingSocial hacked and 50 M poorly encrypted passwords stolen
- 2012
  - Global Payments breached and 1.5 M unencrypted credit card numbers stolen
  - Stratfor breached and thousands of unencrypted credit cards numbers used to make donations to charitable institutions
  - Digital Playground breach nets 40K credit card numbers, CCV numbers and expiration dates (OUCH!)



Millions of Records Lost



Cost of Losing Records

- Facebook: $80 Million
- WordPress: $18 Million
- Texas Comptroller's Office: $3.5 Million
- American Honda Motor: $4.9 Million
- Netflix: $100 Million
- RockYou: $32 Million
- U.S. Military Veterans: $76 Million
- Heartland Payment Systems: $130 Million
- RBS: $1.5 Million
- Countrywide Financial Corp: $17 Million
- Bank of New York Mellon: $12.5 Million
- TJX Corporation: $95 Million
- Ameritrade: $6.3 Million customer
- Fidelity National: $8.5 Million

This includes only the *most recent* breaches that lost more than 1 Million records!

Cost To Remediate An Application Security Incident

- Average total cost to remediate a single application security incident is approximately $300,000
- Average total annual investment (people, processes, technology) in application security initiatives is $400,000
- One average application security incident can almost wipe out the annual application security initiatives expenditures

Analyst Insight

Aberdeen Group
A Harte-Hanks Company

August 2011

Securing Your Applications: Get Started Now

If your organization hasn't gotten started yet in the area of application security – in spite of the **dynamic nature of the application security threat landscape**, the **size and diversity of your application software portfolio**, and the **significant financial impact of the average application security-related incident** – do it because of the **positive impact on your bottom line**. This Analyst Insight reviews several practical steps you can take to get started now.

Business Context: The Biggest No-Brainer in Security?

New headlines provide ongoing evidence that IT Security teams are losing the battle against attackers, reinforcing the need to address the security of enterprise applications. In the recent CitiGroup breach, for example, more than 200,000 cardholders had their names, email addresses, account numbers and transaction histories exposed as a result of a well-known application security vulnerability. As reported by the *New York Times*:

- *The data thieves were able to penetrate the bank's defenses by first logging on to the site reserved for its credit card customers. Once inside, they leapfrogged between the accounts of different Citi customers by inserting various account numbers into a string of text located in the browser's address bar. The hackers' code systems automatically repeated this exercise tens of thousands of times – allowing them to capture the confidential private data.*

In the language of the application security community, this is referred to as a *direct object reference*, which occurs when attackers are able to manipulate direct references to an internal implementation object (e.g., a file, directory or database key) to access unauthorized data. It's actually on the Top 10 list of web application security threats identified by the Open Web Application Security Project (OWASP).

But this Analyst Insight is not about fear-mongering or sensationalizing the latest headlines to gain your focus on securing your applications. It is about your organization's bottom line.

Why You Should Get Started Now

Aberdeen's benchmark study and several follow-on publications (see the *Related Research* section at the end of this report) showed that *all* respondents experienced a positive return on their annual investments in application security – not only the leading performers ("Best-in-Class"), but also the lagging performers ("Laggards").

Analyst Insight

Aberdeen's Analyst Insights provide the analyst perspective of the research as drawn from an aggregated view of surveys, interviews and analysis.

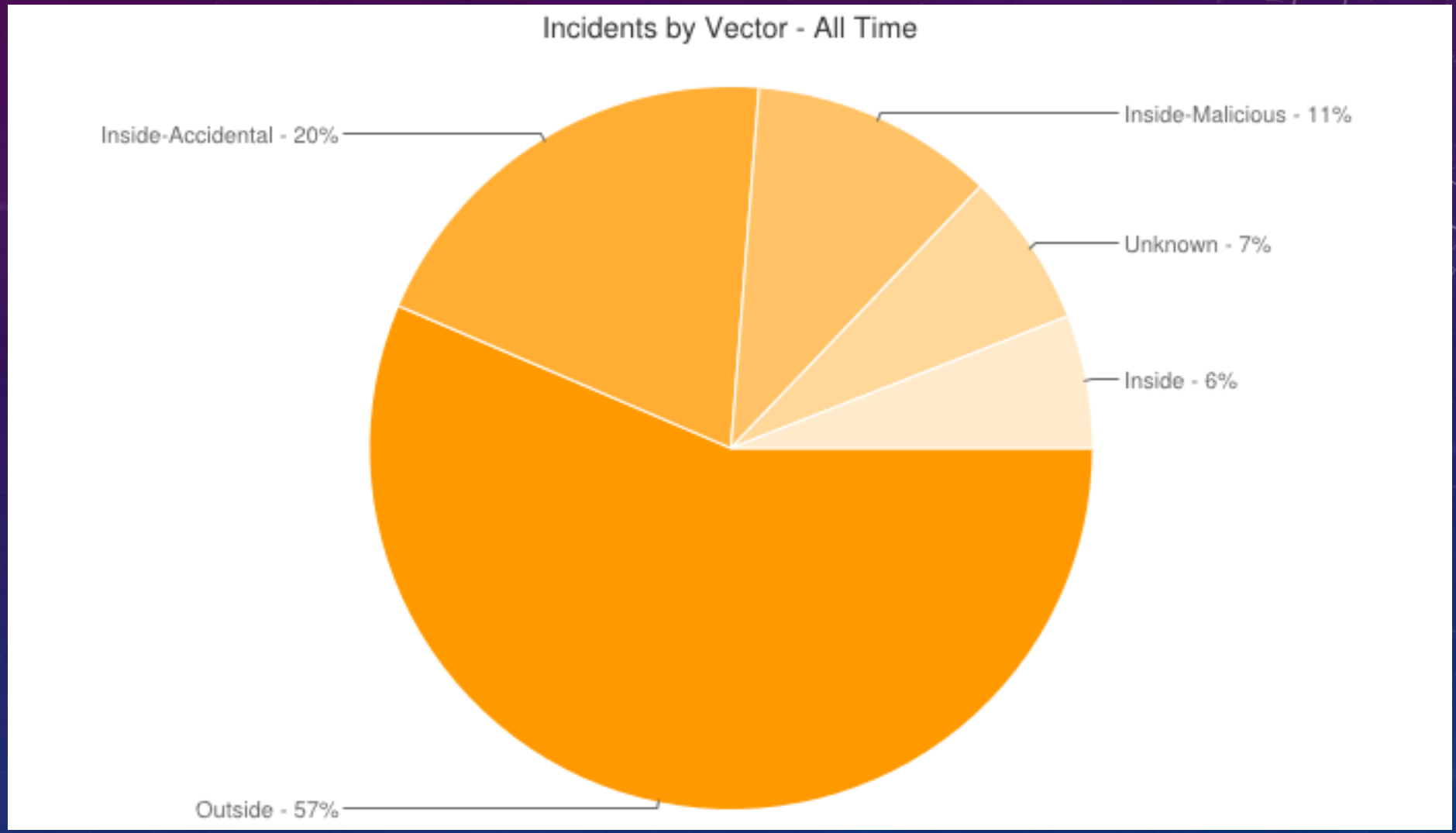
Top 10 Web Application Security Vulnerabilities

- √ Injections
- √ Cross-site scripting
- √ Authentication and session management
- √ Direct object references
- √ Cross-site request forgery
- √ Security misconfiguration
- √ Insecure cryptographic storage
- √ Failure to restrict URL access
- √ Insufficient transport layer protection
- √ Unvalidated redirects and forwards

Source: Open Web Application Security Project, 2010

http://www.aberdeen.com/Aberdeen-Library/7307/AI-application-security-vulnerability.aspx

Where Do The Attacks Originate?



Why Is the State of Application Security Suspect?

- Software has grown up in a trusting, insecure world
  - Systems have historically been built t share data and facilitate collaboratio
  - In the early days, trust was (safely) assumed
  - Software developers failed to see the danger in failed trust
  - The software industry has been slow to treat security as a required attribute of software
  - Software training has similarly failed to address to problem
  - The world is connected, so the nefarious among us have nearly universal access unless blocked

Security Is Not A Network Problem

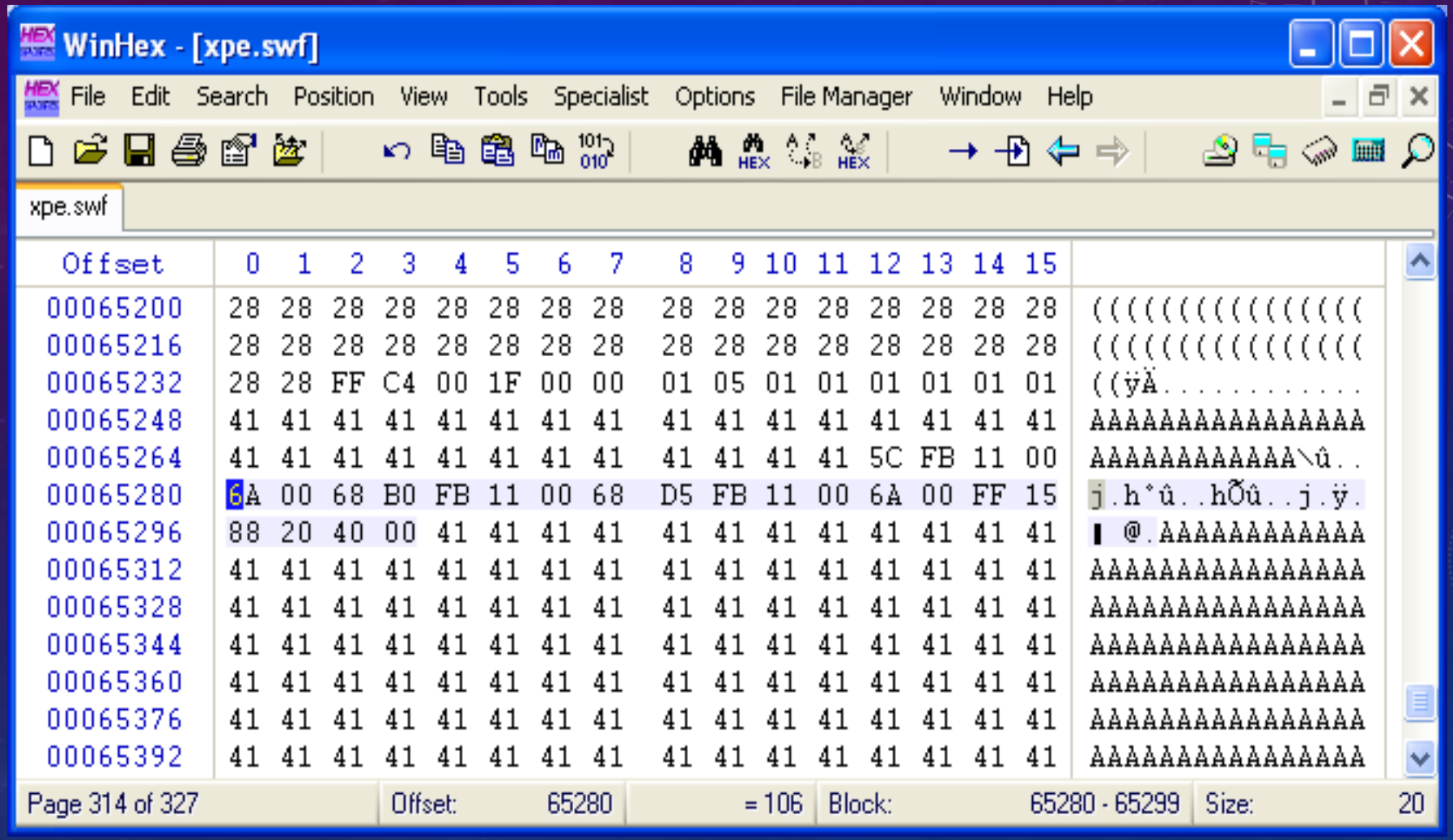
- Some data comes through a firewall destined for an application running on a server

6A0068B0FB110068D5FB11006A00FF1588204000

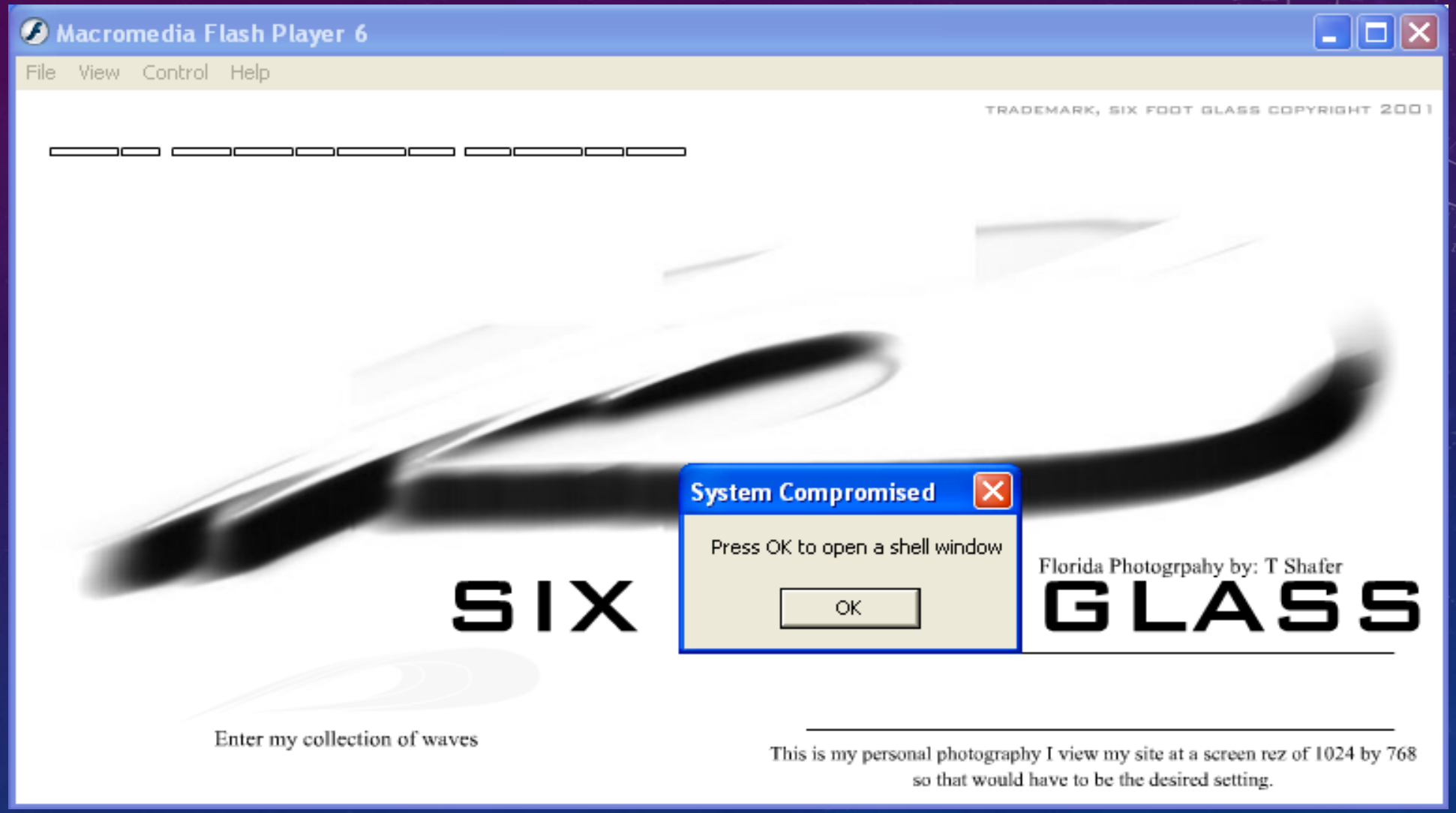
- Is it text?:
- Is it data? Part of a picture perhaps?
- Or is it…something else?

Firewalls Don’t Reach Into Data Files

Add Some Data To The Flash File



And Now We Have An Exploit



Security Is A Software Problem

- Over 70 percent of vulnerabilities are in software (Open Security Foundation)
- All networked applications require some openings in the perimeter security
- Vulnerabilities primarily result from flaws in applications or poor configurations
- Over 2700 vendors are listed in Bugtraq's vulnerability database

The Classic Conflict

- Application security is in conflict with most traditional measures of development performance
  - Complexity is the enemy of security
  - Good security takes time
  - More security typically means less user-friendliness
  - More security typically means less convenience

Total cost of ownership

- The most visible aspect of Total Cost of Ownership is security

Changing Attitudes About Security

- Customers want to minimize costs incurred due to insecure code or configuration
- Software security is a significant component of TCO
- Customers are asking security specific questions in RFPs:
  - What is your vulnerability response process?
  - What process improvements have you made as a result of vulnerabilities reported in your software?
  - Do you offer secure implementation guidance?
  - What training does your development and testing team receive on security?
  - What compliance certifications do you meet?

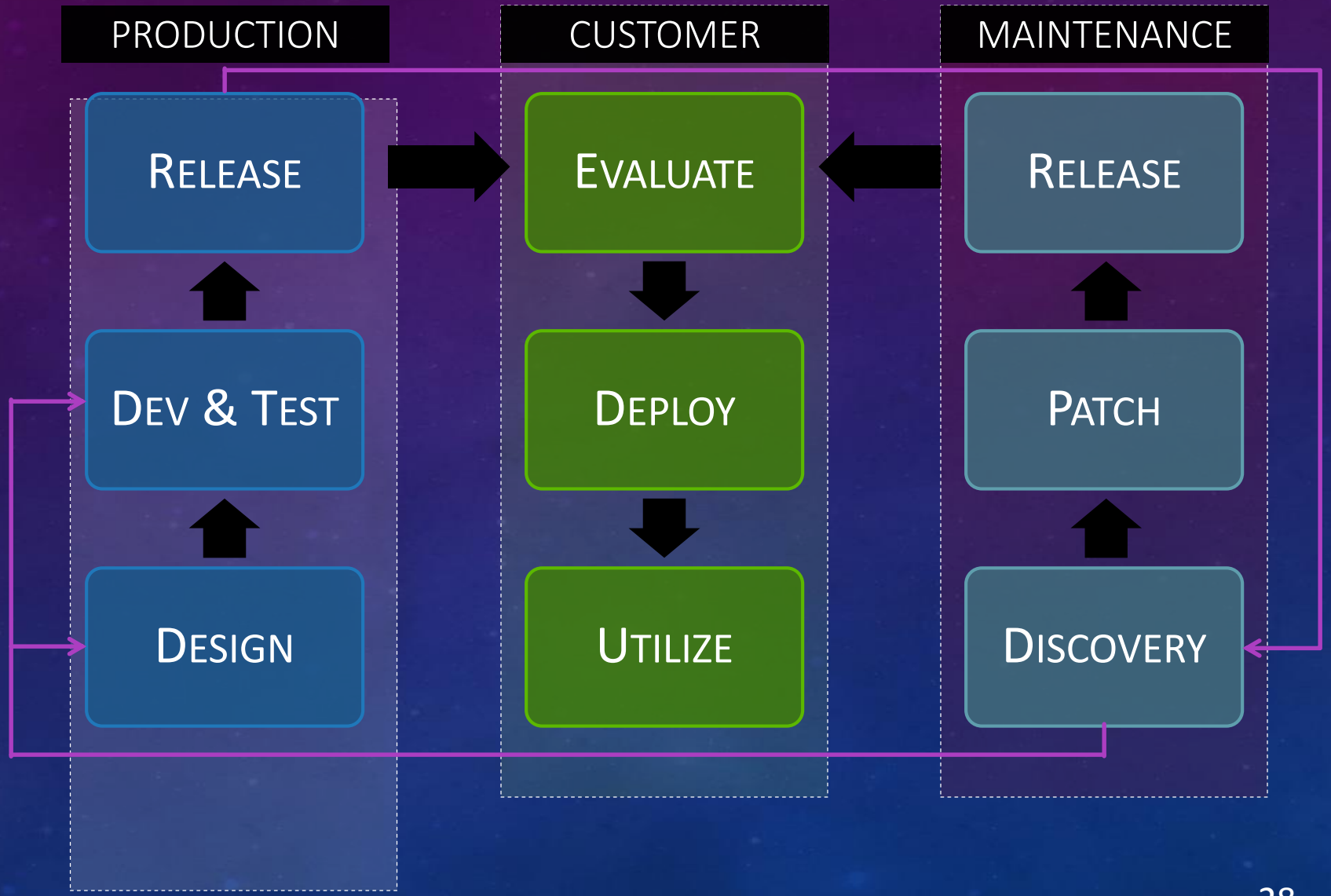
Current Industry Trends

- More regulations are being passed to protect end-users
  - Sarbanes Oxley (SOX)
  - California Senate Bill 1386 (SB1386)
  - Gramm-Leach Bliley Act (GLBA)
  - Health Insurance Portability and Accountability Act (HIPAA)
- Vendors are moving to managed platforms such as .NET and Java
- Attention to security throughout the lifecycle is expected
- Customers receive secure deployment guidelines
- A Security Response Team ready to manage vulnerabilities and exploits is the norm

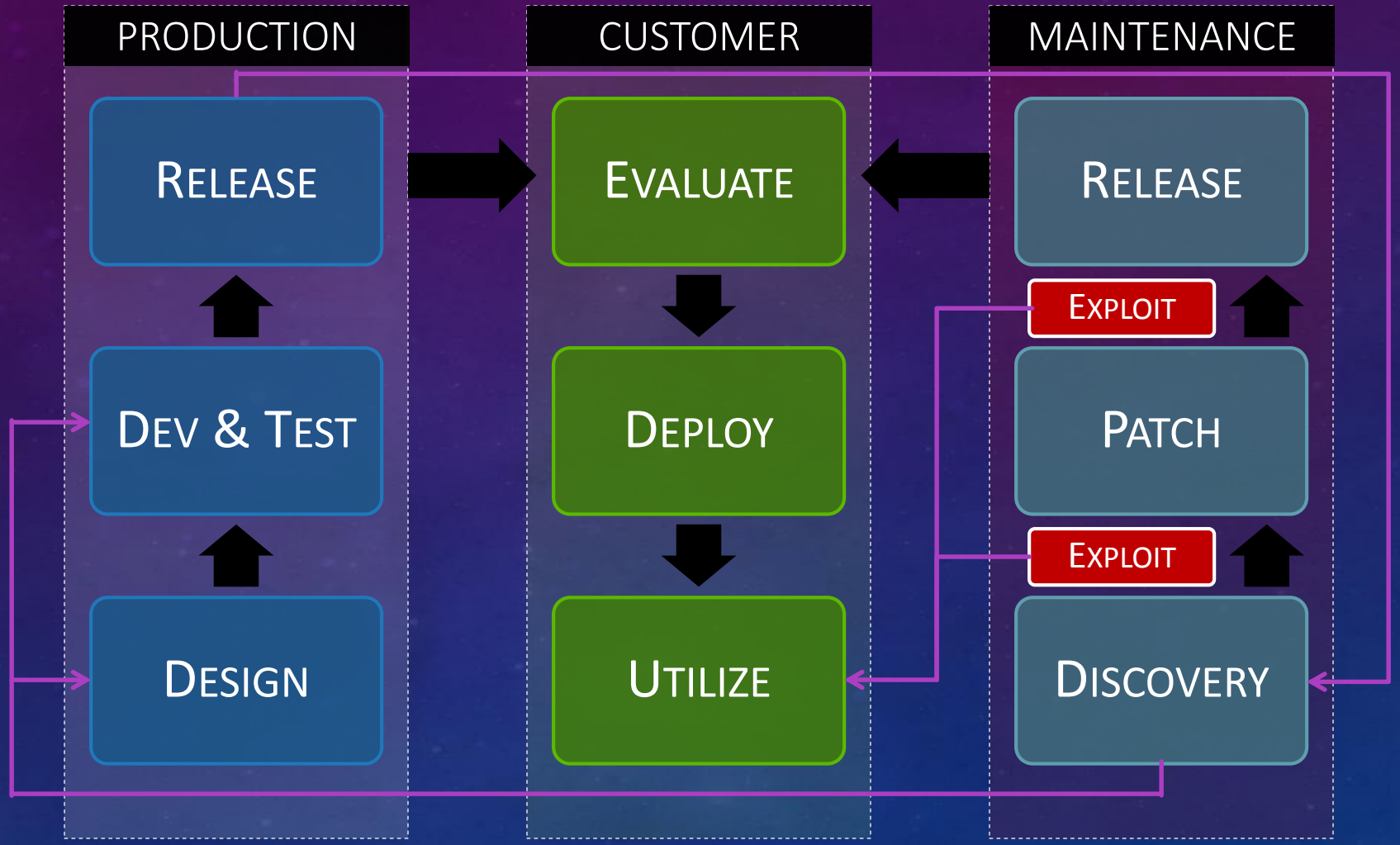
Solving The Application Security Problem

- Security measures for vendors
  - Proactive
    - Employee education & training
    - Threat Modeling & risk management
    - Secure design and coding policies
    - Security audits & penetrating testing
    - Secure deployment
  - Reactive
    - Secure patch management and upgrades
    - Security Response Teams
    - Security incident process

Software Development Lifecycle

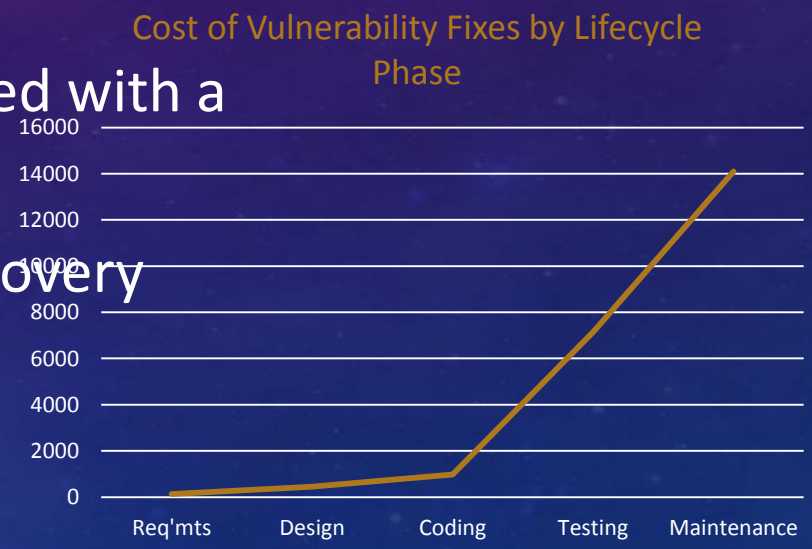


Software Development Lifecycle



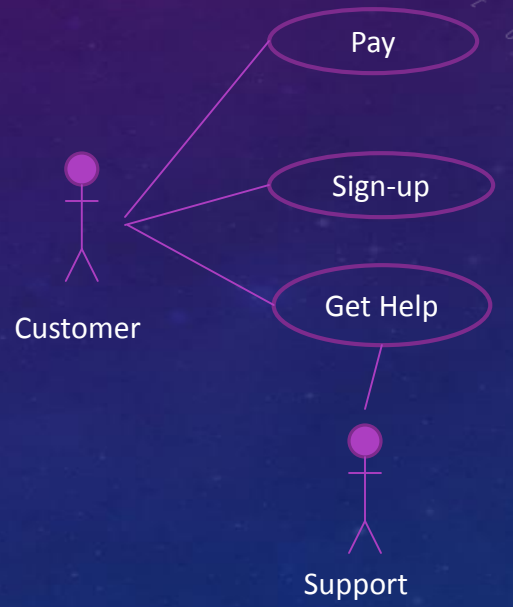
Software Development Lifecycle

- Vulnerabilities and attacks will happen
- Customers may take time in testing and deploying patches
- Each vulnerability opens a window of risk
- The number of vulnerabilities can be reduced with a Secure SDLC
- Cost is exponentially related to time-of-discovery



1. Identify Market Need

- The key is to identify security needs as early as possible
- Create valid and complete **use cases** that consider security implications
- Elicit security information from customers
  - Sensitive data to handle
  - Regulatory concerns and standards
  - Contractual requirements
  - Incoming and outgoing data



2. Establish Requirements

- Use the market analysis
- Plan for scope broadening and feature creep
- Accuracy is key
- Pay particular attention to:
  - What environments might this be deployed in?
  - What other products/components should this product work with and does it adhere to their security standards?
- Requirements must be explicit on expected behavior AND constraints on behavior.

3. General Design Considerations

- Many security flaws originate here
- The most difficult to fix flaws originate here because they are “baked-in”
- Pay special attention to:
  - Component endpoints (data passed from component to component)
  - Authentication
  - Resource protection
- Plan for future extensibility
- Consider supportability, deployability, extensibility and maintainability

3. General Design Considerations

- Build Threat Models
  - Test them against the design
- Create a list of the highest security risk components and schedule them for special attention during development and testing
- Establish secure coding guidelines
- Design tests and testing procedures
- Create a secure deployment strategy, especially for secure configuration

4. Development/Implementation

- By far, most of the vulnerabilities reported in software are the results of mistakes or bad decisions made during implementation
- Developers must take responsibility to ensure that the code they produce adheres to secure coding standards
- Developers must be alert for security issues that may have been missed in the design phase, or due to changes occurring during the coding
- Developers must be fully engaged in security

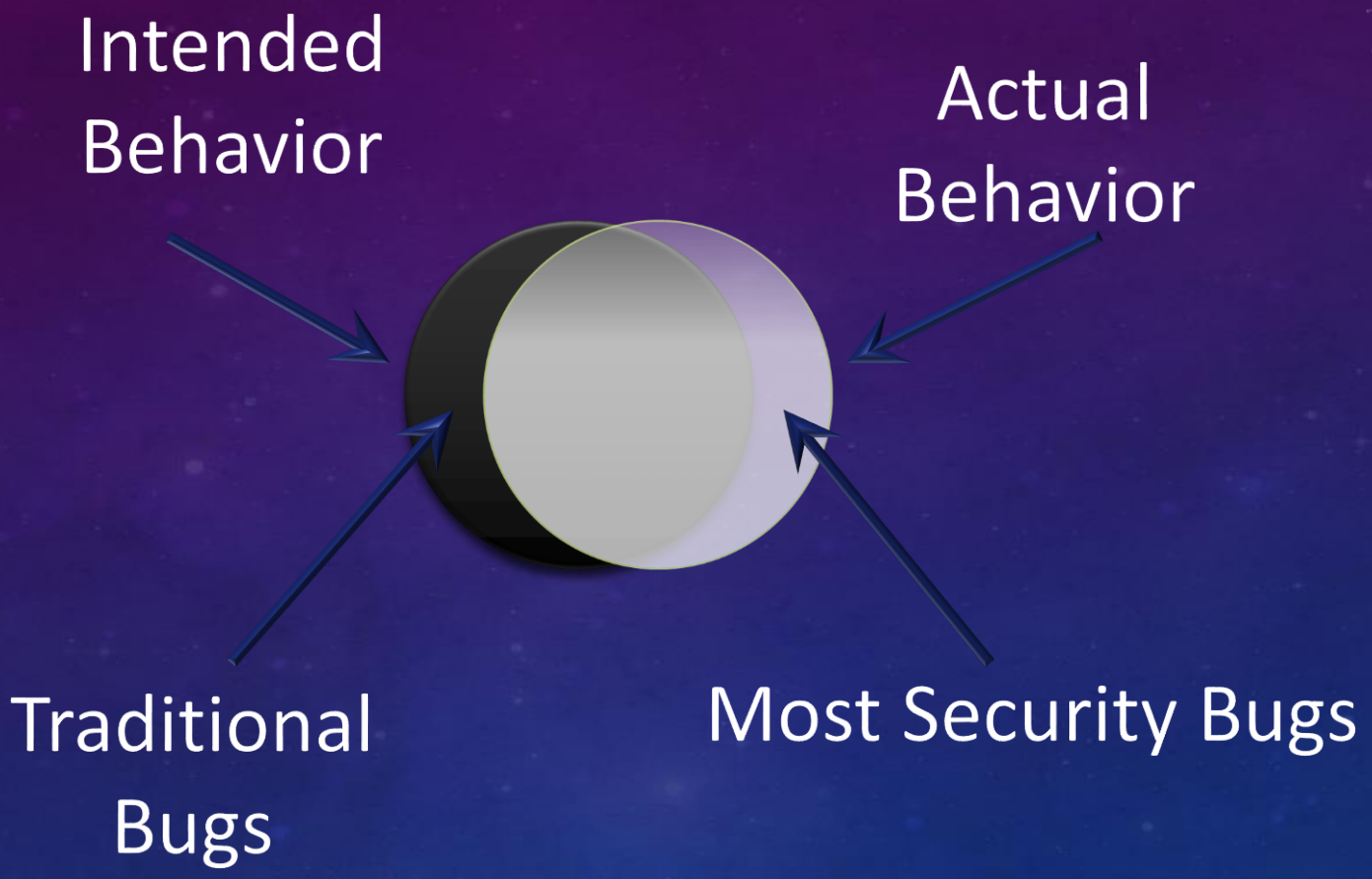
So Many Coding Mistakes…

- Common Coding Errors
  - Overflows, strings, integers, special characters and words
  - Paths, backdoors, temporary files, deletions and swap files
  - DLLs, 3rd party libraries
  - Race conditions
- Handling sensitive data and cryptography
- Assumptions are the bane of security

5. Feature/System Test

- Software Test Engineers are gatekeepers
- Testing for security is VERY different than testing for functional problems
  - It is focused on finding functionality that is NOT supposed to exist
  - There are a limited number of ways software can be right, but it the ways it can be wrong are uncountable
- Security test teams need special skills and test tools
- To test for security, “Think Like An Attacker”

How To Think About Testing Software Security



6. Deployment

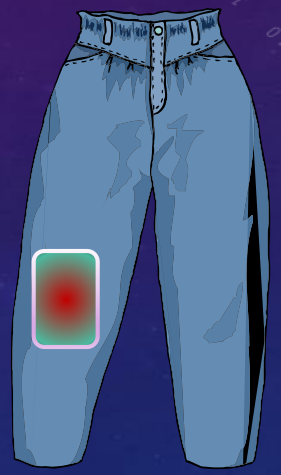
- Deliver a secure or readily securable deployed solution
- Bad deployment decisions can open many security holes
- Deploy secure by default
- Document how to deploy securely for internal and external consumption
- Provide documentation, standards, and procedures

7. Support

- Company business requirements evolve over time as do operating systems, components and environments
- Training support personnel on security is critical
- Help users maintain a secure environment

8. Update and Patch

- Updates and patches are inevitable
- Attackers are constantly looking for software flaws
- Clear versioning of the product and avoiding dependencies that break upgrades is vital
- Customers need to understand the contextual security implication of a patch
  - Detailed advisories
  - Timely delivery of updates
  - Notification of update availability



More on Updates

- Minimize downtime, and updates should not be so difficult that customers forego them
- Patch in a secure fashion
  - If a worm is running rampant, customers should be able to get on the network and update without being infected
- Created with customer needs in mind
  - Easy and manageable deployment
  - Grouped releases

Basic Security Objectives

Confidentiality

Is the information protected from unauthorized disclosure and observation?

Integrity

Is the information complete, whole and unchanged from the previous state?

Availability

Are information & systems available so that they can be accessed in a timely manner for the intended purpose?

Possession

Is the information in the control of the authorized individuals?

Authenticity

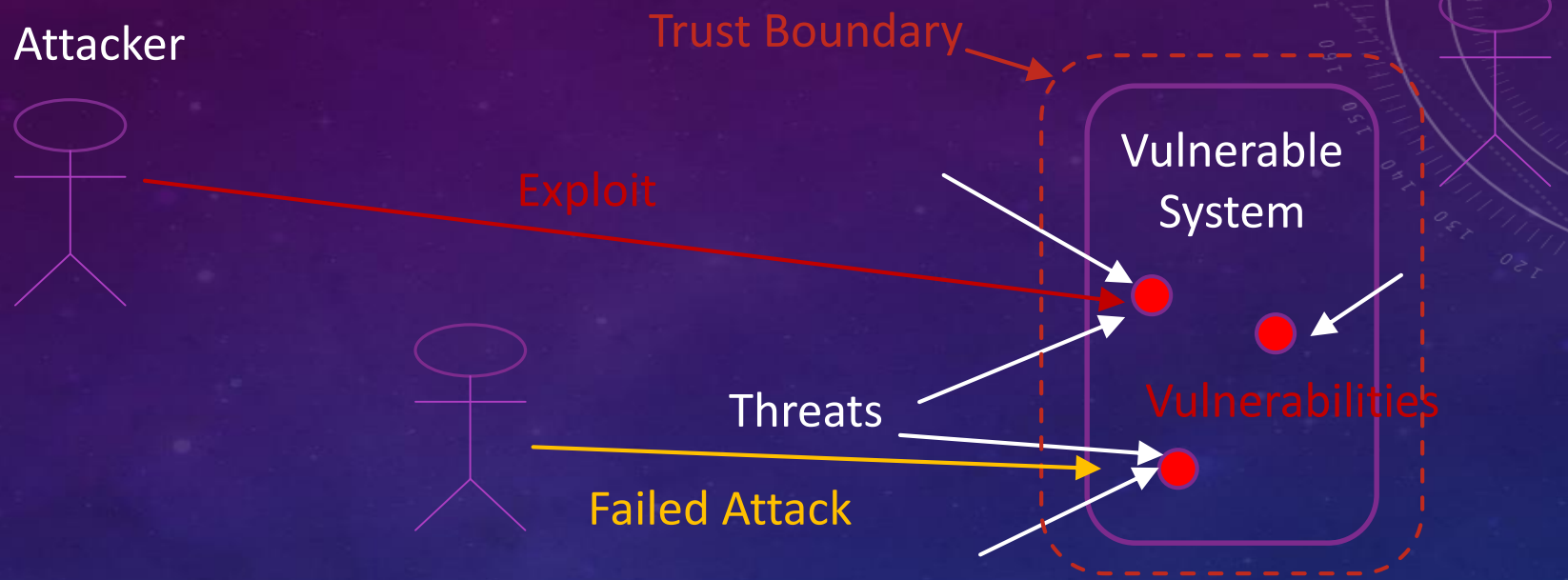
Is the information genuine, valid, and not fraudulent?

Usefulness

Is the information usable for its intended purpose?

http://en.wikipedia.org/wiki/Parkerian_Hexad

Definitions



- A vulnerability is a defect with security consequences
- A threat is a potential avenue of attack against the assets of a system
- An exploit is a successful attack procedure against a system
- Risk is the likelihood of potential damage from an exploit for a given threat
- All data and actions inside the Trust Boundary *should* be trustworthy

Vulnerabilities

- A defect that can result in the CIA properties of an application being violated
  - Design issue vulnerability
    - Unprotected sensitive data; no access controls
  - Implementation issue vulnerability
    - Unvalidated input data or unvalidated library
  - Deployment vulnerability
    - Unvalidated input data or unvalidated library
  - Process vulnerability
    - Backup data not encrypted

Threats

- Threats represent a potential violation of the CIA (confidentiality, integrity, availability) of one or more assets or components.
- Potentially {malicious, accidental, naturally occurring} “bad” thing
  disruptive events
- Are an expression of an impending danger or intention to damage
- A single vulnerability may be the source of multiple threats.
- A **threat agent** is an entity that causes or contributes to an in
  Attacker

Threats

- Design issue vulnerability examples
  - Failure to protect sensitive data
    - CC #’s revealed to support personnel; passwords stolen via a network monitor
  - Failure to implement access controls
    - User accesses OS files; Normal user elevates privileges to Admin user
- Implementation issue vulnerability examples
  - Failure to validate input data
    - SQL Injection threat; User able to delete critical file
  - Failure to check the validity of a library
    - Attacker can replace the library with own code
- Deployment vulnerability examples
  - Default accounts with default passwords
    - Documentation lists default passwords which can be acquired by an attacker
  - Configuration allows simple, short passwords
    - User passwords hacked and accounts compromised; passwords revealed by observing login

The Nature of Threats

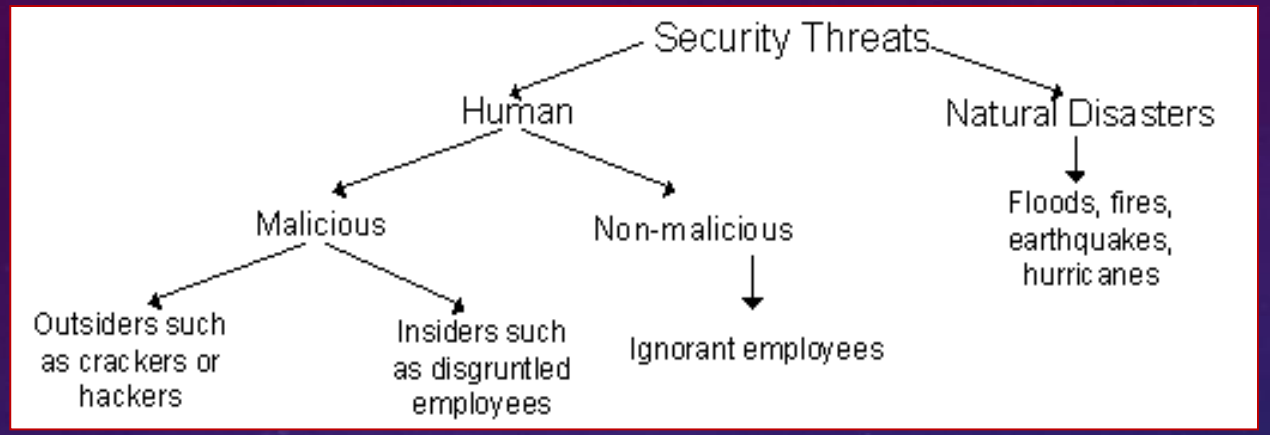


Image from *Security Threats*, http://technet.microsoft.com/en-us/library/cc723507.aspx

- Social threats: people are the primary threat vector
- Operational threats: failures of policy and procedure
- Technology threats: technical issues with the system
- Natural threats: from nature or environmental factors

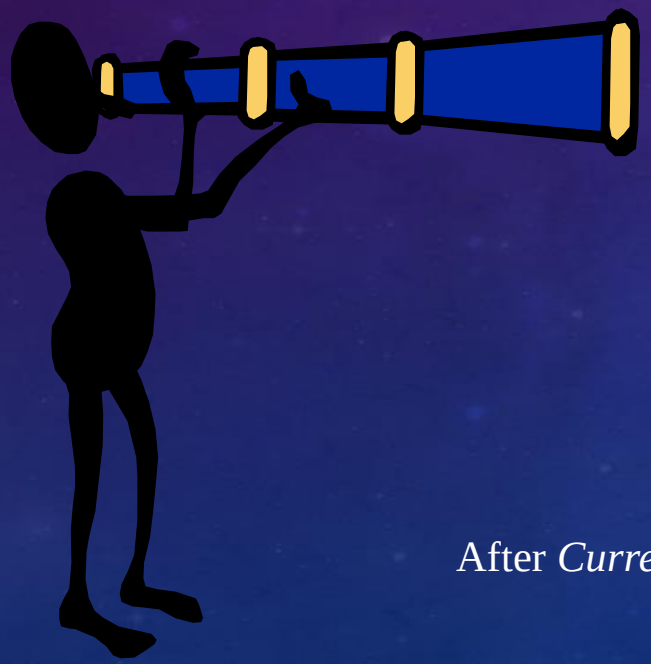
Threats to Confidentiality & Possession

Confidentiality

Is the information protected from unauthorized disclosure and observation?

Possession

Is the information in the control of the authorized individuals?



- Threats to the secrecy of information can come about due to the following:
  - Accessing assets
  - Disclosure
  - Observing / monitoring
  - Copying data

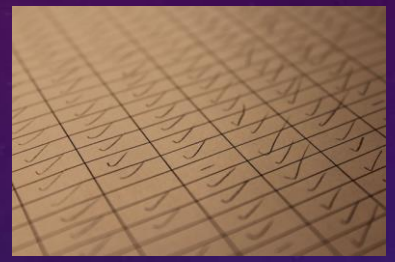
After *Current and Future Danger: A CSI Primer on Computer Crime*, p. 14

Threats to Confidentiality & Possession

- Non-production databases with live production data (dev and test environments)
- Use of weak authorization (passwords, default, backdoor accounts)
- Weak or mis-configured access control;
  - Granting excess privileges
  - Access to data unnecessarily
  - Adhere to Secure By Default principle
- Failure to encrypt data, backups and logs
- Theft via an exploit:
  - Of the software or operating system
  - Of the communication system
  - Of a third party

Can you think of a threat to confidentiality due to the use of cookies?

Threats to Integrity & Authenticity



Integrity

Is the information complete, whole and unchanged from the previous state?

Authenticity

Is the information genuine, valid, and not fraudulent?

- Ability to enter, use, or produce false data
- Modify, replace, or re-order data
- Misrepresent data
- Repudiation (disavowal – "I didn't do it")
- Misuse or failure to use data as required

After *Current and Future Danger: A CSI Primer on Computer Crime*, p. 14

Threats to Integrity & Authenticity

- Unauthorized modification of data
  - Data corruption due to power loss
  - Data corruption due to malicious or inadvertent damaging operations
  - Attacker creates forged data
  - Inadvertent overwrite of data with test data
  - Data is modified and the modification is removed from the logs

What would be required for the last item to occur? What would you do to prevent repudiation?

Threats to Availability & Usefulness

Availability

Are systems available so that they can be accessed in a timely manner for the intended purpose?

Usefulness

Is the information usable for its intended purpose?

- Destruction
- Damage
- Disruption
- Contamination
- Deny, prolong, or delay access

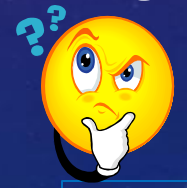
After *Current and Future Danger: A CSI Primer on Computer Crime*, p. 14

Threats to Availability & Usefulness

- Loss of data through invalid or malicious commands
- Denial of Service
  - Overloading and capacity issues
  - Hardware/Equipment/Facilities
    - Fire/flood/bombs
    - Theft of equipment
    - Power loss
    - Broken cables
    - Electronic interference and radiation
- Unable to decrypt encrypted data
  - Lost keys

Risk

- Risk measures the potential cost of a specific threat
  - What are the consequences of an exploit of a given vulnerability
- The cost includes:
  - The cost of remediating the vulnerability and deploying a patch
  - Costs associated with legal and compliance actions
  - Intangible costs, like the loss of reputation
- Risk is used to prioritize design, development, and testing effort



What is the risk if your software is publically reported to have a serious vulnerability?

Trust Boundary

- The Trust Boundary describes an imaginary border around a system
- Everything inside of the Trust Boundary should be known to be safe
- If a dataflow crosses the Trust Boundary from the outside, then it must be validated before it can be used