Web Security From an Auditor's Standpoint

What works, what doesn’t, what might.

IBWAS ‘10

Luis Grangeia

lgrangeia@sysvalue.com

About Me

- IT Security Auditor since 2001
  - First at SideStep, now at SysValue
- Working mostly with Telco, Finance and Government sectors
- Performed Web App Security Testing on:
  - Online banking
  - Stock brokerage
  - Online stores
  - Corporate Web sites
  - Internal Web apps

What will be covered today

- Prologue: Web Application Security
- Security Countermeasures, working or not?
  1. SQL Injection
  2. Cross Site Scripting
  3. Virtual Keyboards
- Welcome to the Social Web
  - Social Security, anyone? ☺

Prologue: Web Application Security

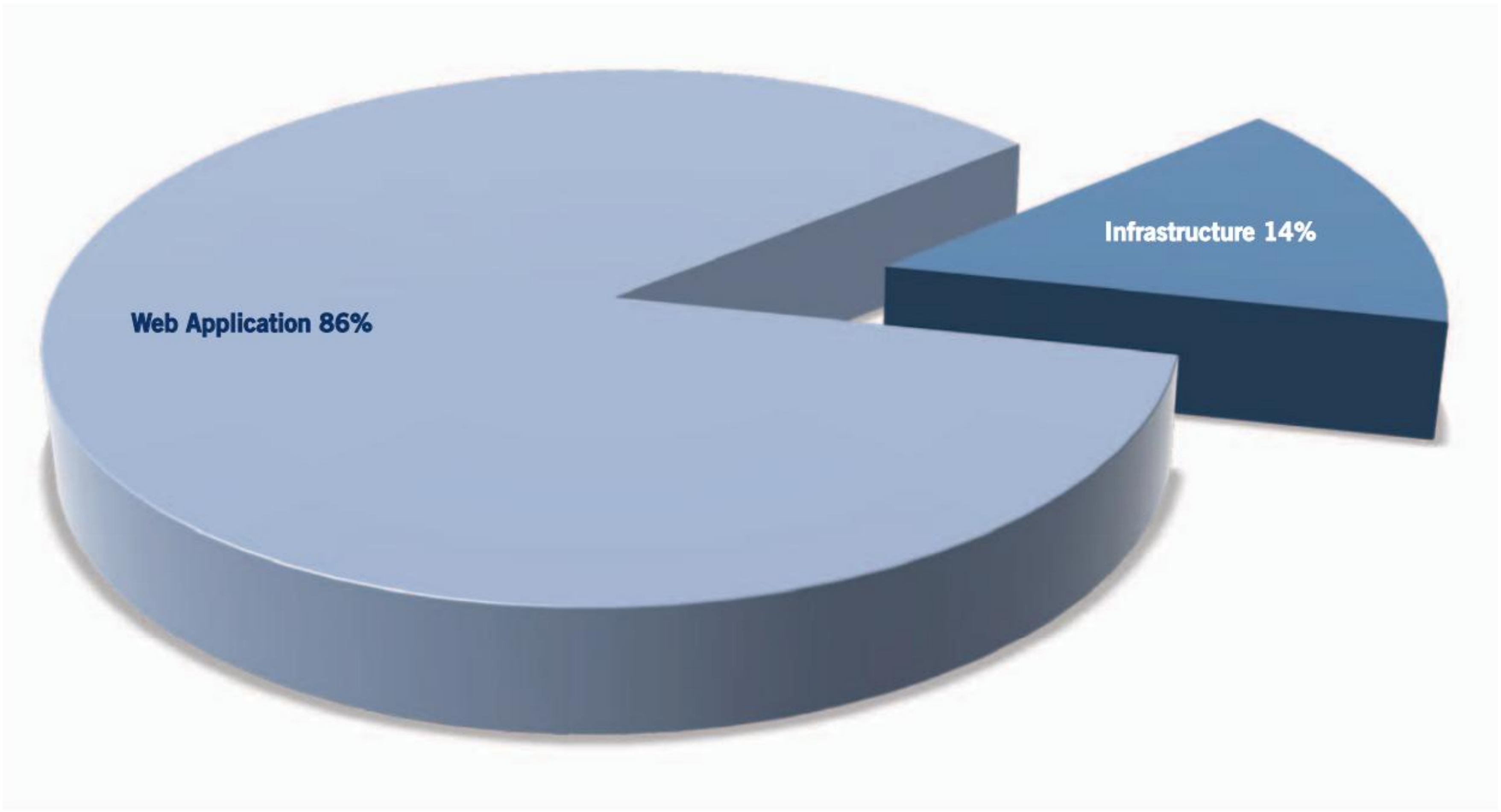
Why are we talking about it?

Applications vs Infrastructure

- Applications are the reason we go Online
- Network infrastructure is just a means to that end
- Infrastructure security is easy now (in comparison)

Infrastructure security is now almost fully commoditized

INFRASTRUCTURE VS APPLICATION



Areas of the compromised systems exploited.

The Web is Where it’s At

“The data used for this study shows that in 86% of all attacks, a weakness in a web interface was exploited.”

UK Security Breach Investigations Report 2010

An Analysis of Data Compromise Cases,

7Safe

Web Application Security is Hard

No turn key products

No easy solutions

Every Application is different

… That’s why it’s fun ☺

Security Countermeasures

What works and what doesn’t.

1. SQL Injection

We are winning the battle for eradication.

SQL Injection: Context

- Still leading the OWASP Top Ten

- Eight years ago it was **much worse**:
  - But the problem was not actively exploited
  - Not many knew how (good and bad guys alike)
  - There was **lower hanging fruit**

- Now it’s in the spotlight

SQL Injection: Analysis

- Hard to detect:
  - Good error handling prevents confident detection by automated tools
  - Web Application Firewalls mask the problem
    (attack pattern blacklisting)

- Easy to exploit:
  - Automated attacks are possible
    (SQL Injection worms)
  - Even Blind SQLi can be attacked easily
    (eg. SQL Power Injector)
  - This makes it a easier do make a point

SQL Injection: Countermeasures

Prevention | Mitigation is easy:

- Possible to easily distinguish between Data (variables) and Control (code):
    - **Use parameterized queries / prepared statements**
- New Web Development platforms are already performing Database Abstraction
    - Makes it a Platform problem, not a Developer's problem

SQL Injection: Conclusions

- Problems that are easy to exploit, are easy to understand by customers/developers
- Problems that are understood are more likely to get solved
- Especially when the solution is easy to understand and easy to implement

SQL Injection: Conclusions

We will eradicate the problem, because we have understood it completely and have a working simple solution that addresses the root cause

2. Cross Site Scripting

We risk losing this battle unless we address its root cause.

Cross Site Scripting: Context

- Second in the OWASP Top Ten

- Like SQL Injection, eight years ago it was **much worse**:
  - But the problem was not actively exploited
  - Not many knew how (good and bad guys alike)
  - There was **lower hanging fruit**

- Now it’s in the spotlight

Misconceptions abound

(not just around Web developers, also around some parts of the security community)

"Oh, it triggers an alert() box, is that it?"

"The user must click on a link in order to trigger the XSS attack."

"Sure, this attack might steal our customers sessions, but we use second-level authentication for financial transactions, so we're safe."

Cross Site Scripting: Risks

- Useful for realistic phishing attacks
- Cross site request forgery “boosts” the impact of XSS instances
- Web Applications are increasingly multi-domain:
  - Facebook “like” buttons
  - Google maps iframes
  - Google analytics
  - Advertising banners
  - Etc.
- A good exploitation means total control of all aspects of the communication between victim and Application.

Cross Site Scripting: Analysis

- (Somewhat) Easier to detect:
  - In fact, good Web App Scanners do it quite well, with low false positives and low false negatives
    - More complex instances are still hard to spot and may be missed
- Hard to exploit effectively:
  - While easy to trigger, good proof of concept code can be hard to write
  - The attack always depends of the “anatomy” of the Web App

Proving the Impact of XSS

- We're not doing our best
- We must prove to the developers the true impact of the problem
  - It's not easy, but it pays off!

Cross Site Scripting: Countermeasures

Prevention | Mitigation is hard:

- Also, it is a problem of distinguishing between Data (variables) and Control (code)
- **But there is no way to clearly make that distinction in HTML / Javascript**
- Input / Output sanitization is the only way, but it is **hard**:
  - Escaping output is dependent of document context:
    - HTML Content Elements, Common Attributes, Javascript Data Values, HTML Style Properties, etc.

Cross Site Scripting: Countermeasures (cont.)

- There is no possible way of addressing the root cause unless we change the protocol (won't happen):
  - Remember, the root cause isn't malicious input/output, it's mixing user supplied data with control code
- HTML5 won't really help:
  - Actually XSS will turn into a feature in HTML5 ☺
    - See HTML5's "Channel Messaging" and "Cross-document messaging"
- In fairness, there is one thing that might help against XSS in HTML5:
  - Sandboxed iframes

Cross Site Scripting: Countermeasures (cont.)

- The solution is complex, take it out of the hands of Developers:
  - Seriously, do you expect a developer to remember this?
    - http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet
  - Incorporate good, context aware output filtering in development platforms with template systems:
    - Google's "Automatic Context-Aware Escaping"
    - Ivan Ristić's "Canoe"
    - How long before .NET and Java adoption?
      - In the meantime, use Microsoft's Web Protection Library, AntiXSS
- Consider changes to the underlying protocol to promote better distinction of Code and Data (I won't go there, for now ☺)

Cross Site Scripting: Conclusions

- Security Problems that are:
  - Somewhat easy to detect, but…
  - Hard to prove their real impacts, and…
  - Have a difficult to execute and hard to understand solution…

…will be swept under the carpet!

And Cross site scripting falls squarely into this category.

Cross Site Scripting: Conclusion

This is a platform and protocol problem that is not well understood by developers.

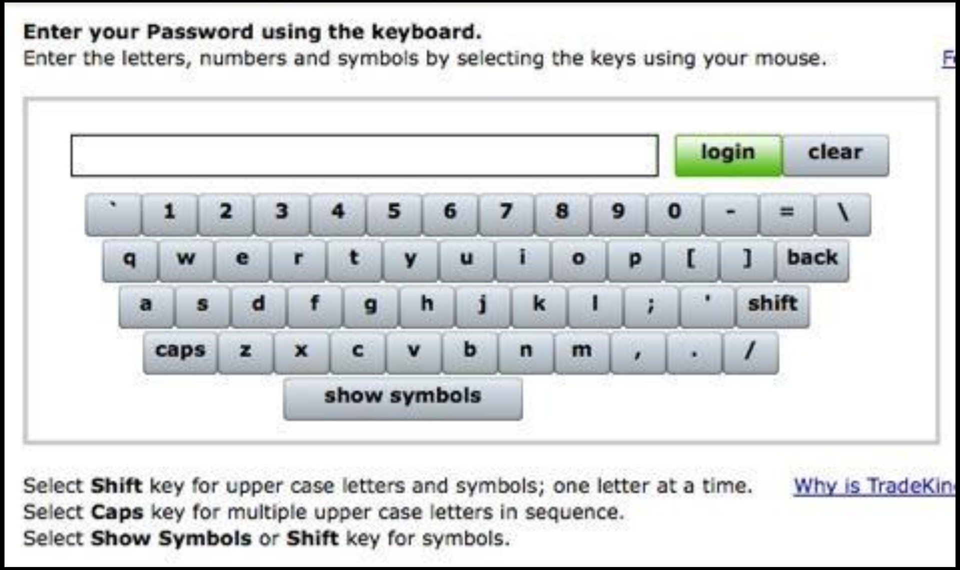
We are fighting a losing battle, unless we spend time developing platform-wide solutions, and at the same time, as auditors, make better proof-of-concept code, to drive research forward.

3. Virtual Keyboards

We're doing it wrong!

Virtual Keyboards: Context

- Introduced around summer 2003 in Portugal
- Implemented by most major online banks
- Response to a specific threat:
  - Malware that logged keystrokes in order to steal credentials



Enter your Password using the keyboard.
Enter the letters, numbers and symbols by selecting the keys using your mouse.

Select **Shift** key for upper case letters and symbols; one letter at a time.
Select **Caps** key for multiple upper case letters in sequence.
Select **Show Symbols** or **Shift** key for symbols.

Why is TradeKin

Virtual Keyboards are obsolete since day one.

Really. We can stop the madness.

Virtual Keyboards: Problems

- Mitigates only a specific instance of an attack:
  - “Key-logging malware” as opposed to “online credential stealing malware”
- Fails to look at the big picture
  - Result of bad/outdated threat assessment
- No “half-decent” malware uses key-logging anymore
  - In fact, key-logging results in a lot of useless data for the attacker

Virtual Keyboards: Threat not mitigated

- Form grabbing is the method of choice for grabbing credentials;
  - All credentials end up in a Web form
  - Some banks try to mask the POST'ed credentials using Javascript encryption
    - doesn't really work, the key is in the previous HTTP response

- Typical behavior (ex. W32/Qhost.JE):
  - Trojan injects a DLL into Internet Explorer
  - The DLL hooks HttpSendRequestA
  - The hook grabs POST data and uploads it to a FTP server

All of this works with or without SSL, with or without a virtual keyboard.

Virtual Keyboards

More of a problem than a solution:

- Fails to protect against current attack methods
- Gives a false sense of security
- Introduce new attack vectors:
  - Shoulder surfing
  - Induces the user to choose a weaker (easier to "type") password

Virtual Keyboards: Threats Mitigated
(for completeness sake)

To be fair, virtual keyboards protect against this:

- Wireless keyboard sniffing
  - Use encryption, corded keyboards

- Hardware key-loggers
  - Look at your hardware ports, don't leave computer unattended in insecure locations

Virtual Keyboards: Conclusion

Don’t be afraid to remove a security countermeasure that doesn’t work.

Educate the public so that this “security theater” is exposed.

The Social Web Revolution

Social Security, anyone?

facebook
December 2010

The Social Revolution – Facebook

- Facebook connects over 500 million users
- That's around 30% of all Internet users worldwide

It's more than 7% of the World's Population (est.)

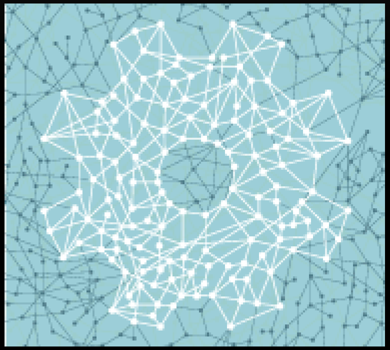
Before talking infosec…

- Before talking about possible security uses for the social graph, we must be able to decentralize it
- Too much power is concentrated on a single organization
- Either use Facebook's graph alongside other tools, or build a distributed social graph altogether:
  - Possible and doable: check the Diaspora Project

How can security benefit from the social graph

Facebook is already showing us how:

- Facebook Messages:
  - Spam filtering by looking at your friend connections
  - Has the potential to be virtually foolproof mitigating spam.
- There are still possible bumps down the road, but the idea has potential



Using the Social Graph to Improve SSL

Warning: crazy idea ahead.

SSL is OK, but Already Shows Age

Two roles of SSL/TLS:

- Authenticating the endpoints
- Providing transport integrity and confidentiality through encryption

- The authentication is based on the trust of organizations that are organized in a hierarchy, originating in root CA’s that are “imposed” on us

The confidence issue

"Directly or indirectly, there are more than 650 different organizations that function as trusted CA's for either Internet Explorer or Firefox"

(source: EFF SSL Observatory)

All you need is to break the security of one to break the SSL model…

SSL: What are the threats

- There are reports stating that MiTM attacks against SSL are being performed using certificates generated by rogue CAs:
  - By governments (espionage)
  - By Law Enforcement

- *"Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL"*
  - Christopher Soghoian and Sid Stammy, April 2010

- *"Law Enforcement Appliance Subverts SSL"*
  - Wired Threat Level, March 24, 2010

Challenge:
Defeating Local MiTM Attacks

"You can fool some of the people all of the time, and all of the people some of the time, but you cannot fool all of the people all of the time."

Abraham Lincoln

Defeating Local MiTM Attacks by "Friendsourcing"

- Assuming no malicious entity is omnipotent on the Internet (not even governments):
    - SSL MiTM attacks will always be localized (to an ISP or a specific country)

- So, in order to validate the authenticity of a Web site, one can request that an online friend (or a random group, or all of them) visits the same site and checks the certificate for us.
    - In geographically distinct locations
    - Using different computers and ISP's

- We only authenticate if all (or most) our friends data matches up.

Defeating Local MiTM Attacks by "Friendsourcing"

- The devil is in the details…
  - Just an idea for discussion, for now
  - Assumes we can securely authenticate friends and pass private messages along the social graph
  - Can be complemented by historic data:
    - "the certificate has changed even though the previous certificate had an expiry date well in the future"
- The general idea is to create the possibility of a custom security model for authentication based on a Web of Trust, and not on a hierarchy of "Trusted Authorities"
- There are endless opportunities for the social graph…

Thank You

Q&A

Luis Grangeia, SysValue

lgrangeia@sysvalue.com