

DEFACED

**An insight into methodologies, tools and
motivations**

Adam “@feabell” Bell – Security Consultant

Event – OWASP Day 2015

Date – 27th February 2015

Company Overview

Company

- Lateral Security (IT) Services Limited
- Founded in April 2008 by Nick von Dadelzen and Ratu Mason (Both Directors)
- Auckland, Wellington Melbourne: ~20 highly specialised security consultants

Services

- Security testing (design & architecture, penetration testing, configuration, code reviews, security devices & controls, mobile apps)
- Security advisory (Lifecycle compliance & audit – ISO, PCI-DSS, NZISM, policy process development, threat modeling and risk assessment)
- Regular ongoing technical testing and assurance programs



Who's This Guy?



It's a me, MARIO!



**I didn't read the abstract,
what's all this?**

"The internet is a hostile place, particularly if you are a charitable organisation.

Websites are compromised and defaced on a daily basis and sometimes their owners need a little help getting to the bottom of what happened.

So what if your website has been compromised and is getting reports of abuse... How do you react? What can you do?

This talk will cover a recent incident response and investigation I carried out and describe how you, as the developer-in-charge of an all-in-one webhost can investigate and respond.

Based on a true story, this talk will describe the incident response methodology I used, take a look at some of the tools that the defacer had left behind and give an insight into the badguys mindset and motives."



While this is a Lateral Security Presentation, this was an IR carried out on behalf of myself

As such, any incompetence is on me, not Lateral.

Names changed to protect the innocent



Dev <> Ops distance is closing

Your code doesn't stand in isolation, it has to interact with tens of other tools and API's.

Once your code leaves your hands, it's often not even yours anymore. And will slowly become obsolete as it sits on a client disk somewhere

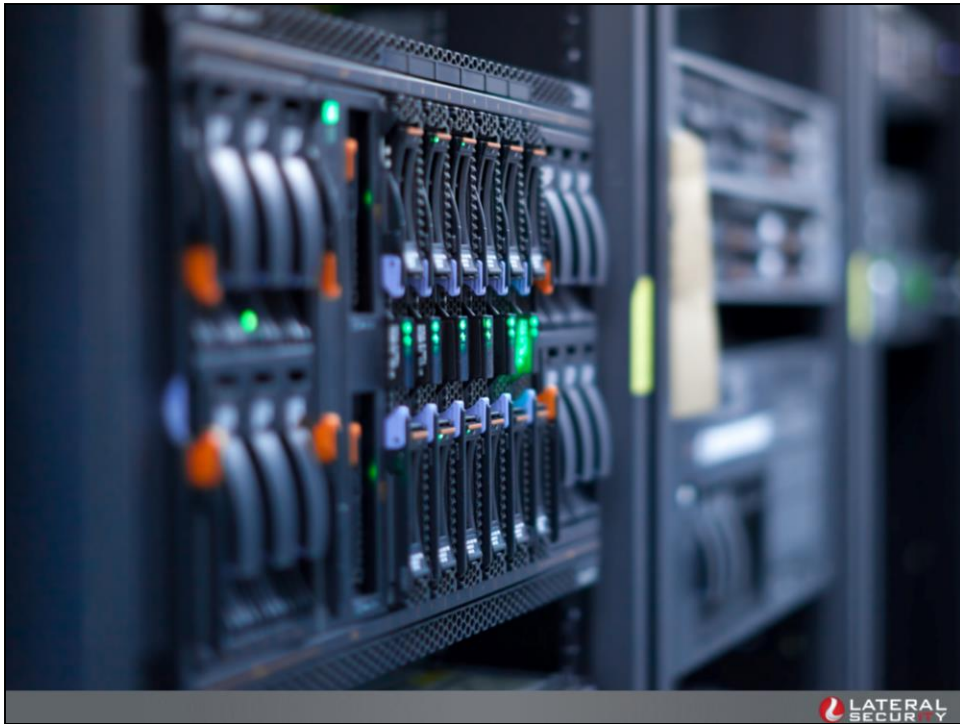


Story Time



Our hero protagonists.

The BAHLMAN and Robell.



Our victim



Our antagonist

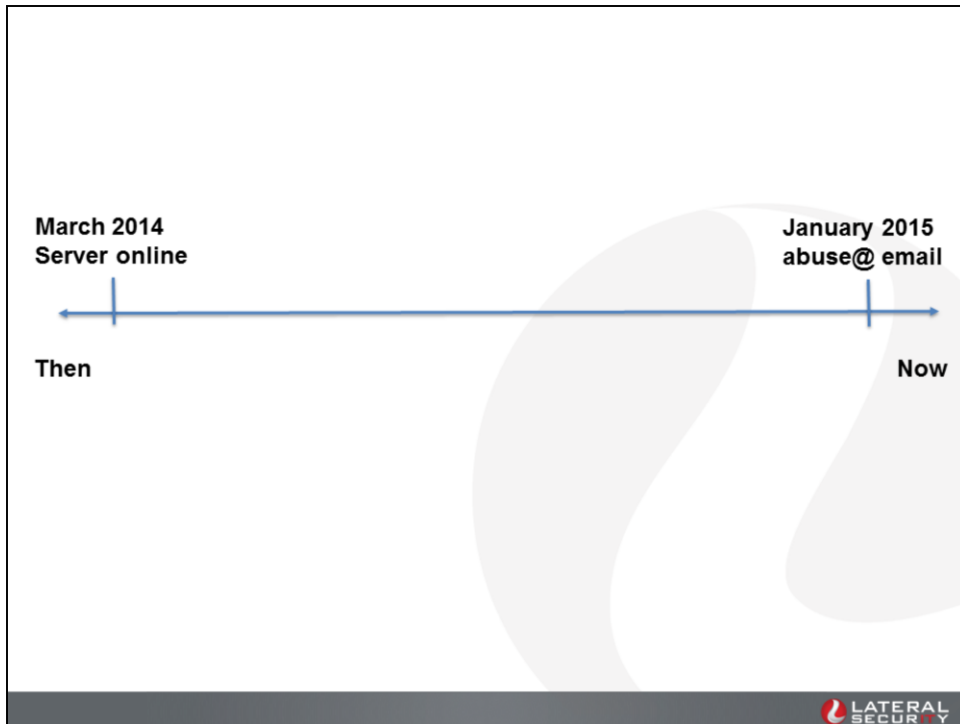


The Bahlman was approached by a third party

Third party is a charity organisation that was critical of their own government.

Third party had previously been compromised, and wanted a secure platform going forward.

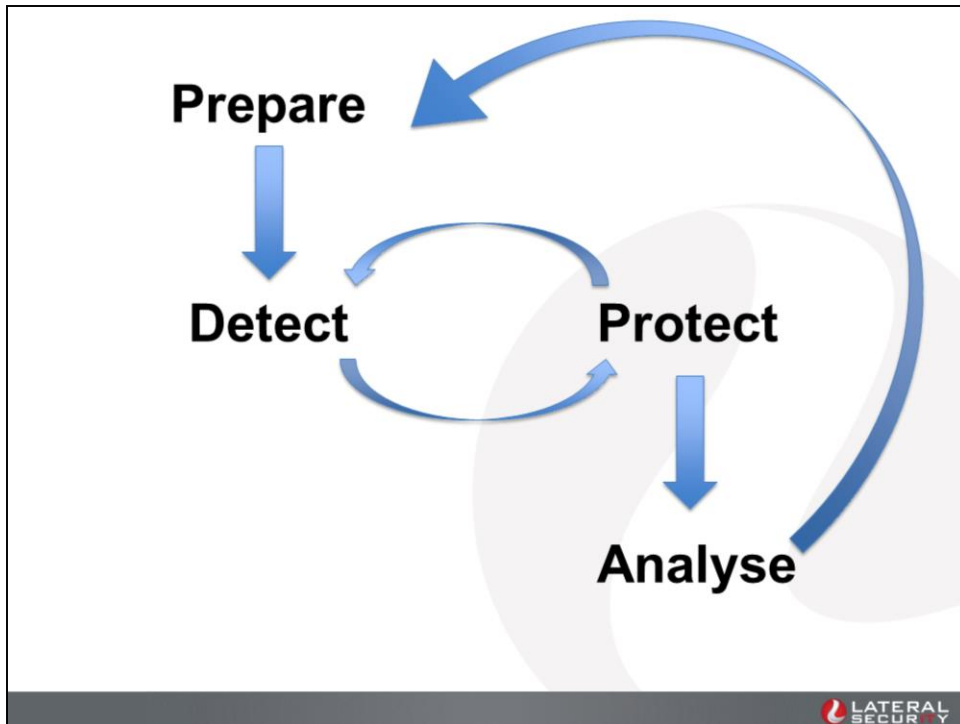
The Bahlman and Robell LEAPED into action, with a new server, migrating their ageing WP install to latest. Pruning old addons and frankly making it idiot proof



In January 2015, our technical contact at the charity forwarded an email to us from a third party who had emailed their abuse@ address.

This third party claimed that our IP had been detected performing malicious attacks against wordpress instances that they controlled. This site looked sketchy as HELL

(and had an unrelated SQL injection vulnerability)



So, we sceptically started the incident response process.

I mean, our server was perfect... right?



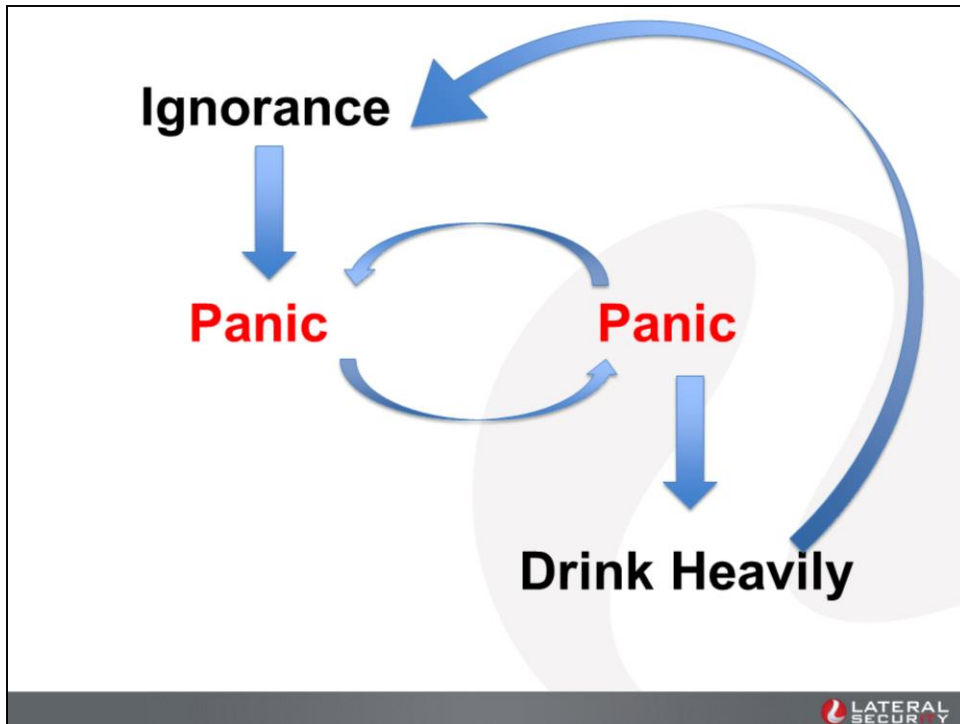
But we found nothing.

There were no malicious users added, no signs of a system intrusion, no (detectable) rootkits. The system seemed secure.

```
feabell@old-b: /var/www/html/ $ ls -la
-rw-r--r-- 1 www-data www-data 62K Jan 14 15:46 403.php
-rw-r--r-- 1 www-data www-data 62K Jan 14 15:21 404.php
-rw-r--r-- 1 www-data www-data 140K Jan 14 16:12 500.php
-rw-rw-r-- 1 www-data www-data 418 Dec 2 00:04 index.php
-rw-r--r-- 1 www-data www-data 69K Sep 3 21:51 Salem.php
-rwxrwxr-x 1 www-data www-data 19 Aug 24 2010 test.php
-rw-rw-r-- 1 www-data www-data 4.9K Dec 2 00:04 wp-activate.php
-rwxrwxr-x 1 www-data www-data 226 Sep 22 2011 wp-atom.php
-rw-rw-r-- 1 www-data www-data 271 Mar 20 2014 wp-blog-header.php
-rw-rw-r-- 1 www-data www-data 4.9K Jan 3 22:59 wp-comments-post.php
-rwxrwxr-x 1 www-data www-data 244 Sep 22 2011 wp-commentsrss2.php
-rwxrwxr-x 1 www-data www-data 3.0K Jan 15 19:25 wp-config.php
-rw-rw-r-- 1 www-data www-data 2.7K Jan 3 22:59 wp-config-sample.php
-rw-rw-r-- 1 www-data www-data 2.9K Dec 2 00:04 wp-cron.php
-rwxrwxr-x 1 www-data www-data 246 Sep 22 2011 wp-feed.php
-rw-rw-r-- 1 www-data www-data 2.4K Mar 20 2014 wp-links-opml.php
-rw-rw-r-- 1 www-data www-data 2.7K Dec 2 00:04 wp-load.php
-rw-rw-r-- 1 www-data www-data 33K Jan 3 22:59 wp-login.php
-rw-rw-r-- 1 www-data www-data 8.1K Dec 2 00:04 wp-mail.php
-rwxrwxr-x 1 www-data www-data 413 Jan 5 2012 wp-pass.php
-rwxrwxr-x 1 www-data www-data 224 Sep 22 2011 wp-rdf.php
-rwxrwxr-x 1 www-data www-data 334 Jan 5 2012 wp-register.php
-rwxrwxr-x 1 www-data www-data 226 Sep 22 2011 wp-rss2.php
-rwxrwxr-x 1 www-data www-data 224 Sep 22 2011 wp-rss.php
-rw-rw-r-- 1 www-data www-data 11K Dec 2 00:04 wp-settings.php
-rw-rw-r-- 1 www-data www-data 25K Jan 3 22:59 wp-signup.php
-rw-rw-r-- 1 www-data www-data 4.0K Jan 3 22:59 wp-trackback.php
-rw-rw-r-- 1 www-data www-data 3.0K Dec 2 00:04 xmlrpc.php
feabell@old-b: /var/www/html/ $
```

403.Php, 404.php (ignore) 500.php are not the real files for these error messages. There are dynamically generated by apache as we have not specified custom error handlers.

Also Salem.php is just obviously malicious.



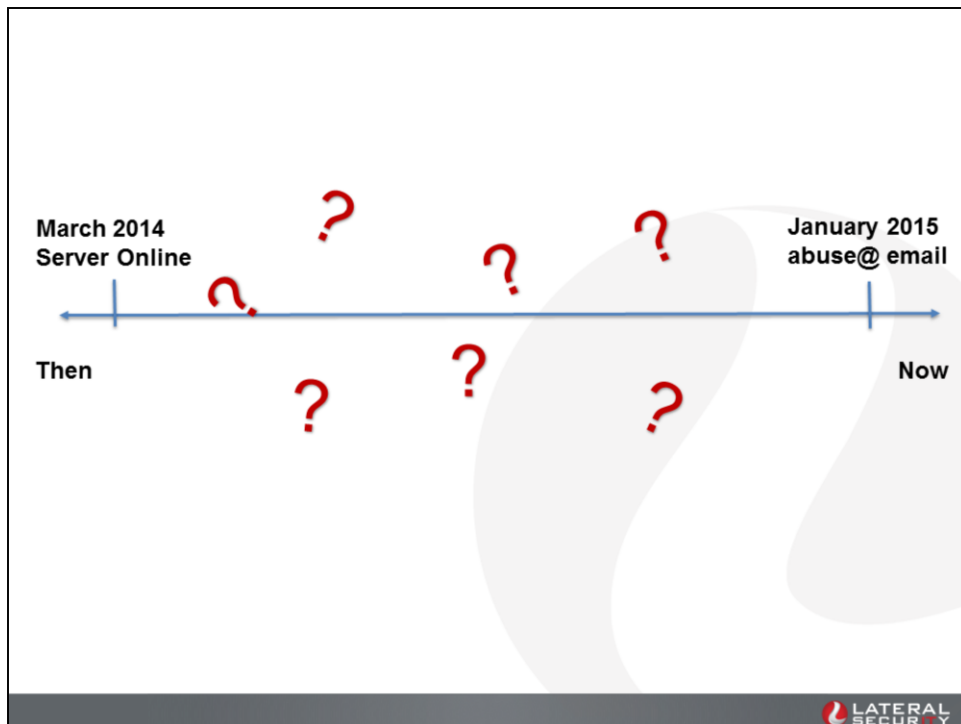
While the previous incident response lifecycle may be unfamiliar to you, I am happy to present one that may be more recognisable



We knew the host was compromised, so real incident response could begin. The first step?



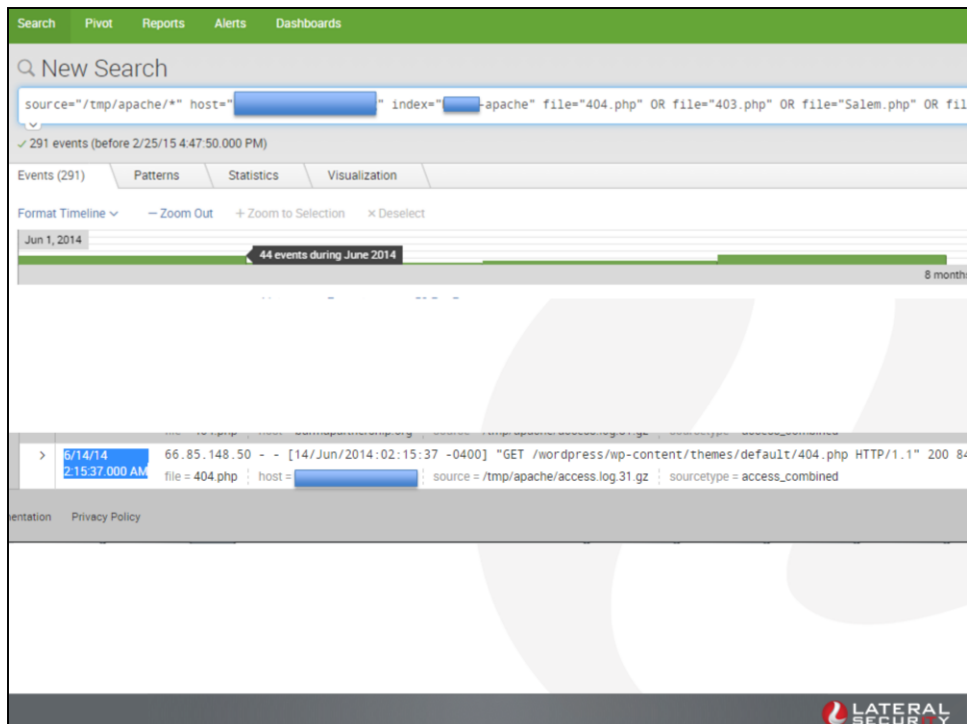
The Bahlman delivered the customary fee for “free” incident response



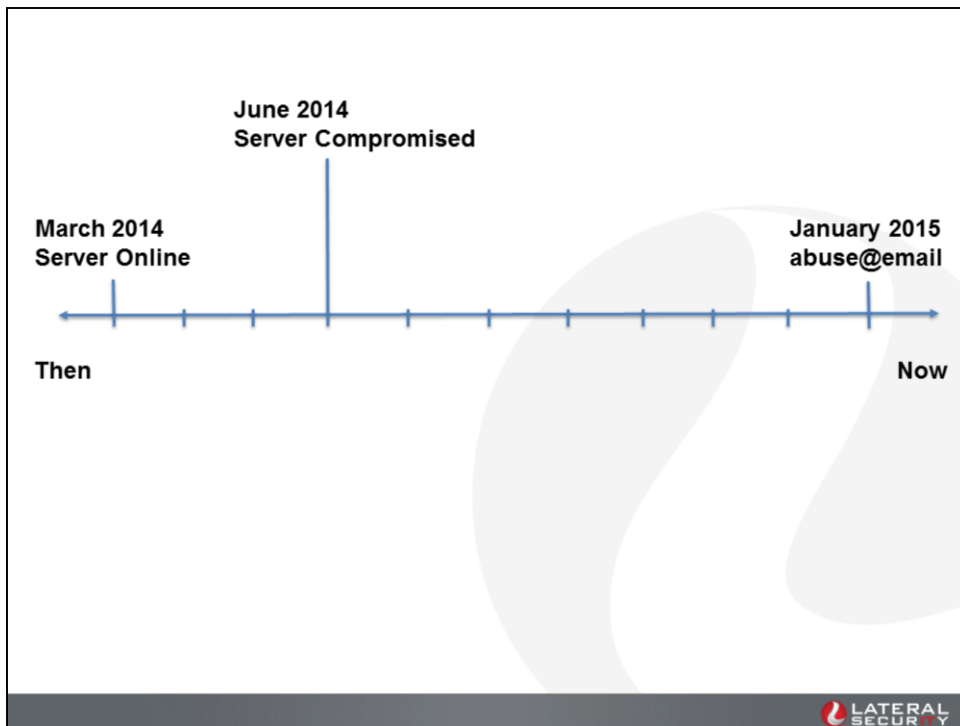
So, where to start? Lets fix our timeline



So lets grab all of our log files and throw them into a log aggregation tool. I like Splunk



So how long has the box been compromised?



So that's not good.

3 months online before compromise. 7 months of uninterrupted compromise.
facepalm





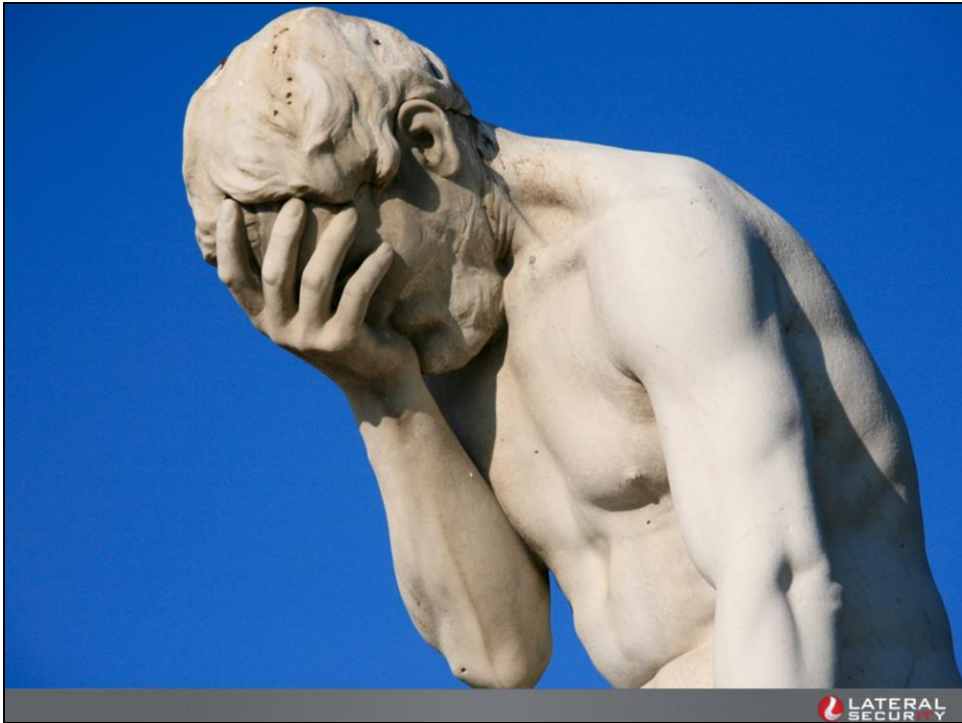
So how did this even happen?

This is the first request for the 'badguy' file. To /wordpress/foo/bar/whatsit.

But /wordpress ISN'T THE ROOT DIRECTORY FOR THIS SITE. This is a DIFFERENT wordpress install. WHAT THE ACTUAL HECK.

So, where did this WordPress come from?

- Charity had previously had “sub” sites on their old WP install, that they had deemed ‘no longer needed’
- Charity had decided they were needed after all
- Dropped the old WP install folder into /wordpress
- Noticed that things didn’t work, because the backend database no longer existed.



How Did Those Files Get In The WebRoot?

- We can see that before they were in the webroot, they were in a WP theme.

```
400] "GET /wordpress/wp-content/themes/default/404.php HTTP/1.1" 200 778 "http://[redacted]
source = /tmp/apache/access.log.31.gz | sourcetype = access_combined
```

- And it appears that this client IP logged into the WP admin interface to put them there

```
AM 66.85.148.50 - - [14/Jun/2014:02:15:42 -0400] "POST /wordpress/wp-content/themes/default/404.php HTTP/1.1" 200 778 "http://[redacted]
default/404.php" "Mozilla/5.0 (X11; Linux x86_64; rv:17.0) Gecko/20121201 icecat/17.0.1"
file = 404.php | host = [redacted] | source = /tmp/apache/access.log.31.gz | sourcetype = access_combined

AM 66.85.148.50 - - [14/Jun/2014:02:15:24 -0400] "POST /wordpress/wp-admin/theme-editor.php HTTP/1.1" 302 540 "http://[redacted]
file=/themes/default/404.php&theme=WordPress-Default" "Mozilla/5.0 (X11; Linux x86_64; rv:17.0) Gecko/20121201 icecat/17.0.1"
file = theme-editor.php | host = [redacted] | source = /tmp/apache/access.log.31.gz | sourcetype = access_combined

AM 66.85.148.50 - - [14/Jun/2014:02:14:36 -0400] "POST /wordpress/wp-login.php HTTP/1.1" 302 1053 "http://[redacted]wordpress
.org%2Fwordpress%2Fwp-admin%2F" "Mozilla/5.0 (X11; Linux x86_64; rv:17.0) Gecko/20121201 icecat/17.0.1"
file = wp-login.php | host = [redacted] | source = /tmp/apache/access.log.31.gz | sourcetype = access_combined
```

How Were Admin Creds Obtained?

- We see a different third party hits the WP install script for the old WP install
- They then login to the admin console
- And upload a DIFFERENT backdoor

```
176.194.78.55 - - [12/May/2014:12:04:13 -0400] "POST /wordpress/wp-content/plugins/doc/doc.php HTTP/1.1" 200 4713 "http://[redacted]
/doc/doc.php" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.132 Safari/537.36 OPR/2
file = doc.php : host = [redacted] : source = /tmp/apache/access.log.35.gz : sourcetype = access_combined

176.194.78.55 - - [12/May/2014:11:54:47 -0400] "POST /wordpress/wp-admin/update.php?action=upload-plugin HTTP/1.1" 200 3844 "http
/plugin-install.php?tab=upload" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.132 S
file = update.php : host = [redacted] : source = /tmp/apache/access.log.35.gz : sourcetype = access_combined

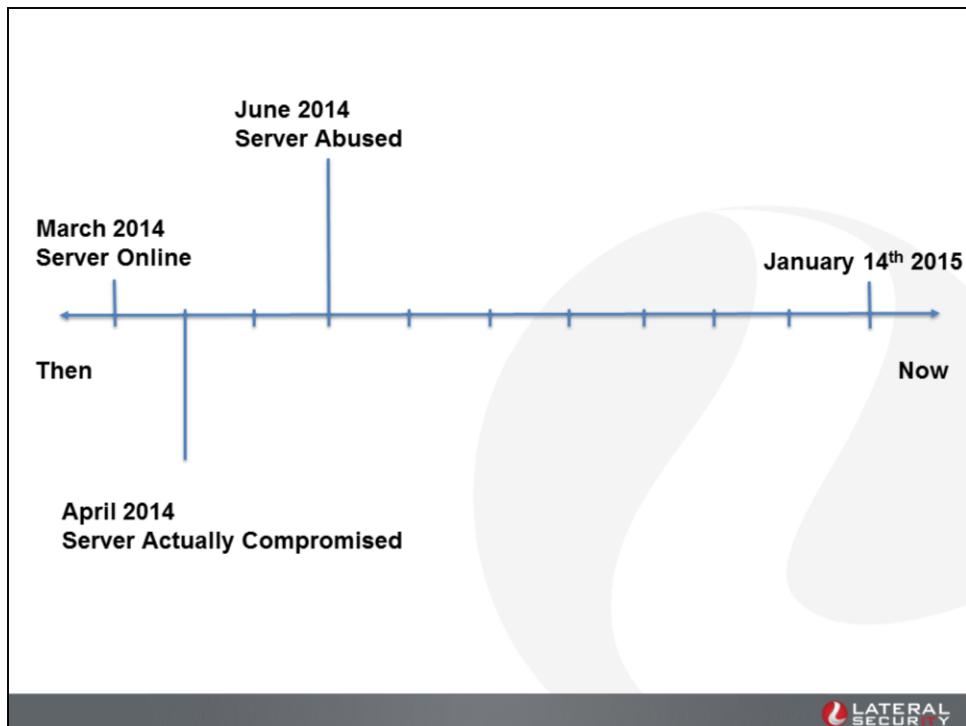
176.194.78.55 - - [12/May/2014:11:52:52 -0400] "POST /wordpress/wp-login.php HTTP/1.1" 302 1047 "http://[redacted]g/wordp
NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.132 Safari/537.36 OPR/21.0.1432.57"
file = wp-login.php : host = [redacted] : source = /tmp/apache/access.log.35.gz : sourcetype = access_combined

176.194.78.55 - - [12/May/2014:11:51:24 -0400] "POST /wordpress/wp-login.php HTTP/1.1" 200 1492 "http://[redacted]g/wordp
2Fwordpress%2Fwp-admin%2F" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
432.57"
file = wp-login.php : host = [redacted] : source = /tmp/apache/access.log.35.gz : sourcetype = access_combined

176.194.78.55 - - [30/Apr/2014:21:52:36 -0400] "POST /wordpress/wp-admin/install.php?step=2 HTTP/1.1" 200 1422 "-" "-"
file = install.php : host = [redacted] : source = /tmp/apache/access.log.37.gz : sourcetype = access_combined

176.194.78.55 - - [30/Apr/2014:21:51:46 -0400] "POST /wordpress/wp-admin/install.php?step=2 HTTP/1.1" 200 1808 "-" "-"
file = install.php : host = [redacted] : source = /tmp/apache/access.log.37.gz : sourcetype = access_combined
```

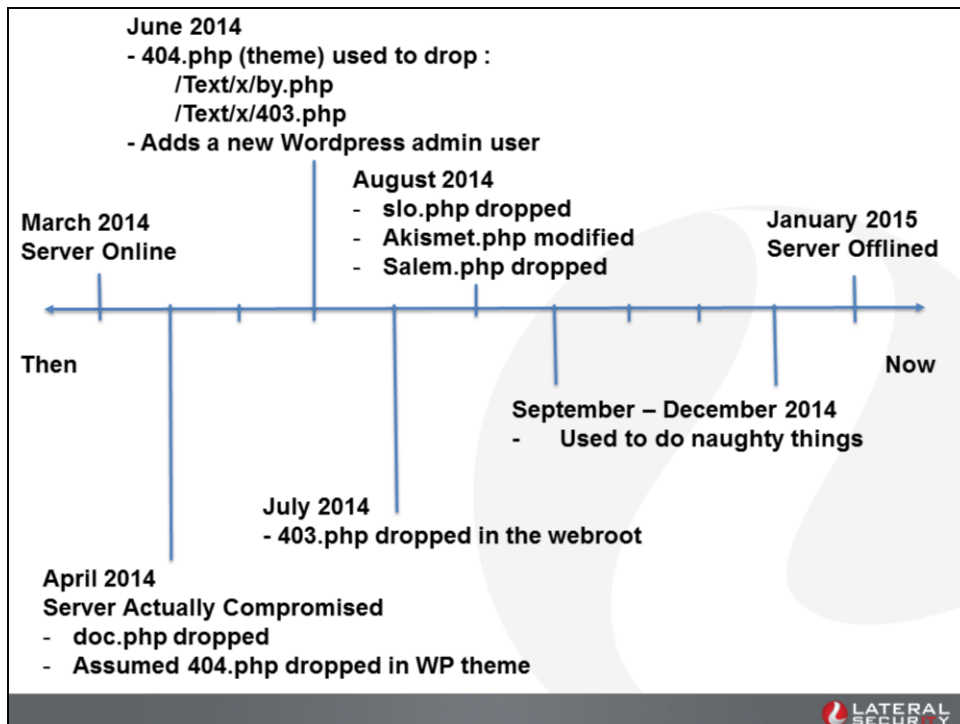
SECURITY



So that's not good.

3 months online before compromise. 7 months of uninterrupted compromise.
facepalm

**No facepalm.gif suitably
demonstrates the
magnitude of facepalming
required.**



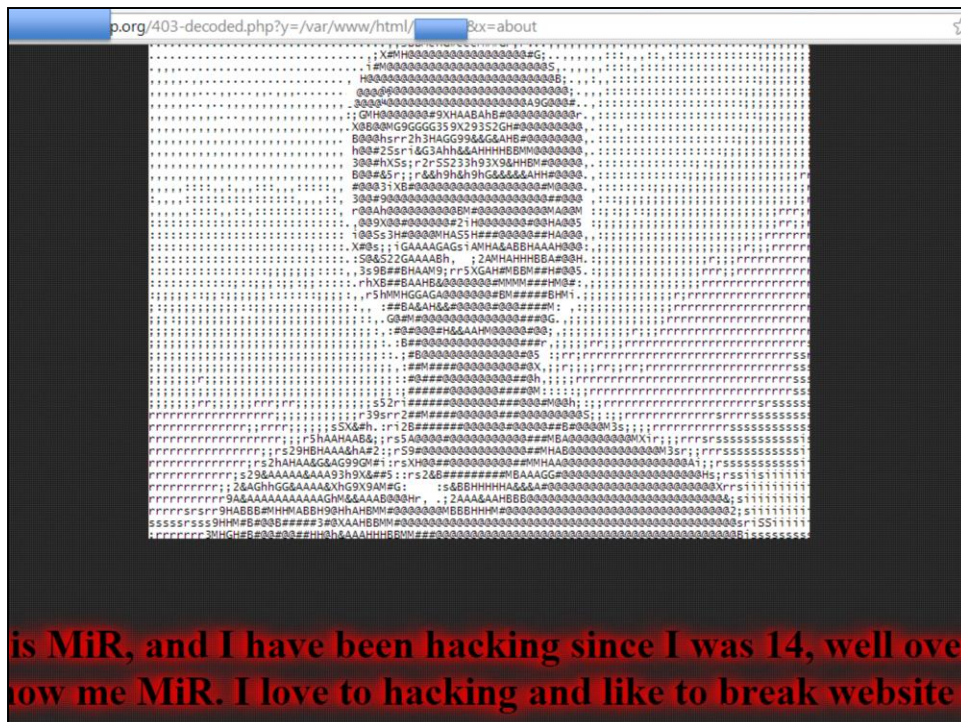


1337mir is a well know website defacer, with a LOT of defaces. In the tens of thousands on zone-h

Rather than perform a deface, they just posted a MESSAGE to zone-h in a non-indexable file.

1337mirs website claims that he is a sub-17 year old guy “interested in security”.

Was this a charitable act to a company that is a charity?





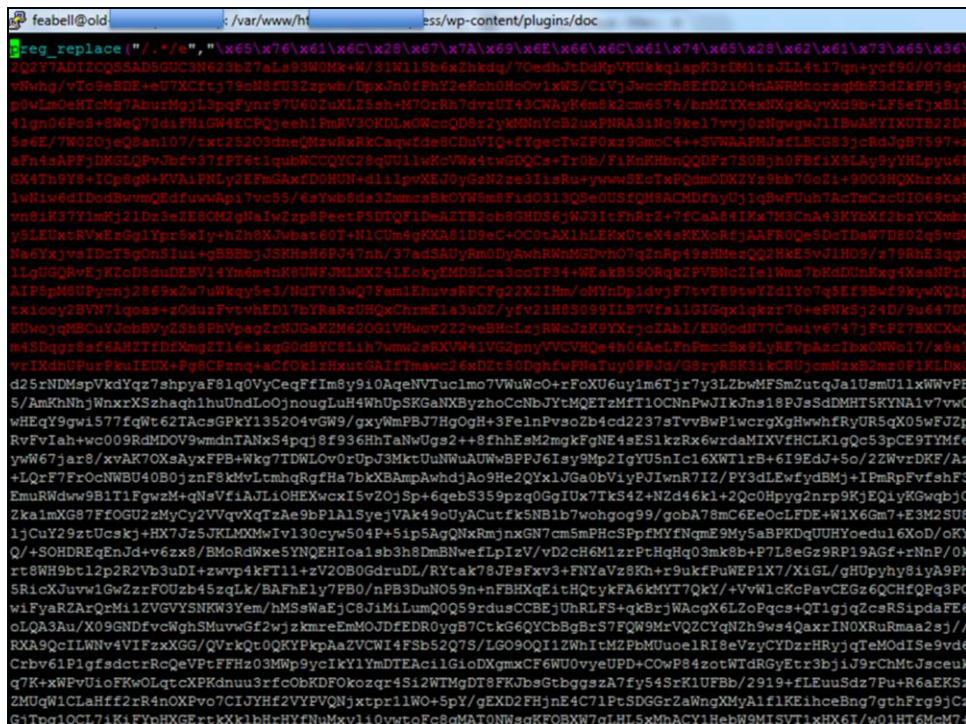
And this is where our actual evil hacker comes in.

1337mir wasn't evil, he didn't use our server for badness. But this third party did.

We don't know what our server was being made to do, apart from that it was attacking other WP hosts. I'd guess either scanning for vulnerable components or bruteforcing login creds.



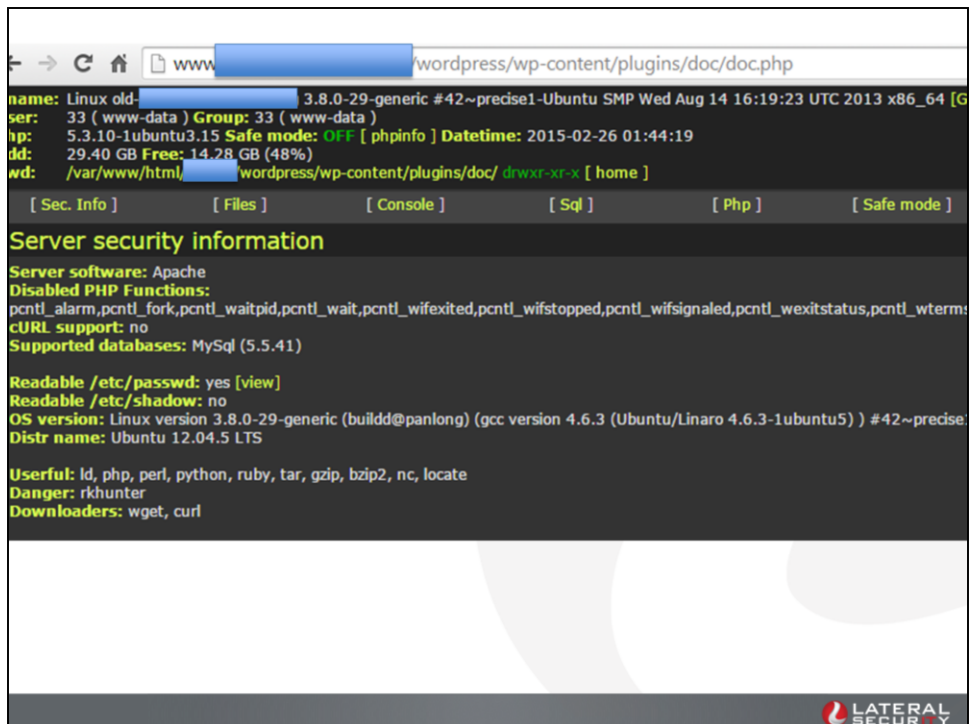
Tool time!



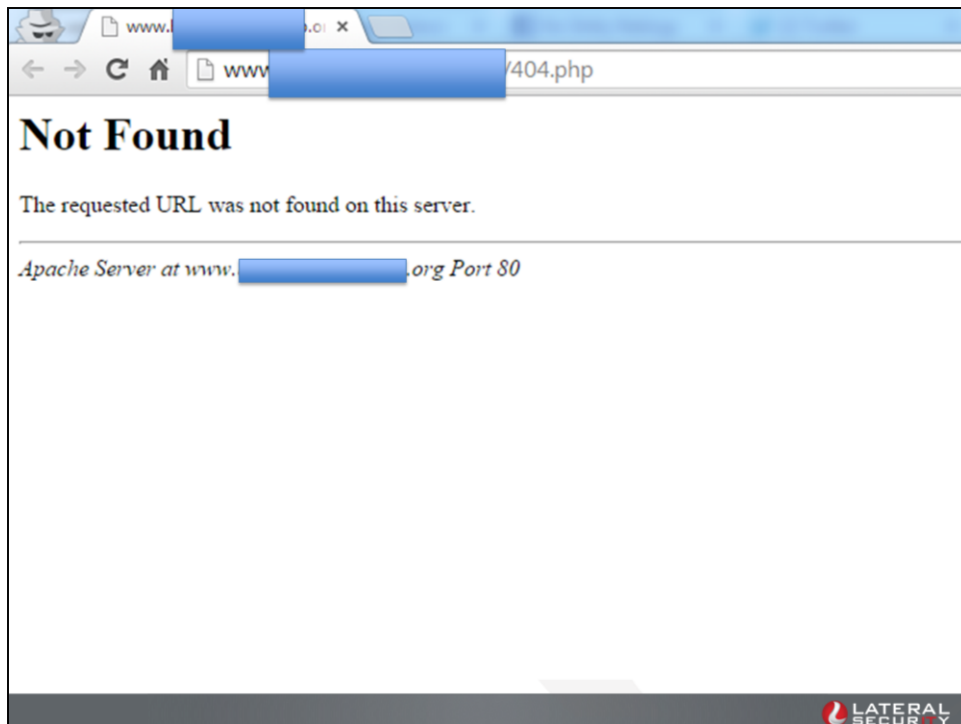
A quick note on obfuscation

This is how most of the dropped files look when opened in the text editor of your choice (which is Vim, naturally).

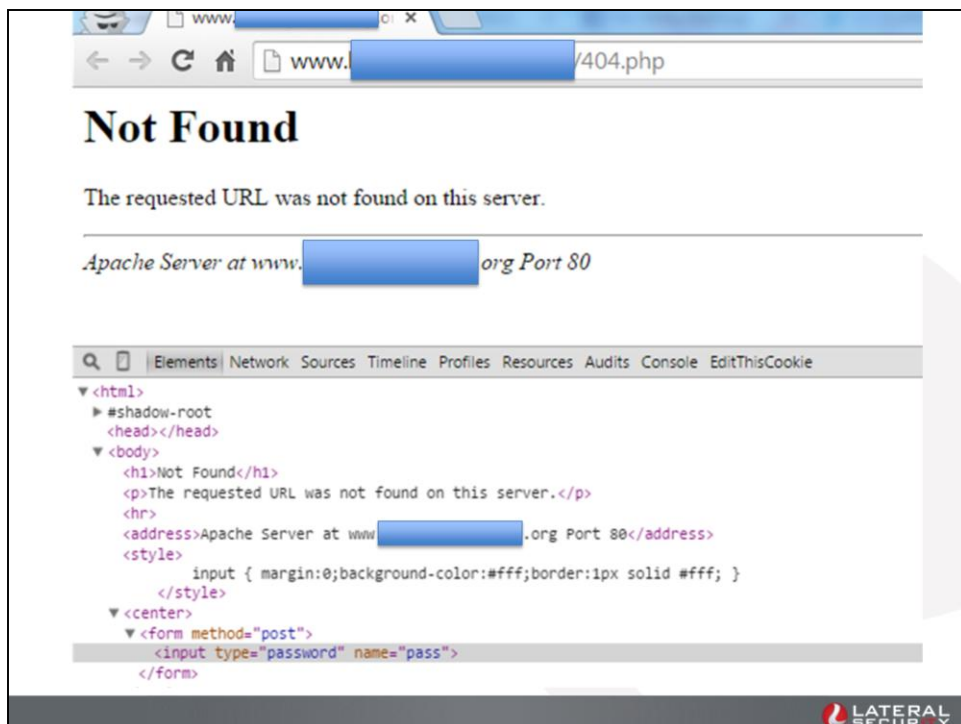
*eval(gzinflate(base64_decode('massiveblobof
base64goeshereImeanseriouslyhowlongisthis
bigchunkofbase64?It'ssolongthatit'ssevenlinewr
appedonthe presentationslide.It'saseriouslylarg
e amount of text.GztoDavidonhisnewarrival!Is the
rereallymoreneedtopadthisout?CanIkeeptypin
gandfillthisslide?I'mprettysureIcan.#Fobsthoug
htleadertrainingatOWASP2016#dreamisalive')*



Pretty basic tool, no privsecs to root or anything... but allows you to do what you want.



Just a 404, right?



Secret login on 404 page

About	Explore	Upload	Mass	WP Mass	String	Shell	Eval	MySQL
PHP	NetSploit	BruteForce	Readable Dir	DDos	LocalDomain	WP Reset		
Joomla Reset	pass changer	VB Changer	wp index changer	Joomla index changer				
Zone-H	Symlink	Mail	CMS Scanner	Bypass	Website Whois	Port-Scanner		
			Hash Analyzer	Encode/Decode				



So if these tools are all so amazing, why can't we just google some unique strings and find all the compromised websites?

Because every tool, as it's first line says "if you are in this list of useragents, i'm not showing you a thing"

What Went Well

- **System Hardening**
- **Not allowing client root access**
- **Logrotate & logfile histories**
- **IR process**
- **A somewhat benevolent hacker?**
- **Poor opsec by the attackers**

What Didn't Go So Well

- **Allowing client any kind of access at all**
- **Active log monitoring**
- **File integrity monitoring**
- **Client communications and siloing**

What Can We Do To Prevent This?

- **Share YOUR stories of compromise**
- **Understand the software stack**
- **Get cosy with your Ops team**
- **Don't assume**

Questions and Contacts



Presentation Download
[www.lateralsecurity.com/
presentations](http://www.lateralsecurity.com/presentations)

Lateral Security (IT) Services Limited

Wellington

38-42 Waring Taylor Street (level 7, Petherick Tower)
PO Box 8093, Wellington 6143, New Zealand
Phone: +64 4 4999 756
Email: sas@lateralsecurity.com

Auckland

187 Queen Street (level 8, Landmark House)
PO Box 7706, Auckland, New Zealand
Phone: +64 9 3770 700
Email: sas@lateralsecurity.com

Melbourne

200 Queen Street (level 13)
Melbourne, VIC 3000, Australia
Phone: +61 1300 554745
Email: sas@lateralsecurity.com

