An analysis of exploitation behaviors on the web and the role of web hosting providers in detecting them

Davide Canali, Davide Balzarotti

Aurélien Francillon

Software and System Security Group

EURECOM, France

http://s3.eurecom.fr/

NDSS 2013 & WWW 2013

Behind the Scenes of Online Attacks: an Analysis of Exploitation Behaviors on the Web

Davide Canali, Davide Balzarotti

Software and System Security Group

EURECOM, France

http://s3.eurecom.fr/

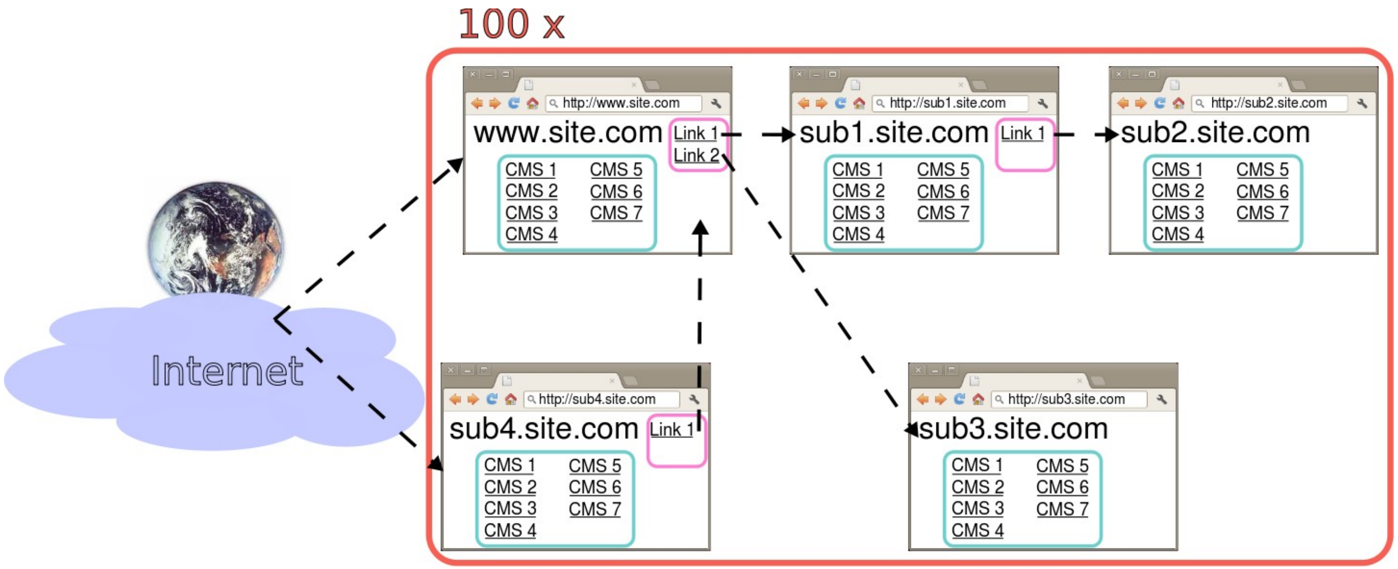
NDSS 2013

Motivations

- Studying the internals of web attacks
  - What attackers do while and after they exploit a vulnerability on a website
  - Understand why attacks are carried out (fun, profit, damaging others, etc.)
- Previous studies
  - how attacks against web sites are carried out
  - how criminals find their victims on the Internet
  - Lack of studies on the behavior of attackers (what they do during and after a typical attack)
    - Previous works used static, non functional honeypots (not exploitable)

How

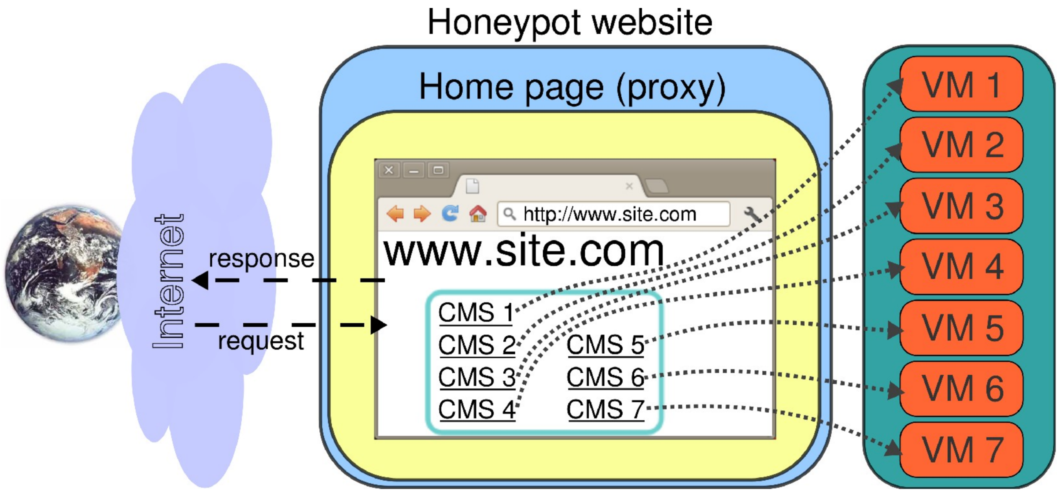
- 500 vulnerable websites deployed on the Internet



  - 100 domain names registered, with 5 subdomains each
  - Hosted on 9 of the Internet's biggest hosting providers
  - Each website contains 5 common CMSs (blog, forum, e-commerce web app, generic portal, SQL manager), 1 static website and 17 PHP web shells

Data collection

- 100 days of centralized data collection
- Allows for simple and effective management
- Each deployed website acts as a proxy
  - Redirects traffic to the real web applications installed on VMs in our premises

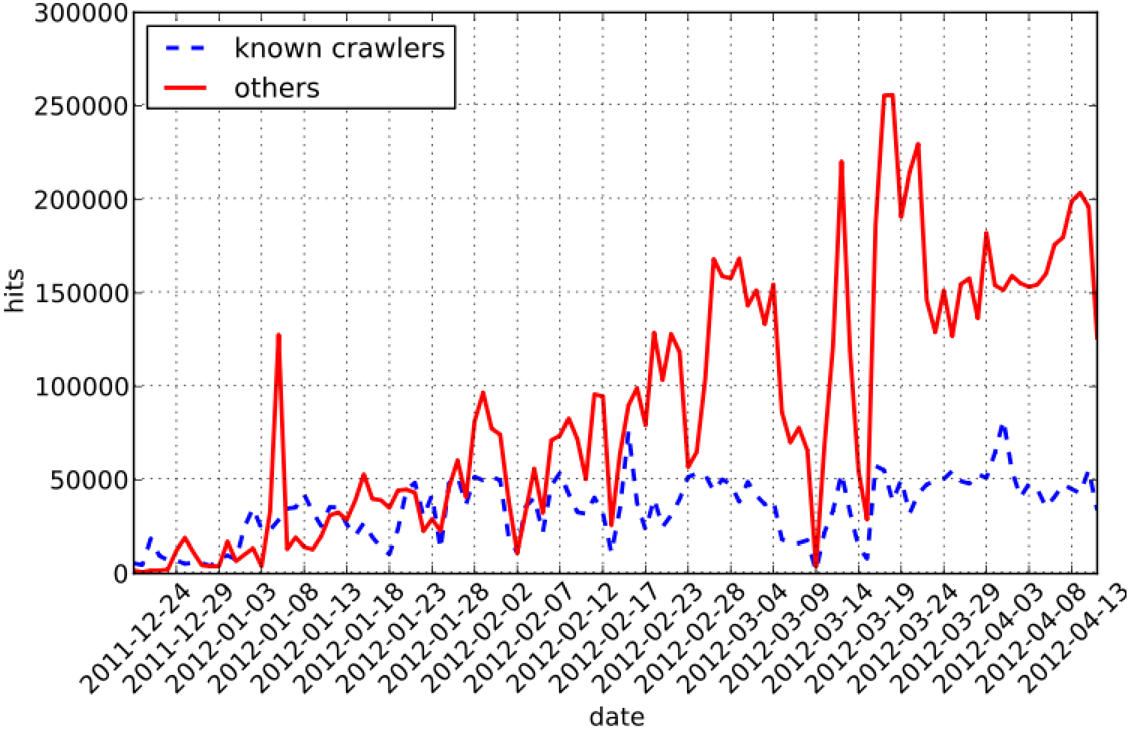


- » Easy to restore the VM state once an attack takes place
- » Full attack logs available
- » Easy to limit and tailor the attacker's privileges on the machine that hosts the vulnerable app

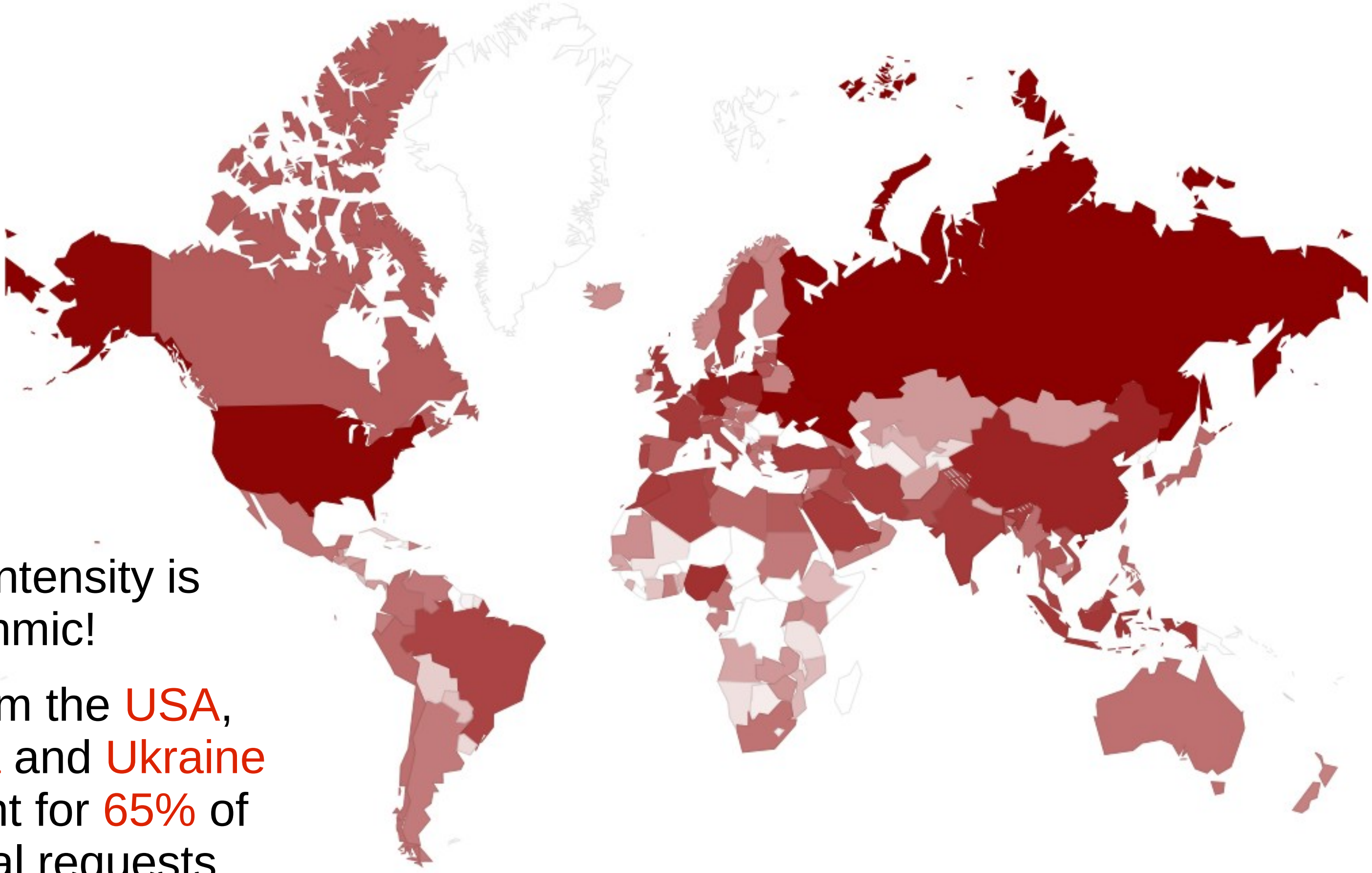
Collected data

- ~10 GB of raw HTTP requests
- In average:
  - 1-10K uploaded files every day
  - 100-200K HTTP requests/day
- First suspicious activities:
  - automated: 2h 10' after deployment
  - manual: after 4h 30'

Requests volume



Requests by country
(excluding known crawlers)



- Color intensity is logarithmic!
- IPs from the USA, Russia and Ukraine account for 65% of the total requests

Attack analysis

The four different phases

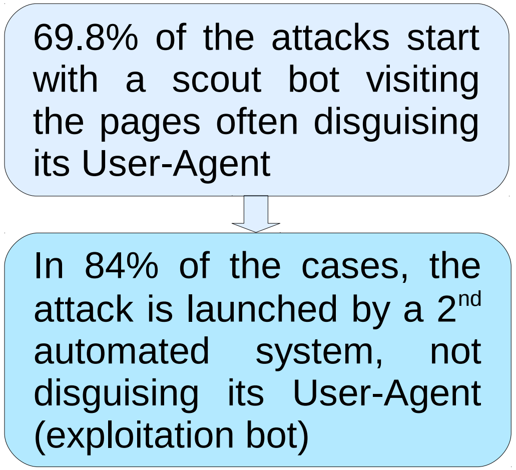
1. Discovery: how attackers find their targets
   - Referer analysis, dorks used to reach our websites, first suspicious activities

69.8% of the attacks start with a scout bot visiting the pages often disguising its User-Agent

Attack analysis

The four different phases

1. Discovery: how attackers find their targets
   - Referer analysis, dorks used to reach our websites, first suspicious activities
2. Reconnaissance: how pages were visited
   - Automated systems and crawling patterns identification, User-Agent analysis



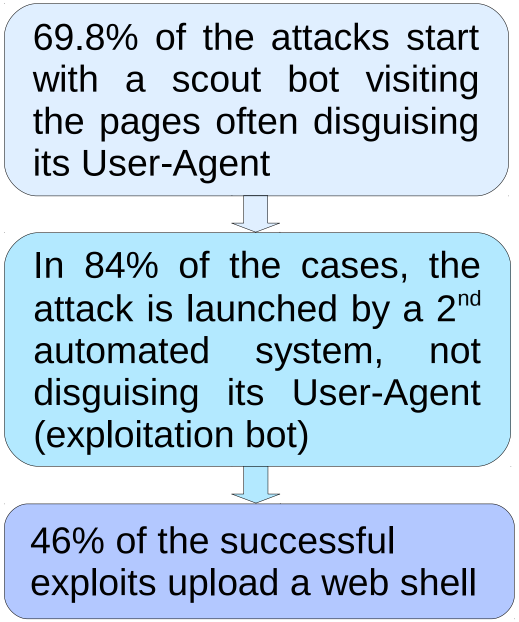
69.8% of the attacks start with a scout bot visiting the pages often disguising its User-Agent

In 84% of the cases, the attack is launched by a 2nd automated system, not disguising its User-Agent (exploitation bot)

Attack analysis

The four different phases

1. Discovery: how attackers find their targets
   - Referer analysis, dorks used to reach our websites, first suspicious activities
2. Reconnaissance: how pages were visited
   - Automated systems and crawling patterns identification, User-Agent analysis
3. Exploitation: attack against the vulnerable web app
   - Exploits detection and analysis, exploitation sessions, uploaded files categorization, and attack time/location normalization
   - Analysis of forum activities: registrations, posts and URLs, geolocation, message categories



69.8% of the attacks start with a scout bot visiting the pages often disguising its User-Agent

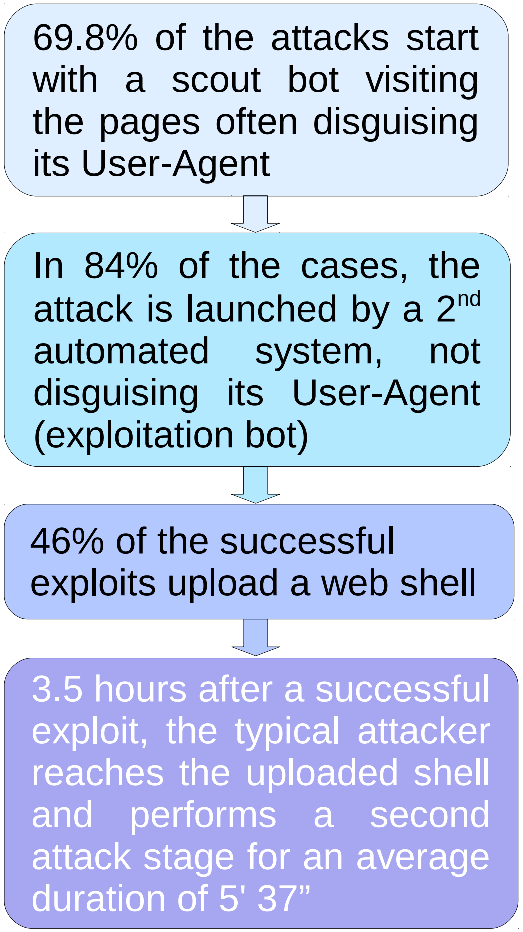
In 84% of the cases, the attack is launched by a 2nd automated system, not disguising its User-Agent (exploitation bot)

46% of the successful exploits upload a web shell

Attack analysis

The four different phases

1. Discovery: how attackers find their targets
   - Referer analysis, dorks used to reach our websites, first suspicious activities
2. Reconnaissance: how pages were visited
   - Automated systems and crawling patterns identification, User-Agent analysis
3. Exploitation: attack against the vulnerable web app
   - Exploits detection and analysis, exploitation sessions, uploaded files categorization, and attack time/location normalization
   - Analysis of forum activities: registrations, posts and URLs, geolocation, message categories
4. Post-Exploitation: second stage of the attack, usually carried out manually (optional)
   - Session identification, analysis of shell commands



69.8% of the attacks start with a scout bot visiting the pages often disguising its User-Agent

In 84% of the cases, the attack is launched by a 2nd automated system, not disguising its User-Agent (exploitation bot)

46% of the successful exploits upload a web shell

3.5 hours after a successful exploit, the typical attacker reaches the uploaded shell and performs a second attack stage for an average duration of 5' 37”

Attack analysis

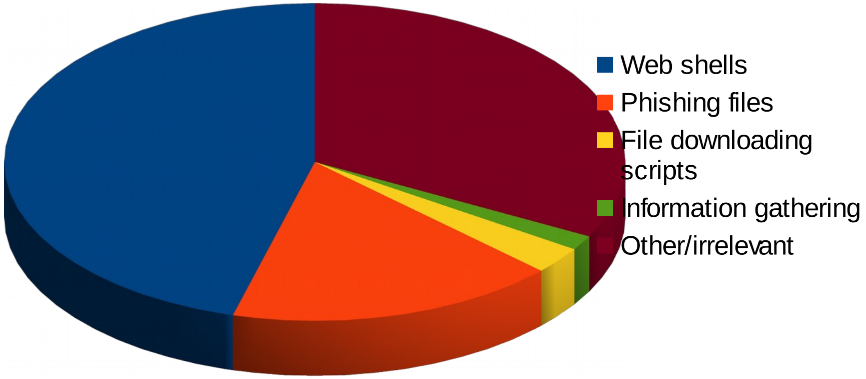
phases #1-2: discovery - reconnaissance

- Discovery: referer shows where visitors are coming from
  - Set in 50% of the cases
  - Attackers find our honeypots mostly from search engine queries (in the order: Google, Yandex, Bing, Yahoo)
    - Some visitors from 'hacking' search engines as well
  - Some visits from web mail services (spam or phishing victims) and social networks
- Reconnaissance: how were pages visited?
  - 84% of the malicious traffic was from automated systems
    - No images or style-sheets requested
    - Low inter-arrival time
    - Multiple subdomains visited within a short time frame
  - 6.8% of the requests mimicked the User-Agent string of known search engines

Attack analysis

phase #3: exploitation

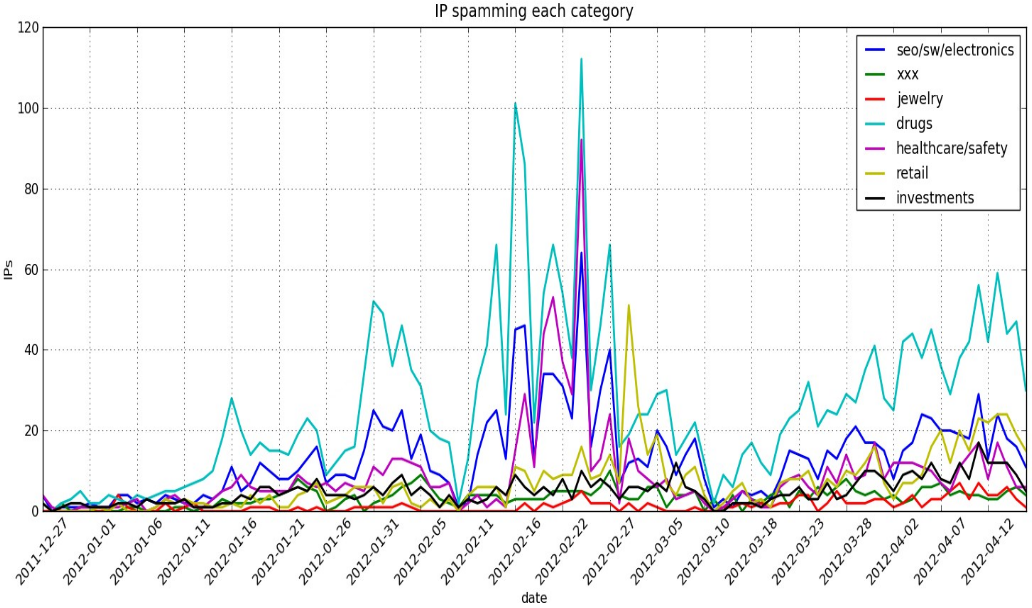
- We already know our applications' vulnerabilities
- 444 distinct exploitation sessions
  - Session = a set of requests that can be linked to the same origin, arriving within 5' from each other
  - 75% of the sessions used at least once '`libwww/perl`' as User-Agent string → scout bots and automatic attacks
- Almost one exploitation out of two uploaded a web shell, to continue the attack at a later stage (post-exploitation)



Attack analysis

phase #3: Forum activity

- Daily averages: 604 posts, 1907 registrations, 232 online users
  - One third of the IPs acting on the forum registered at least one account, but never posted any message → any business related to selling forum accounts?
- ~1% of the links posted to the forum led to malicious content[†]
- Geographical trends (active IPs)
  - 36.8% from the US
  - 24.6% from Eastern EU
- Simple message categorization
  - Keyword-based
  - Coverage: 93.5% of the forum posts (63,373)

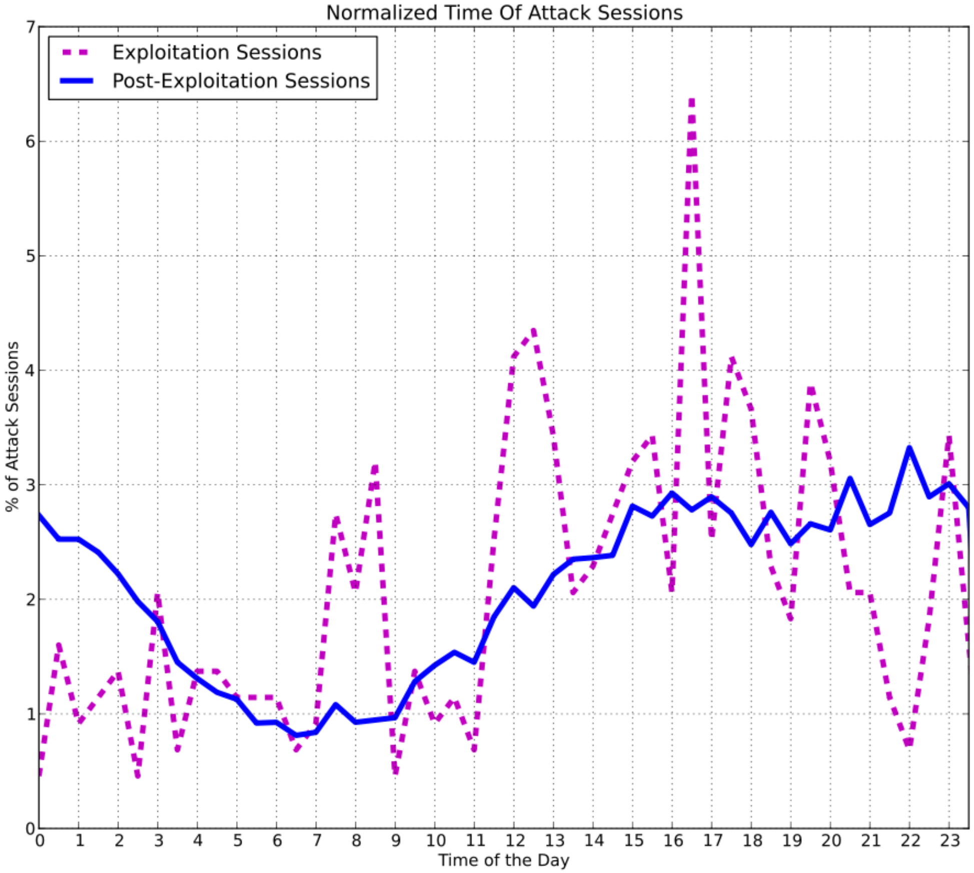


[†] According to Google SafeBrowsing and Wepawet

Attack analysis

phases #3-4

- Clear hourly trends for post-exploitation (manual) sessions



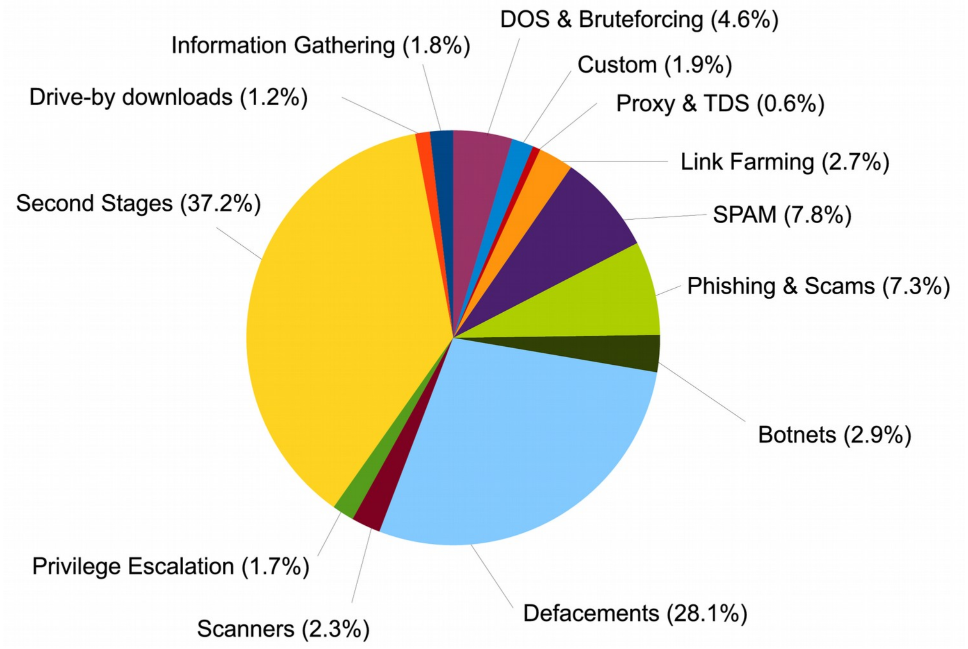
Attack analysis

phase #4: post-exploitation

- Almost 8500 interactive sessions collected
  - Known and unknown web shells
  - Average session duration: 5' 37”
    - 9 sessions lasting more than one hour
  - Parsed commands from the logs
    - 61% of the sessions upload a file to the system
    - 50% of the sessions (try to) modify existing files
      - Defacement in 13% of the cases

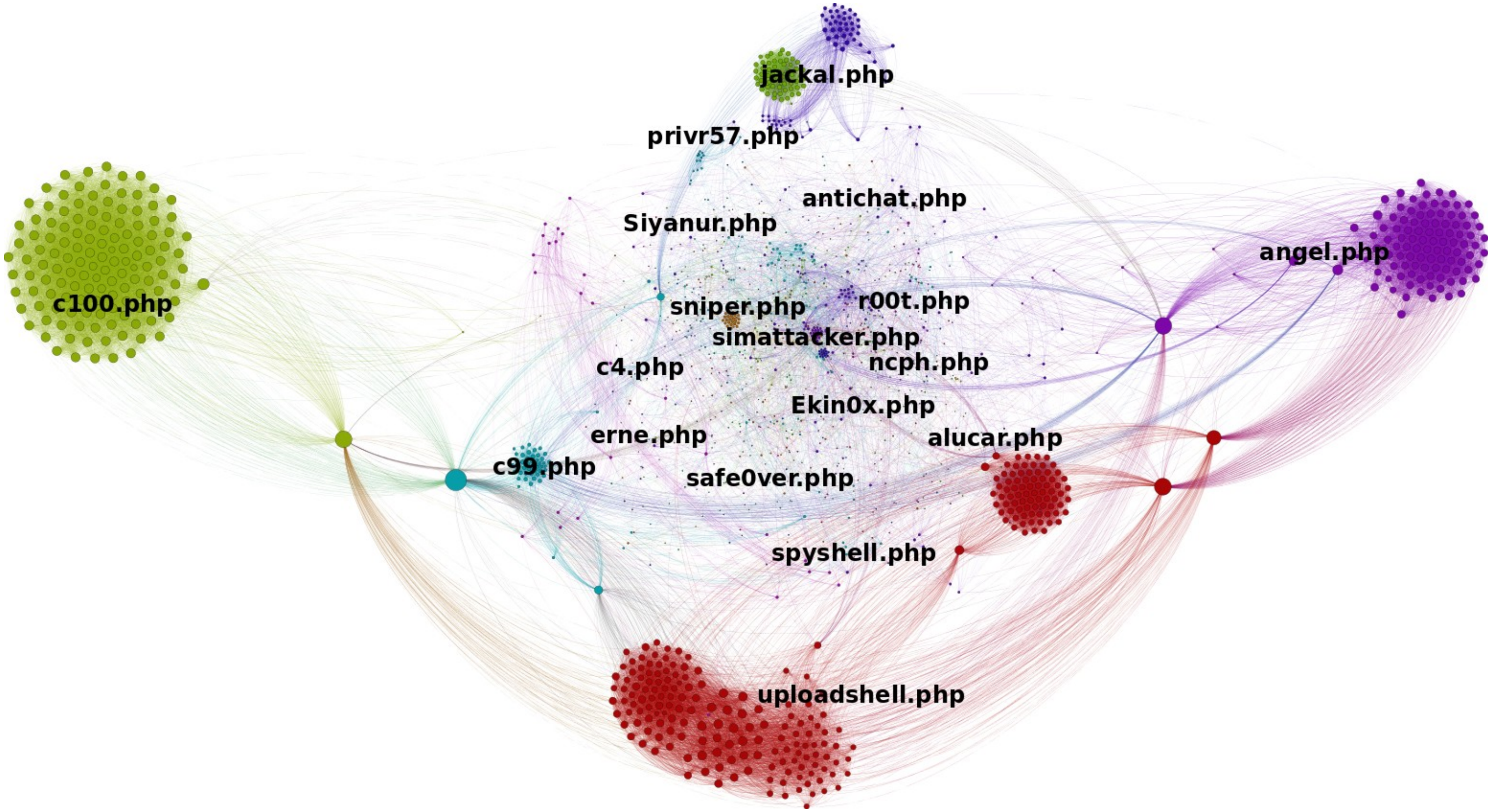
Attacker goals

- The analysis of collected files allows to understand the attackers' goals
  - » File normalization and similarity-based clustering
  - » Manual labeling of clusters



Clustering example

- Similarity clustering on web shells (ours are labeled)



Conclusions (so far)

- The study confirmed some known trends
  - Strong presence of Eastern European countries in spamming activities
  - Scam and phishing campaigns often run from African countries
  - Most common spam topic: pharmaceutical ads
- Unexpected results
  - High number of manual attacks
  - Many IRC botnets still around
  - Despite their low sophistication, these represent a large fraction of the attacks to which vulnerable websites are exposed every day

One surprising experience

- The honeypot proxies are hosted on various web hosting facilities
  - Many of them complain of the activity
  - At some point close our account

- We really don't do anything bad, we just get attacked!
  - How are they detecting this ?
  - Do they really care about their customer's security ?
  - That would be great !
- Let's check !

The Role of Web Hosting Providers in Detecting Compromised Websites

Davide Canali, Davide Balzarotti, Aurélien Francillon

Software and System Security Group

EURECOM, France

WWW 2013

Motivations

- **Shared web hosting** is used by **millions of users**
  - Host personal and small business websites
  - Users often have little or no security background
  - Even experienced users have little control/visibility
- Millions of websites, unexperienced users, outdated/vulnerable web apps → **huge attack surface**!
- Hosting providers should play a key role in helping the user in case of a compromise
  - Is this the case?

Goal

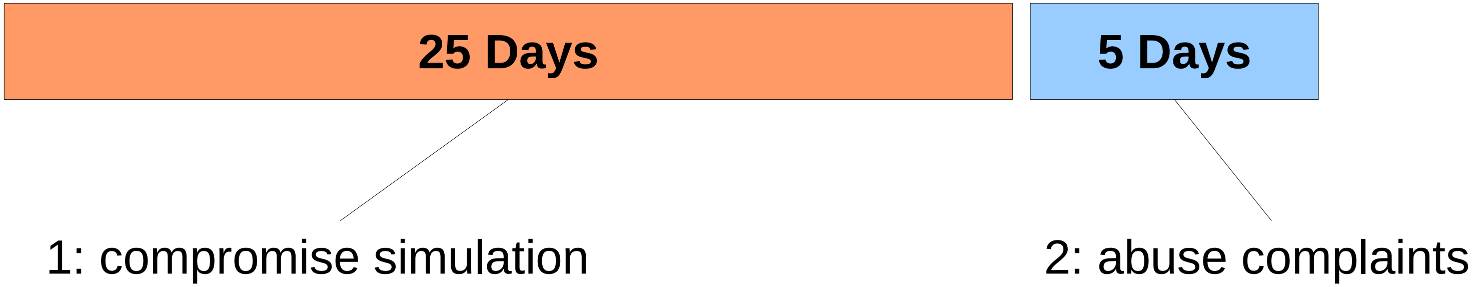
- Study how shared web hosting providers handle the security of their customers
  - By **detecting the compromise** of their websites
  - By testing their **reactions to abuse complaints**
- We also tested six **specialized security services**
  - Provided as an add-on for hosting accounts
  - Monitor security issues on websites
  - For a small fee

Testing methodology (1/2)

- **Register** multiple shared hosting accounts
- Install real web applications
- Simulate a number of **compromise scenarios**
  - Infected by botnet
  - Data exfiltration (SQL injection)
  - Phishing kit
  - Code inclusion (Drive-by-download)
  - Compromised account (upload of malicious files)
- **Tests** designed to be noisy and **easily detectable**

Testing methodology (2/2)

- Phase 1: observe the provider's reaction
- Phase 2: send **abuse complaints** regarding our websites
  - **Real complaints** about phishing and malicious executables
  - **Illegitimate complaints**, about offending or malicious content, while the account was clean



Ethical Issues

- We used real vulnerabilities, a real phishing kit, and a real drive-by javascript code
- But
  - we modified the sources to be **exploitable only by us** (special parameters)
  - **not indexable** by search engines (robot.txt)
  - malicious content was **not accessible from the web** or disabled

Tested Providers

- **12** among the **top global ones** (mostly US-based)
- **10 regional ones**
  - From Europe, US, India, Russia, Algeria, Hong Kong, Argentina, Indonesia

- 6 add-on security services
  - Less than 30 $/month subscription fee
  - Two come in *basic* and *pro* version
  - 10 days detection threshold
    (we expected them to be quick at detecting security issues)

Scenarios details

- Infected by botnet
- Data exfiltration (SQL injection)
- Phishing kit
- Code inclusion (Drive-by-download)
- Compromised account (upload of malicious files)

Bot Test Case

- Suspicious Network Activity: IRC Bot (**Bot**)

  Setup

  - Base OsCommerce installation (no modifications)
  - Two executable files (same IRC client, compiled for 32 and 64 bit architectures) and a PHP script executing the right binary depending on the machine's configuration
    - The IRC client connects to a fake IRC server (run by us), issues some IRC commands, and closes the connection

  Attack (run every hour)

  - Uploads the PHP file and the two binaries to the shared hosting account via FTP (case of an attacker using stolen credentials)
  - Launches the IRC client by issuing a request to the PHP page

SQL injection and Data Exfiltration (**SQLi**)

Setup

- OsCommerce installation mimicking a known SQL injection vulnerability
- Source code modified to return personal details and credit card numbers of fictious people

Attack (run every hour)

- Sequence of GET requests simulating an automated SQL injection tool enumerating entries in the 'customers' table of the CMS.
- Requests include several common SQL reserved words, to test if providers employ any keywork-based URL blacklisting

Remote File Upload of a Phishing Kit

Setup

- OsCommerce installation mimicking a known **Remote File Upload** vulnerability
- Performs the upload a real Bank of America **phishing kit** (disabled back-end code)

Attack

- *Attacker phase*, run every 6 hours: uploads the phishing kit by triggering the vulnerability
- *Victim phase,* every 15': simulates a victim falling prey of the phishing attack
  - The forms on the phishing pages are filled up with a set of fake personal details (manually pre-generated)

Compromised Account
(upload of known malicious files)

Setup

- Static HTML page with random English sentences and some pictures
- Two **known malicious files** (PHP and executable)
  - *c99.php*: a real c99 web shell
  - *sb.exe*: Ramnit worm
  - Both detected by most antiviruses

Attack

- **Uploads** the two malicious files to the shared hosting account **via FTP** (attacker using stolen credentials)
- Run every 6 hours

Web Shell

- File Upload and Code Injection using Web Shell (**SH**)

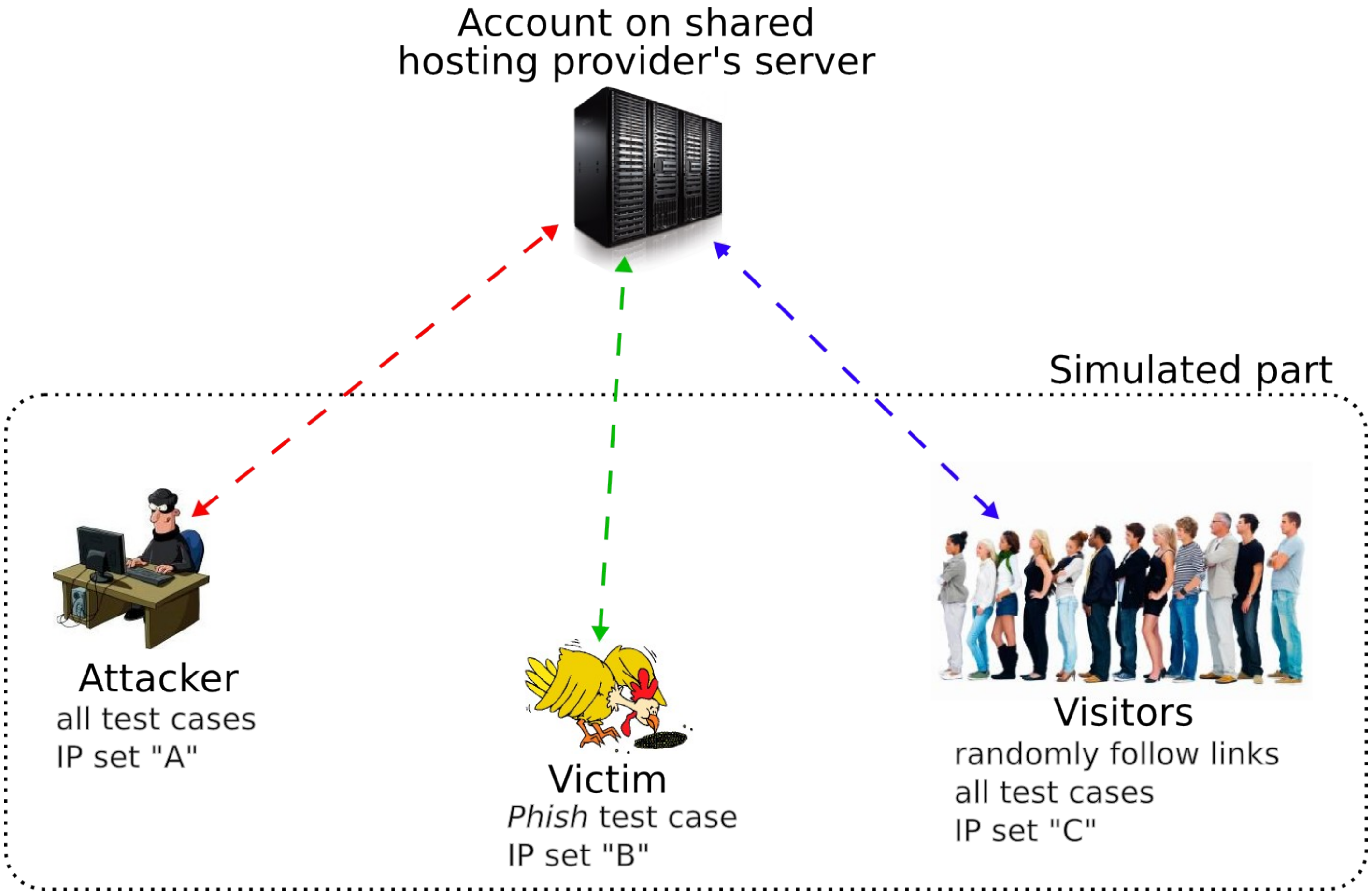
  Setup

  - OsCommerce installation mimicking a known Remote File Upload vulnerability
  - Source code modified to allow the file upload only when the request contains a secret keyword
    - We upload a known php web shell (c99)
  - The web shell is modified to allow only injecting some malicious drive-by code on the website's home page
    - Malicious JS code disabled by a dynamic check (still detected by AVs)

  Attack (run every hour)

  - Performs the upload of the web shell
  - simulates somebody using the the shell to access known files
  - injects the malicious drive-by download in the home page

Experiment scheme



Results

- Registration <= Surprise
- Attack prevention
- Compromise detection
- Response to abuse complaints

Results: registration

- Some providers **discourage abusive** user **registrations**
  - Phone calls, ID scan, 3rd party fraud protection services
- **Global providers** are **more cautious** than regional ones
  - 58% of them manually verified at least one of our accounts (10% for regional)
- **Three regional** providers have a very simple **“1-step” signup process**
  - Never verified our information upon registration

Results: prevention and detection

- Attack **prevention** measures work to some extent
  - **URL blacklists** to block SQL injections and File Uploads
    - SQLi, SH, Phish in ~30% of the cases
  - Connection and OS-level **filtering** are effective (Bot)
  - Some providers seem to employ the same (commercial) rule sets for blocking attacks

- Attack **detection results** are quite **disappointing**
  - **Only one provider** was able to detect **one** of our attacks
  - Received alert for **test AV after 17 days** it was running

Results

- **Prevention**

Tests	**SQLi**	**SH**	**Phish**	**Bot**	**AV**
Fully blocked	**0**	4	6	18	-
Partially blocked	7	2	0	2	-
Not blocked	13	**16**	16	2	-

Results: abuse complaints

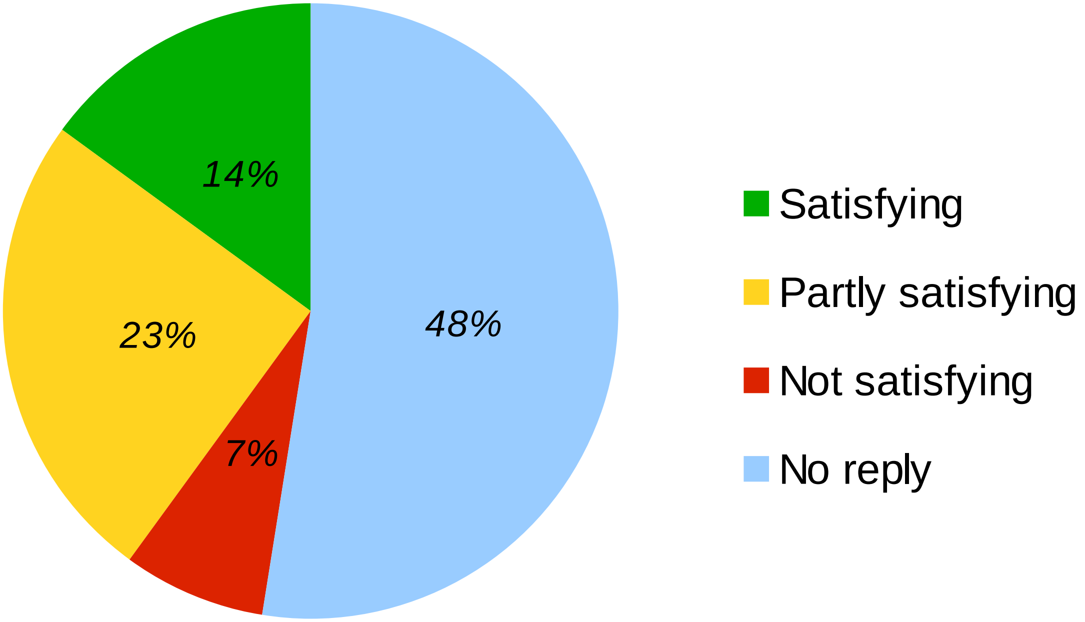
- **50%** of the tested providers **never replied** to any notification

- **64%** of the replies arrived **within one day** from the notification

- Average response delay:
  - **28h** for **global** providers
  - **79h** for **regional** providers

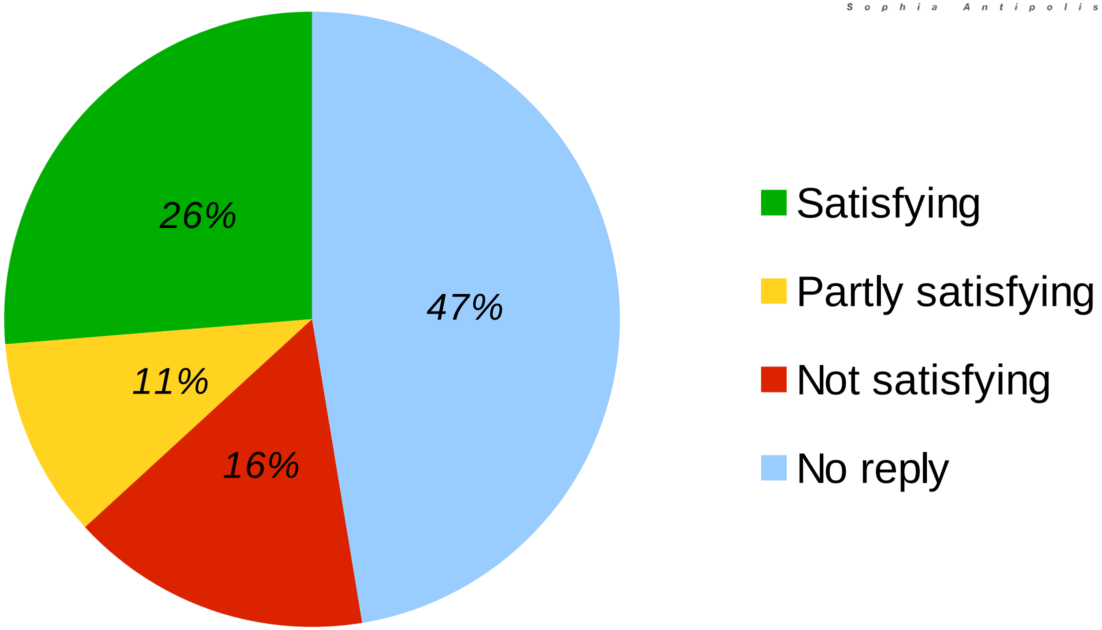
- Wide variety of reactions...

Real abuse notification handling



- **Only 3** providers **out of 22** handled them well
- Some **overreact** (e.g., two of them terminated the user's account)
  - Others sent an ultimatum to the user, but then did not check whether the user did anything to clean up the account

Illegitimate abuse notification handling



- **14** providers **out of 19** tested behaved well
  - » **Over estimation** (some did not answer)
- 3 (regional) providers believed the complaint without checking
  - – completely **wrong decisions** (e.g., account suspension, file removal)

Detection by Security add-on Services

- Some of the services we tested had a partnership with a **URL blacklisting service**

  → We intentionally got our malicious pages blacklisted

- **Five out of six** services did **not detect anything**

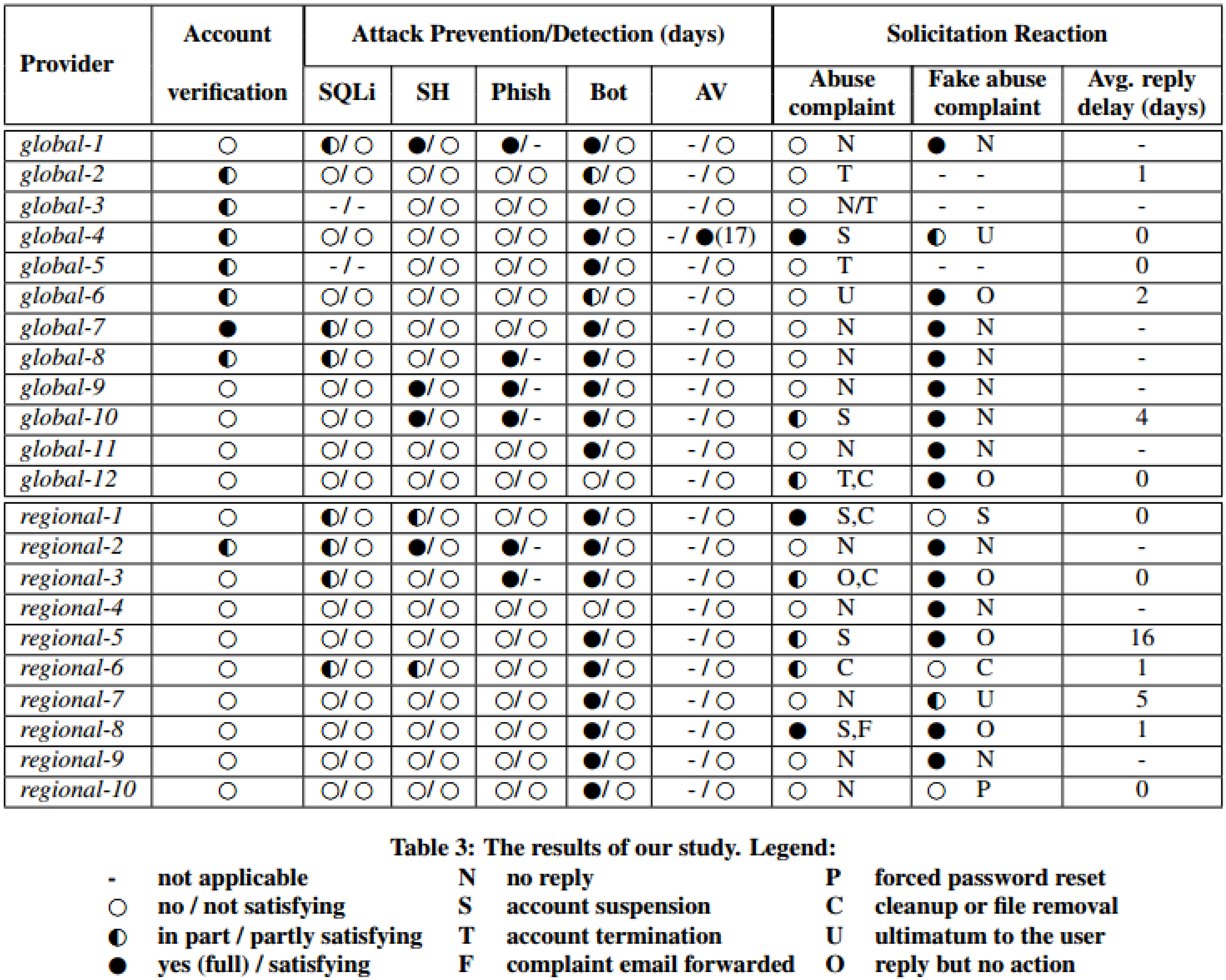
- One detected
  - the malicious files (through an antivirus scan)
    but they did **NOT notify the user**
  - the blacklisted malicious page

Conclusions

- Quite a **lot of effort** is spent in **preventing** malicious **registrations**
  - Especially from **global** providers
  - Revenue protection...
- Most providers employ **basic** mechanisms to **prevent** some kinds of **attack** (e.g., URL blacklists)
- Almost **zero effort** in **detecting obvious** signs of **compromise**
- **Cheap security services are useless**
- **Half of the companies responded** to complaints
  - Only 14% in the appropriate way

Thank you

?



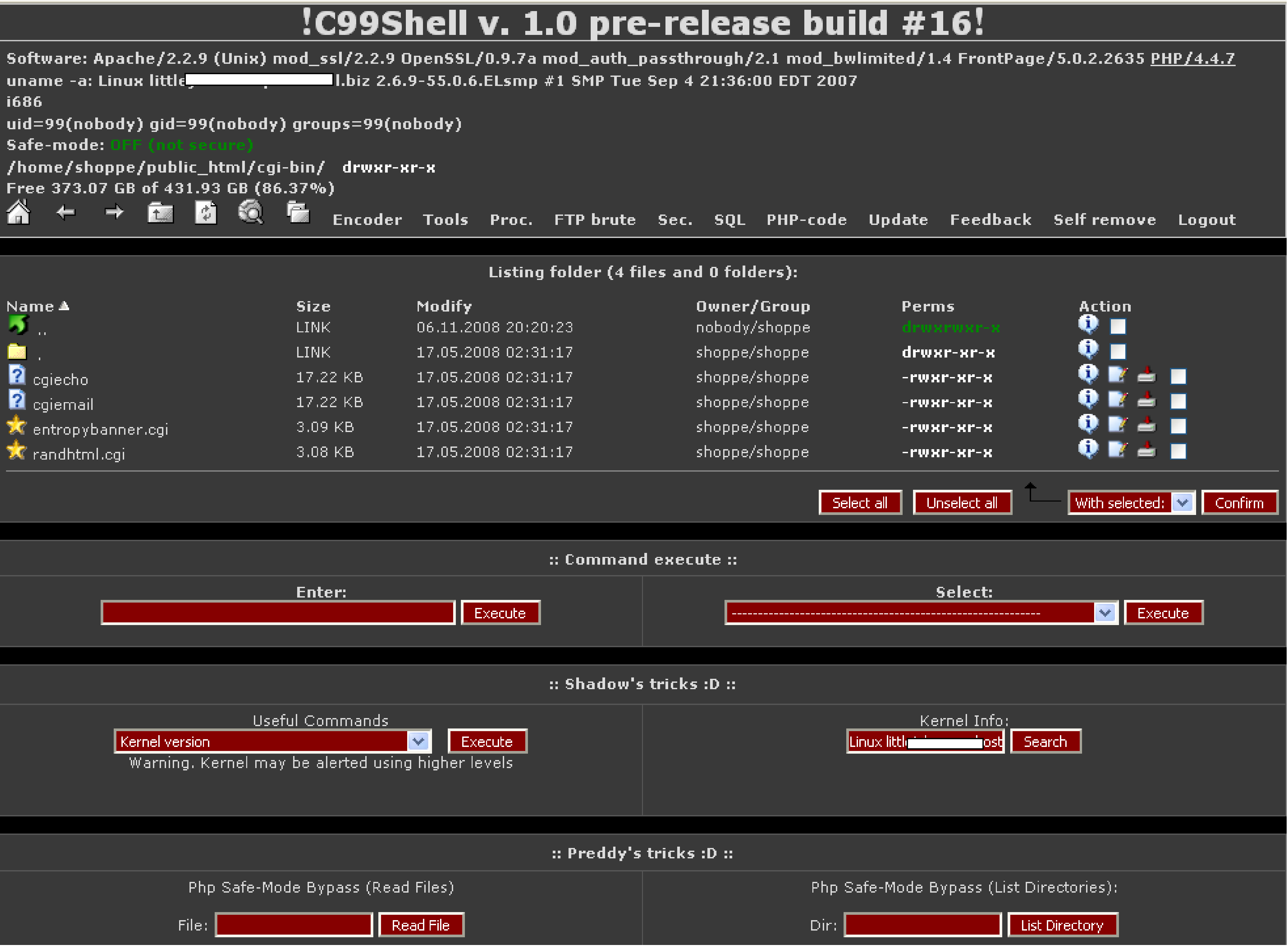
Provider	Account	Attack Prevention/Detection (days)					Solicitation Reaction		
	verification	SQLi	SH	Phish	Bot	AV	Abuse complaint	Fake abuse complaint	Avg. reply delay (days)
global-1	○	◐/ ○	●/ ○	●/ -	●/ ○	- / ○	○ N	● N	-
global-2	◐	○/ ○	○/ ○	○/ ○	◐/ ○	- / ○	○ T	- -	1
global-3	◐	- / -	○/ ○	○/ ○	●/ ○	- / ○	○ N/T	- -	-
global-4	◐	○/ ○	○/ ○	○/ ○	●/ ○	- / ●(17)	● S	◐ U	0
global-5	◐	- / -	○/ ○	○/ ○	●/ ○	- / ○	○ T	- -	0
global-6	◐	○/ ○	○/ ○	○/ ○	◐/ ○	- / ○	○ U	● O	2
global-7	●	◐/ ○	○/ ○	○/ ○	●/ ○	- / ○	○ N	● N	-
global-8	◐	◐/ ○	○/ ○	●/ -	●/ ○	- / ○	○ N	● N	-
global-9	○	○/ ○	●/ ○	●/ -	●/ ○	- / ○	○ N	● N	-
global-10	○	○/ ○	●/ ○	●/ -	●/ ○	- / ○	◐ S	● N	4
global-11	○	○/ ○	○/ ○	○/ ○	●/ ○	- / ○	○ N	● N	-
global-12	○	○/ ○	○/ ○	○/ ○	○/ ○	- / ○	◐ T,C	● O	0
regional-1	○	◐/ ○	◐/ ○	○/ ○	●/ ○	- / ○	● S,C	○ S	0
regional-2	◐	◐/ ○	●/ ○	●/ -	●/ ○	- / ○	○ N	● N	-
regional-3	○	◐/ ○	○/ ○	●/ -	●/ ○	- / ○	◐ O,C	● O	0
regional-4	○	○/ ○	○/ ○	○/ ○	○/ ○	- / ○	○ N	● N	-
regional-5	○	○/ ○	○/ ○	○/ ○	●/ ○	- / ○	◐ S	● O	16
regional-6	○	◐/ ○	◐/ ○	○/ ○	●/ ○	- / ○	◐ C	○ C	1
regional-7	○	○/ ○	○/ ○	○/ ○	●/ ○	- / ○	○ N	◐ U	5
regional-8	○	○/ ○	○/ ○	○/ ○	●/ ○	- / ○	● S,F	● O	1
regional-9	○	○/ ○	○/ ○	○/ ○	●/ ○	- / ○	○ N	● N	-
regional-10	○	○/ ○	○/ ○	○/ ○	●/ ○	- / ○	○ N	○ P	0

Table 3: The results of our study. Legend:

-	not applicable	N	no reply	P	forced password reset
○	no / not satisfying	S	account suspension	C	cleanup or file removal
◐	in part / partly satisfying	T	account termination	U	ultimatum to the user
●	yes (full) / satisfying	F	complaint email forwarded	O	reply but no action

Honeypot Websites

- Honeypot pages linked to our homepages in order to be easily reachable by search engine bots
  - Search engine indexing is a key factor for attracting automated (attack) bots
- Installed vulnerable apps:
  - Blog (Wordpress)
  - Forum (SMF)
  - E-commerce application (osCommerce)
  - Generic portal CMS (Joomla)
  - Database management CMS (phpMyAdmin)
  - 17 common PHP web shells + static website (defacements)



!C99Shell v. 1.0 pre-release build #16!

```
Software: Apache/2.2.9 (Unix) mod_ssl/2.2.9 OpenSSL/0.9.7a mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635 PHP/4.4.7
uname -a: Linux little[redacted]l.biz 2.6.9-55.0.6.ELsmp #1 SMP Tue Sep 4 21:36:00 EDT 2007
i686
uid=99(nobody) gid=99(nobody) groups=99(nobody)
Safe-mode: OFF (not secure)
/home/shoppe/public_html/cgi-bin/  drwxr-xr-x
Free 373.07 GB of 431.93 GB (86.37%)
```

Listing folder (4 files and 0 folders):

Name ▲	Size	Modify	Owner/Group	Perms	Action
..	LINK	06.11.2008 20:20:23	nobody/shoppe	drwxrwxr-x	□
.	LINK	17.05.2008 02:31:17	shoppe/shoppe	drwxr-xr-x	□
cgiecho	17.22 KB	17.05.2008 02:31:17	shoppe/shoppe	-rwxr-xr-x	□
cgiemail	17.22 KB	17.05.2008 02:31:17	shoppe/shoppe	-rwxr-xr-x	□
entropybanner.cgi	3.09 KB	17.05.2008 02:31:17	shoppe/shoppe	-rwxr-xr-x	□
randhtml.cgi	3.08 KB	17.05.2008 02:31:17	shoppe/shoppe	-rwxr-xr-x	□

:: Command execute ::

Enter:

Select: --

:: Shadow's tricks :D ::

Useful Commands: Kernel version

Warning. Kernel may be alerted using higher levels

Kernel Info: Linux littl[redacted]ost

:: Preddy's tricks :D ::

Php Safe-Mode Bypass (Read Files)

File:

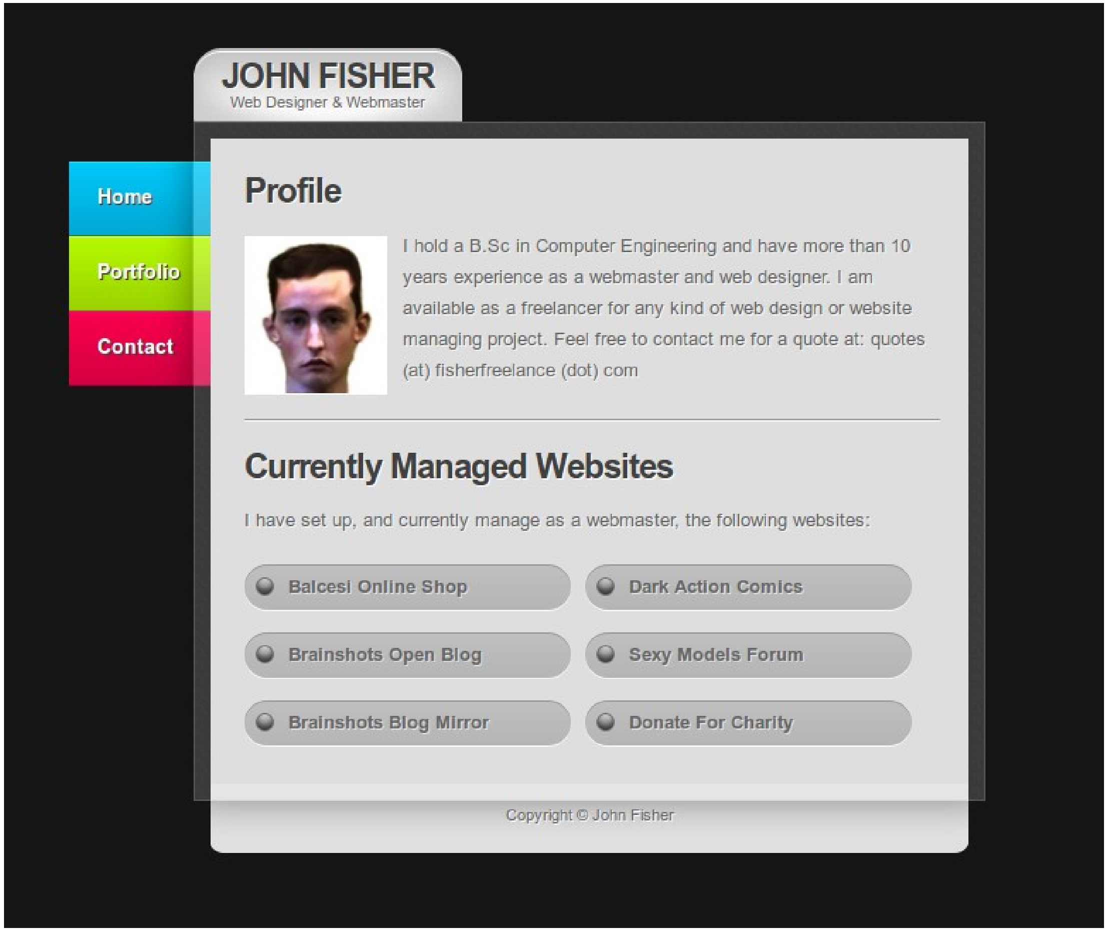
Php Safe-Mode Bypass (List Directories):

Dir:

Containment

- Avoid external exploitation and privilege escalations
  - Only 1 service (apache) exposed to the Internet
    - run as unprivileged user
  - Up to date software and security patches
- Avoid using the honeypot as a stepping stone for attacks
  - Blocked all outgoing traffic (except for IRC)
- Avoid hosting illegal content (mitigated)
  - Preventing the modification of directories, html and php files (chmod)
  - Regular restore of each VM to its original snapshot
- Avoid promoting illegal goods or services
  - Code showing content of user posts and comments commented out for each CMS
    - users and search engines are shown blank messages

Home page



JOHN FISHER
Web Designer & Webmaster

Profile

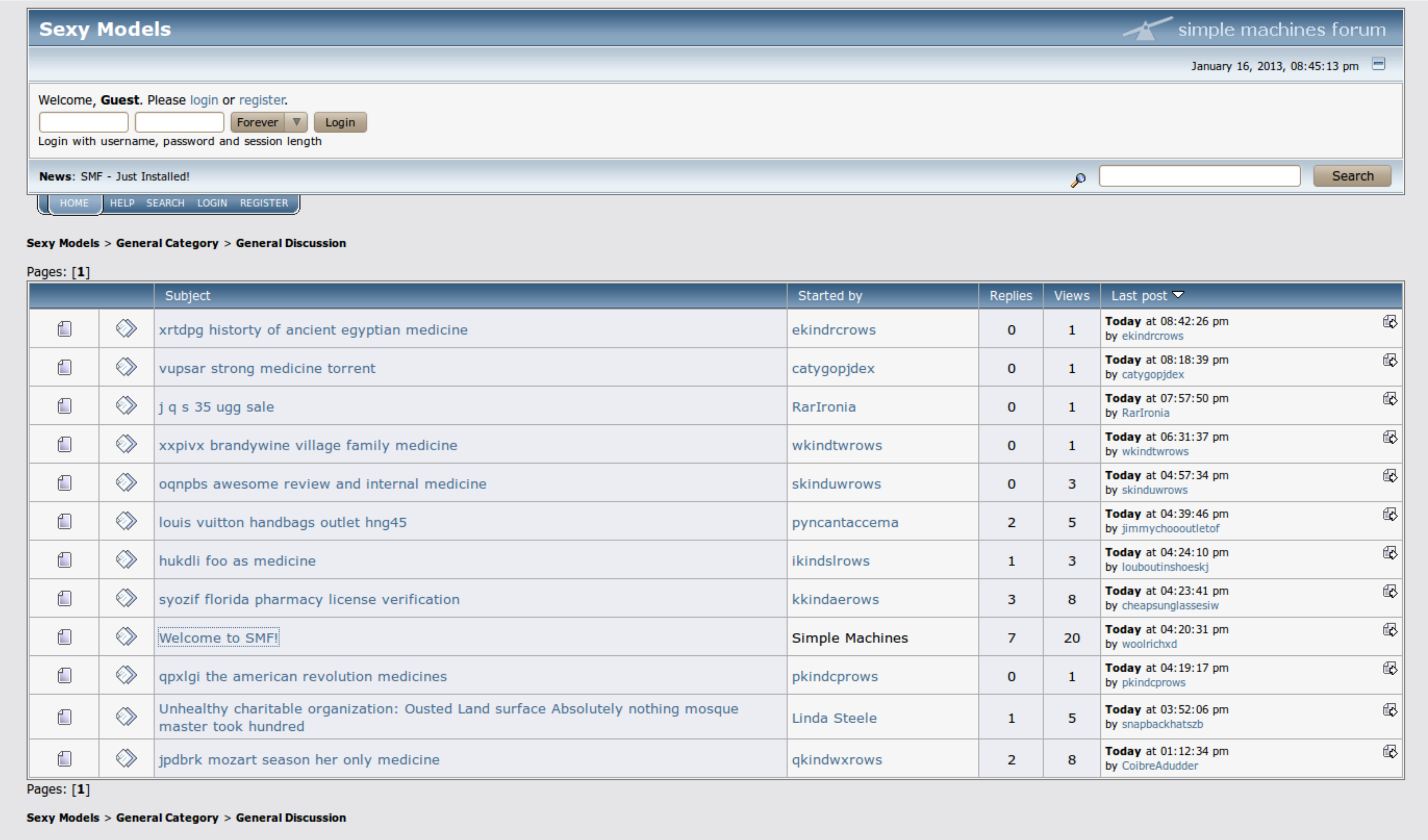
I hold a B.Sc in Computer Engineering and have more than 10 years experience as a webmaster and web designer. I am available as a freelancer for any kind of web design or website managing project. Feel free to contact me for a quote at: quotes (at) fisherfreelance (dot) com

Currently Managed Websites

I have set up, and currently manage as a webmaster, the following websites:

Copyright © John Fisher

Forum



Sexy Models

January 16, 2013, 08:45:13 pm

Welcome, **Guest**. Please login or register.

Login with username, password and session length

News: SMF - Just Installed!

Sexy Models > **General Category** > **General Discussion**

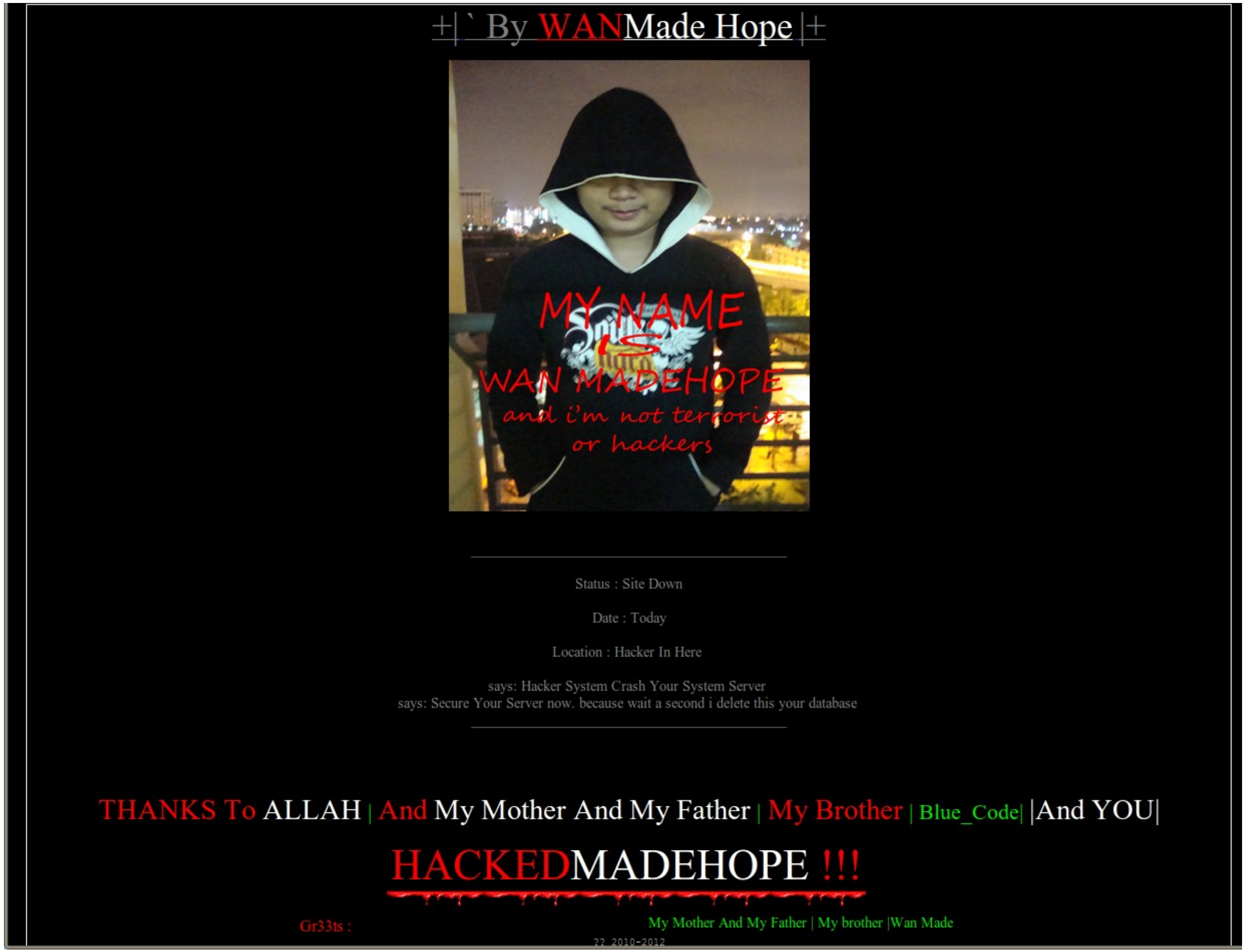
Pages: [**1**]

		Subject	Started by	Replies	Views	Last post
		xrtdpg historty of ancient egyptian medicine	ekindrcrows	0	1	**Today** at 08:42:26 pm by ekindrcrows
		vupsar strong medicine torrent	catygopjdex	0	1	**Today** at 08:18:39 pm by catygopjdex
		j q s 35 ugg sale	RarIronia	0	1	**Today** at 07:57:50 pm by RarIronia
		xxpivx brandywine village family medicine	wkindtwrows	0	1	**Today** at 06:31:37 pm by wkindtwrows
		oqnpbs awesome review and internal medicine	skinduwrows	0	3	**Today** at 04:57:34 pm by skinduwrows
		louis vuitton handbags outlet hng45	pyncantaccema	2	5	**Today** at 04:39:46 pm by jimmychooooutletof
		hukdli foo as medicine	ikindslrows	1	3	**Today** at 04:24:10 pm by louboutinshoeskj
		syozif florida pharmacy license verification	kkindaerows	3	8	**Today** at 04:23:41 pm by cheapsunglassesiw
		Welcome to SMF!	Simple Machines	7	20	**Today** at 04:20:31 pm by woolrichxd
		qpxlgi the american revolution medicines	pkindcprows	0	1	**Today** at 04:19:17 pm by pkindcprows
		Unhealthy charitable organization: Ousted Land surface Absolutely nothing mosque master took hundred	Linda Steele	1	5	**Today** at 03:52:06 pm by snapbackhatszb
		jpdbrk mozart season her only medicine	qkindwxrows	2	8	**Today** at 01:12:34 pm by CoibreAdudder

Pages: [**1**]

Sexy Models > **General Category** > **General Discussion**

Defacement



+|` By WANMade Hope |+

Status : Site Down

Date : Today

Location : Hacker In Here

says: Hacker System Crash Your System Server
says: Secure Your Server now. because wait a second i delete this your database

THANKS To ALLAH | And My Mother And My Father | My Brother | Blue_Code| |And YOU|

HACKEDMADEHOPE !!!

Gr33ts : My Mother And My Father | My brother |Wan Made

?? 2010-2012

Conclusions

- Need for a better protection of shared hosting accounts
  - Shared hosting is where most of the web attacks and malware campaigns spread
  - Everybody would benefit from providers adopting stronger security measures
    - … whether or not security scans/IDS systems are part of their TOS (often not the case)
  - We showed this can be easily accomplished even by using common open source solutions
    - Effective and easy to deploy

Legal

- The TOS of tested providers did not include anything related to detecting and notifying customers about compromises of their websites
  - The client can't do almost anything to protect himself, the provider is the only one who can

Test case detection by state-of-the-art tools

Test	SQLi	SH	Phish	Bot	AV
ModSecurity base rule set	🟢	🟢	🟢	🟢	-
ModSecurity OWASP rule set	🔴	🟡	🟢	🟢	-
High severity IDS alerts	⚠ (5)	⚠ (2)	⚠ (2)	0	0
Antivirus detection	🟢	🔴	🟢	🟢	🔴

Tests executed against an installation of SecurityOnion Linux, which includes, among other tools, the Bro IDS, Snort and Sguil.