Maintaining Websites with Security and Privacy

Kevin Gallagher
@ageis
GPG: 0x5F4F47885921D69C

NERDSummit 2014

Prerequisites . . .

- Experience with DevOps or systems administration
- Knowledge of LAMP stack (Linux, Apache, MySQL, PHP)
- Familiarity with the shell (bash) and editing (vim/nano/emacs)

Because it's important!

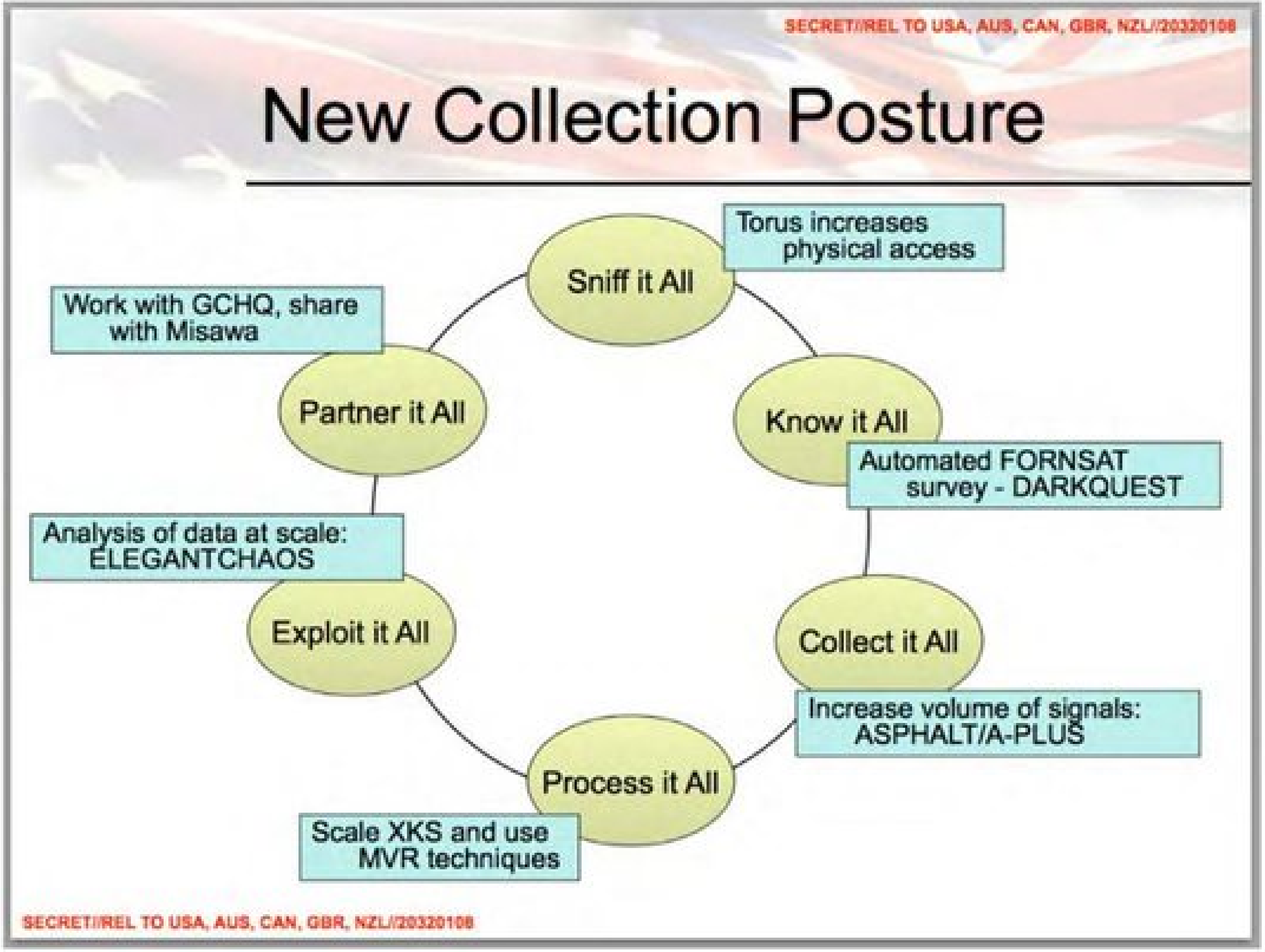
- Surveillance, both mass and targeted
See: Snowden documents, NSA,GCHQ

- Hackers and cyber-criminals (botnets, rootkits, malware, brute-force login attempts)

- Insecure public WiFi networks, corporate monitoring

- Data privacy for your clients and their visitors

- Cybersecurity incidents are on the rise



SECRET//REL TO USA, AUS, CAN, GBR, NZL//20320108

New Collection Posture

Sniff it All
Torus increases physical access
Know it All
Automated FORNSAT survey - DARKQUEST
Collect it All
Increase volume of signals: ASPHALT/A-PLUS
Process it All
Scale XKS and use MVR techniques
Exploit it All
Analysis of data at scale: ELEGANTCHAOS
Partner it All
Work with GCHQ, share with Misawa

SECRET//REL TO USA, AUS, CAN, GBR, NZL//20320108

Setting up the virtual host . . .

Drupal

```
# protect Git data
RewriteEngine On
RewriteRule ^/.git.*$ - [F,L]

# block access to Drupal stuff
RewriteRule .*/admin$          - [F]
RewriteRule .*/admin/(.*)?$    - [F]
RewriteRule .*/user$           - [F]
RewriteRule .*/user/(.*)?$     - [F]
RewriteRule .*/update.php    - [F]

# prevent Drupal version fingerprinting
RewriteRule ^/CHANGELOG.txt$ - [F]
RewriteRule ^/COPYRIGHT.txt$ - [F]
RewriteRule ^/INSTALL.mysql.txt$ - [F]
RewriteRule ^/INSTALL.pgsql.txt$ - [F]
RewriteRule ^/INSTALL.sqlite.txt$ - [F]
RewriteRule ^/INSTALL.txt$ - [F]
RewriteRule ^/LICENSE.txt$ - [F]
RewriteRule ^/MAINTAINERS.txt$ - [F]
RewriteRule ^/README.txt$ - [F]
RewriteRule ^/UPGRADE.txt$ - [F]

# disable trace and track
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

WordPress

```
# block the include-only files
RewriteBase /
RewriteRule ^wp-admin/includes/ - [F,L]
RewriteRule !^wp-includes/ - [S=3]
RewriteRule ^wp-includes/[^/]+\.php$ - [F,L]
RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
RewriteRule ^wp-includes/theme-compat/ - [F,L]

<Files wp-config.php>
        Order Allow,Deny
        Deny from All
</Files>
```

```
# block any file that starts with "."
<FilesMatch "^\..*$">
        Order allow,deny
</FilesMatch>
<FilesMatch "^.*\..*$">
        Order allow,deny
</FilesMatch>

# allow "." files with safe content types
<FilesMatch "^.*\.(css|html?|txt|js|xml|xsl|gif|ico|jpe?g|png)$">
        Order deny,allow
</FilesMatch>
```

```
Options -Indexes
```

Avoid version fingerprinting, don't serve extraneous files, and don't expose the admin login page to the whole internet. Login through a subdomain with HTTP Basic Authentication, or rename directories such as *wp-admin* for security through obscurity.

What to do with logs?

```
Apache's default log format:

    LogFormat "%h %l %u %t \"%r\" %>s %b" common

Make your own:

    LogFormat "- %l %u %t \"%r\" %>s %b" noip

Use it (/etc/apache/sites-enabled/*):

    CustomLog /var/logs/apache2/access.log noip

    LogLevel warn
/etc/logrotate.conf:

    maxage 30

$ ln -s /dev/null /var/log/apache2/access.log

$ ln -s /dev/null ~/.bash_history

~/.bashrc:
    HISTSIZE=32
    HISTFILESIZE=0
```

See cryptolog
or mod_log_iphash to
anonymize IP addresses.

If the data doesn't exist,
it can't be turned over!

https://git.eff.org/?p=cryptolog.git;a=summary
http://wiki.bitstreet.org/Mod_log_iphash

File permissions . . .

Directories

```
$ find . -type d -exec chmod 755 {} \;
```

Files

```
$ find . -type f -exec chmod 644 {} \;
```

Lock things down when you're done!

```
$ chmod a-w ./sites/default/settings.php

$ chmod a-w ./wp-config.php

$ chmod 444 .htaccess
```

You have multiple sites on one box and need to compartmentalize so that scripts run with the permission of their owners.

Use suEXEC, suPHP or FastCGI!

Why are we interested in HTTP?

facebook

YAHOO!

twitter

myspace.com a place for friends

Because nearly everything a typical user does on the Internet uses HTTP

CNN.com

Google Earth

WIKIPEDIA The Free Encyclopedia

@mail.ru

Gmail by Google BETA

Doing SSL/TLS right . . .

Redirect HTTP→HTTPS

```
RewriteEngine On
RewriteCond %{HTTPS} Off
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

HSTS (HTTP Strict Transport Security)

```
Header set Strict-Transport-Security "max-
age=16070400; includeSubDomains"
```

Preferred Cipher Suites

```
SSLHonorCipherOrder On
SSLProtocol all -SSLv2 -SSLv3
SSLCompression Off
SSLCipherSuite
EECDH+AES128:RSA+AES128:EECDH+AES256:RSA+AES256:
EECDH+3DES:RSA+3DES:EECDH+RC4:RSA+RC4:!MD5
```

- Perfect Forward Secrecy

Test your site at SSL Labs:
https://www.ssllabs.com/ssltest/

Secure Cookies

```
Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure

/etc/php5/apache2/php.ini:
      session.cookie_secure = 1

bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path [,
string $domain [, bool $secure = false [, bool $httponly = false ]]]]]] )

setcookie("cookie_name", "cookie value", 0, "/", $_SERVER['HTTP_HOST'], true, true);

session_set_cookie_params(0, '/', $_SERVER['HTTP_HOST'], true, true);
```

https://www.eff.org/https-everywhere/deploying-https
https://github.com/ioerror/duraconf/
https://github.com/cloudflare/sslconfig/

PHP hardening . . .

```
/etc/php5/apache2/php.ini:

    Add exec, system, shell_exec, and passthru to disable_functions.

    Change expose_php to Off.

    Ensure that display_errors, track_errors and html_errors are Off.
```

Hide identifying information from PHP . . .

WordPress: add to wp-config.php
Drupal: add to sites/default/settings.php

```
$_SERVER['HTTP_REFERER'] = 'https://remote.server.name/';

$_SERVER['HTTP_USER_AGENT'] = 'Generic web browser';

$_SERVER['REMOTE_ADDR'] = '127.0.0.1';

$_SERVER['REMOTE_HOST'] = 'localhost'
```

Security settings & headers . . .

```
# prevents drive-by downloads
Header set X-Content-Type-Options: nosniff
# prevents cross-site scripting attacks
Header set X-XSS-Protection: "1; mode=block"
# prevents clickjacking
Header set X-Frame-Options: "sameorigin"

Header always append X-Frame-Options: DENY
# prevents XSS, etc.
Header set X-Content-Security-Policy: "default-src 'self'"

Header set X-Download-Options: noopen

Header set X-Permitted-Cross-Domain-Policies: master-only
```

```
/etc/apache2/conf.d/security:

     ServerSignature Off

     ServerTokens Prod

     TraceEnable Off

# Etag is optional and assists tracking

     Header unset Etag

     FileETag None
```

https://www.owasp.org/index.php/List_of_useful_HTTP_headers

Keep out unwanted visitors . . .

auth_basic

```
# it's all password-protected
<Directory "/var/www/htdocs/">
    AuthName "Authentication Required"
    AuthType Basic
    AuthUserFile /var/www/.htpasswd
    Require valid-user
    AllowOverride All
</Directory>
```

Two-factor authentication

```
GoogleAuthUserPath /usr/local/apache2/gauth
GoogleAuthCookieLife 7200
GoogleAuthEntryWindow 5
```

https://code.google.com/p/google-authenticator-apache-module/

Use plain HTML links as social media share buttons. . .

```
https://twitter.com/intent/tweet/?text=TITLE%20URL

https://www.facebook.com/sharer/sharer.php?u=URL&t=TITLE

https://plus.google.com/share?url=URL

https://www.linkedin.com/shareArticle?mini=true&url=URL&title=TITLE

http://tumblr.com/share/link/?name=TITLE&url=URL

https://www.reddit.com/submit?url=URL
```

Use Ghostery, Lightbeam or EFF's Privacy Badger
to see the trackers embedded in each page.

Don't use Twitter's widget . . .

```
require_once('./TwitterAPIExchange.php');

$settings = array(
  'consumer_key' => '',
  'consumer_secret' => '',
  'oauth_access_token' => '',
  'oauth_access_token_secret' => ''
);

$url = 'https://api.twitter.com/1.1/statuses/user_timeline.json';
$getfield = '?screen_name=NERDSummit&count=10';
$requestMethod = 'GET';

$twitter = new TwitterAPIExchange($settings);
$tweets_json = $twitter->setGetfield($getfield)
           ->buildOauth($url, $requestMethod)
           ->performRequest();
```

Apache modules!

modsecurity

ModSecurity is an open source, cross-platform web application firewall (WAF) module.

* Real-time application security monitoring and access control
* Virtual patching
* Full HTTP traffic logging
* Continuous passive security assessment
* Web application hardening

mod_evasive

mod_evasive is an evasive maneuvers module for Apache to provide evasive action in the event of an HTTP DoS or DDoS attack or brute force attack. It is also designed to be a detection and network management tool, and can be easily configured to talk to ipchains, firewalls, routers, and etcetera. mod_evasive presently reports abuses via email and syslog facilities.

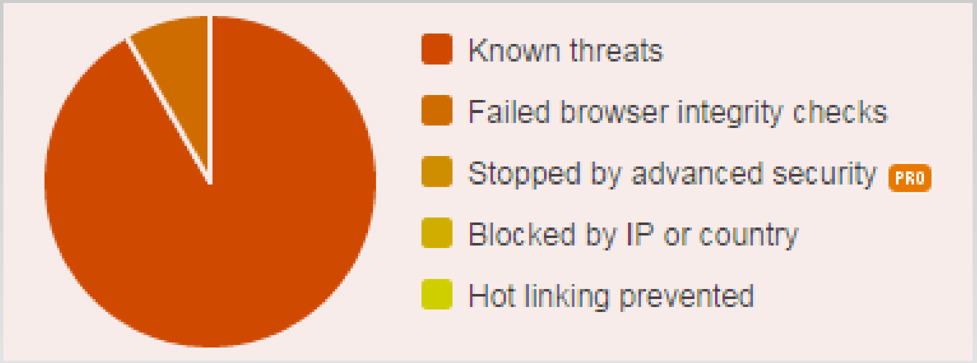
OWASP ModSecurity Core Rule Set (CRS):
https://github.com/SpiderLabs/owasp-modsecurity-crs

CloudFlare's CDN is your friend

Browser integrity
Automatically performs a browser integrity check for all requests to your website by evaluating the HTTP headers for threat signatures. If a threat signature is found, the request will be denied.

Visitor reputation
CloudFlare uses threat data from a variety of sources to build a reputation for every visitor online. You set the desired security setting for your site and then CloudFlare⁹s network stops the threats before it reaches your website. Reputation-based security provides a first line of defense for your website.

Block list / trust list
In addition to CloudFlare⁹s automatic detection, you can easily add an IP address, IP ranges or entire countries to your Trust and Block list.



Saved bandwidth and server resources
By stopping threats before they get to your website you save bandwidth and resources. Your server is also freed up to serve your legitimate traffic optimally.

Protect SSH / Telnet / FTP ports
Add a layer of protection to ports like SSH, FTP and Telnet by disabling them for your root domain. Continue to access them from a subdomain of your choosing.

Collaborative security
CloudFlare uses the collective intelligence of its community to get smarter. CloudFlare⁹s network learns from every new attack and then shares that information with the rest of the CloudFlare community.

Breaking the cycle of malware
Websites are empowered to inform visitors with compromised computers so these visitors can take action to clean the malware infection.

Source: https://www.cloudflare.com/features-security

Securing SSH

denyhosts

Utility to help sys admins thwart SSH crackers

DenyHosts is a program that automatically blocks SSH brute-force attacks by adding entries to /etc/hosts.deny. It will also inform Linux administrators about offending hosts, attacked users and suspicious logins.

sshguard

Protects from brute force attacks against ssh

Protects networked hosts from the today's widespread brute force attacks against ssh servers. It detects such Attacks and blocks the author's address with a firewall rule.

```
/etc/ssh/sshd_config:

# root login with key only
PermitRootLogin without-password

# non-default port
Port 23
```

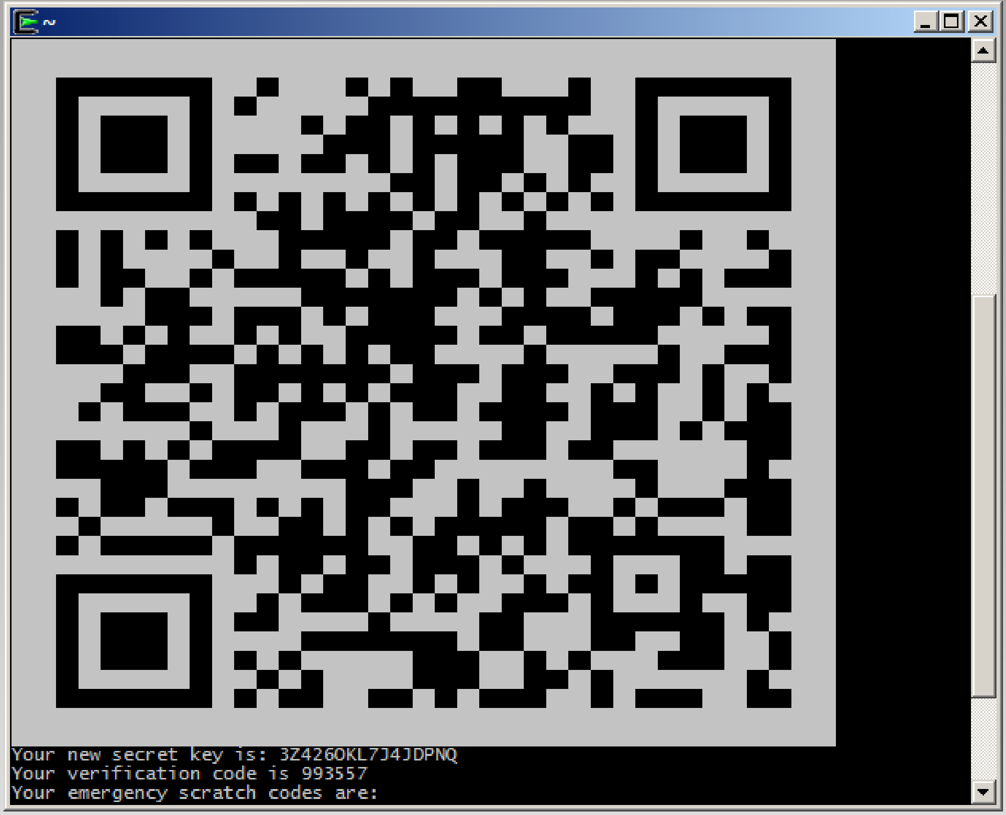
Fail2ban

ban hosts that cause multiple authentication errors

Fail2ban monitors log files (e.g. /var/log/auth.log, /var/log/apache/access.log) and temporarily or persistently bans failure-prone addresses by updating existing firewall rules. Fail2ban allows easy specification of different actions to be taken such as to ban an IP using iptables or hostsdeny rules, or simply to send a notification email.

Two-factor auth

```
$ apt-get install libpam-google-authenticator

/etc/pam.d/common-auth:

     auth required pam_google_authenticator.so

$ google-authenticator
```



```
Your new secret key is: 3Z426OKL7J4JDPNQ
Your verification code is 993557
Your emergency scratch codes are:
```

https://code.google.com/p/google-authenticator/

Firewalls

Use nmap to see what ports are open.

Block null packets:
```
    $ iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

Reject syn-flood:
```
    $ iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
```

Reject recon packets:
```
    $ iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
```

Allow HTTP, HTTPS and SSH from your IP only:
```
    $ iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
    $ iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
    $ iptables -A INPUT -p tcp -s YOUR_IP_ADDRESS -m tcp --dport 22 -j ACCEPT
```

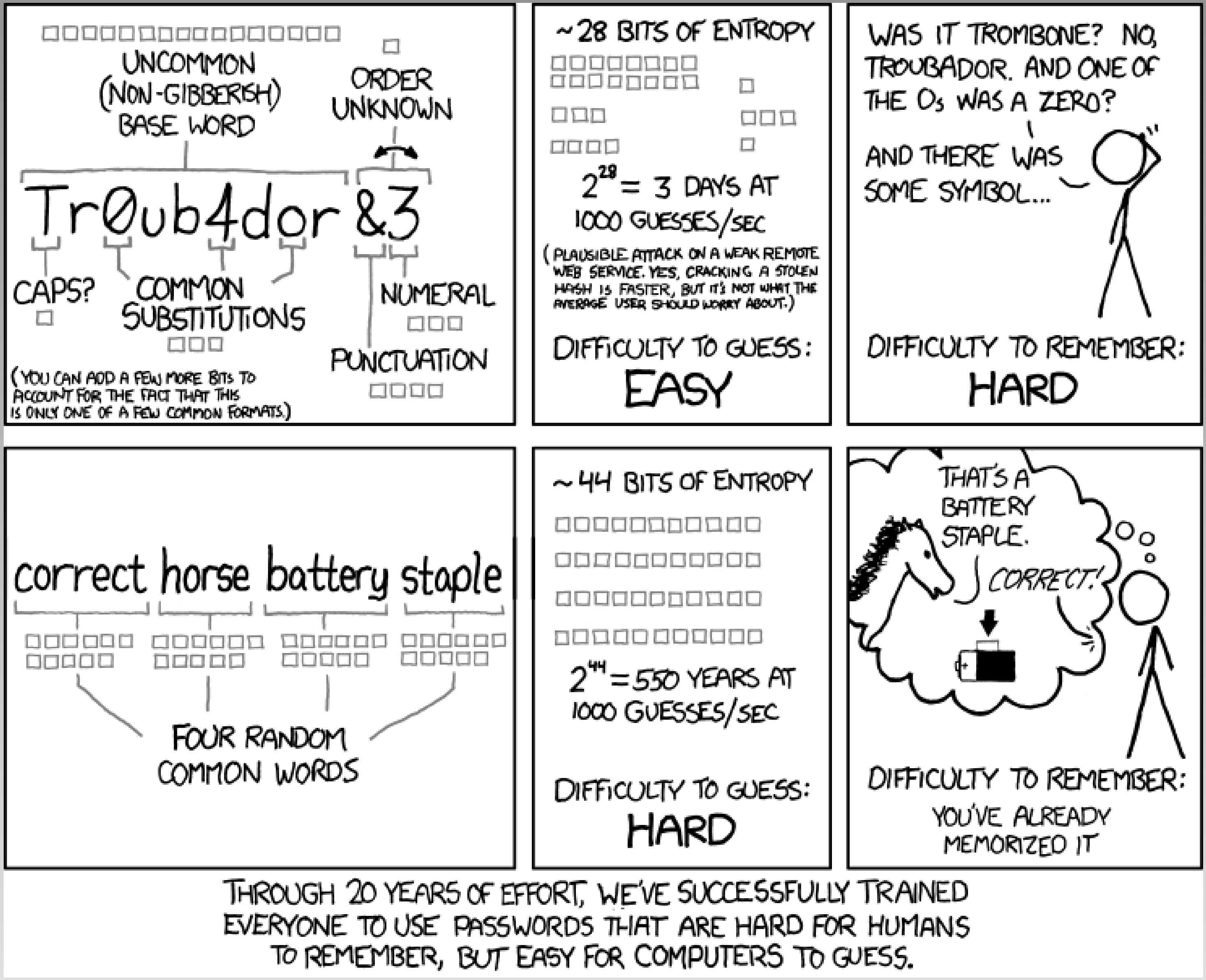
Allow outgoing connections and block everything else:
```
    $ iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
    $ iptables -P OUTPUT ACCEPT
    $ iptables -P INPUT DROP
```

ufw: Uncomplicated FireWall
```
    $ ufw allow 80/tcp
    $ ufw allow 443/tcp
    $ ufw allow 22/tcp
```

Network hardening . . .

```
/etc/sysctl.conf:

# Ignore ICMP broadcast requests
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Disable source packet routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0

# Ignore send redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Block SYN attacks
net.ipv4.tcp_max_syn_backlog = 2048
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_syn_retries = 5

# Log martians
net.ipv4.conf.all.log_martians = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Ignore ICMP redirects
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0

# Ignore directed pings
net.ipv4.icmp_echo_ignore_all = 1
```

Complex passwords . . .



UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Source: xkcd.com

Quick reminders . . .

Input sanitization: escape inputs to the shell or MySQL

```
string escapeshellcmd ( string $command )

string mysql_real_escape_string ( string $unescaped_string [, resource
$link_identifier = NULL ] )
```

Hash and salt passwords, don't store them in the clear!

```
string password_hash ( string $password , integer $algo [, array
$options ] )
```

Please DON'T use the root MySQL database user on your site!

Linux security extras

OSSEC is an Open Source Host-based Intrusion Detection System that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response.

http://www.ossec.net/

Grsecurity is an extensive security enhancement to the Linux kernel that defends against a wide range of security threats through intelligent access control, memory corruption-based exploit prevention, and a host of other system hardening that generally require no configuration.

https://grsecurity.net/

SELinux is a Linux kernel security module that provides the mechanism for supporting access control security policies

rkhunter scans systems for known and unknown rootkits, backdoors, sniffers and exploits.

chkrootkit searches the local system for signs that it is infected with a rootkit.

Penetration testing!

Metasploit

http://www.metasploit.com

w3af

http://w3af.org

Acunetix

https://www.acunetix.com

sqlmap

http://sqlmap.org

fierce.pl

http://ha.ckers.org/fierce/

Miscellaneous

Update early, update often!

Bind daemons to localhost

Developers: encrypt your local machines, scan for viruses and malware

See: Privacy Tricks for Activist Web Developers, by Micah Lee
https://www.eff.org/files/filenode/hope_privacy_tricks.pdf

MySQL with SSL:
https://dev.mysql.com/doc/refman/5.5/en/ssl-connections.html

Verify signatures and hashes of downloaded software, compile from source

Copy web fonts, JavaScript to your server

Host your own analytics using Piwik:
http://piwik.org/

Free SSL certificates (as long as it's not for profit):
https://www.startssl.com/?app=1
https://github.com/ioerror/duraconf/blob/master/startssl/README.markdown

Use OTR, PGP to communicate with clients, exchange credentials

If you want you can use FDE (full disk encryption) on dedicated servers, but it's also important to *secure-delete* and zero the disks when it's retired

Defense in depth, security through obscurity. Make sites Tor-friendly.