

Security holes...Who cares?

Eric Rescorla

RTFM, Inc.

<http://www.rtfm.com/>

Preview of presentation

◆ Longitudinal study of administrator response

In response to an announced vulnerability

OpenSSL buffer overflows of July 2002

◆ Main findings

Many people don't deploy fixes

Most fixing happens almost immediately

With a second round after the worm was released

Some weak predictors

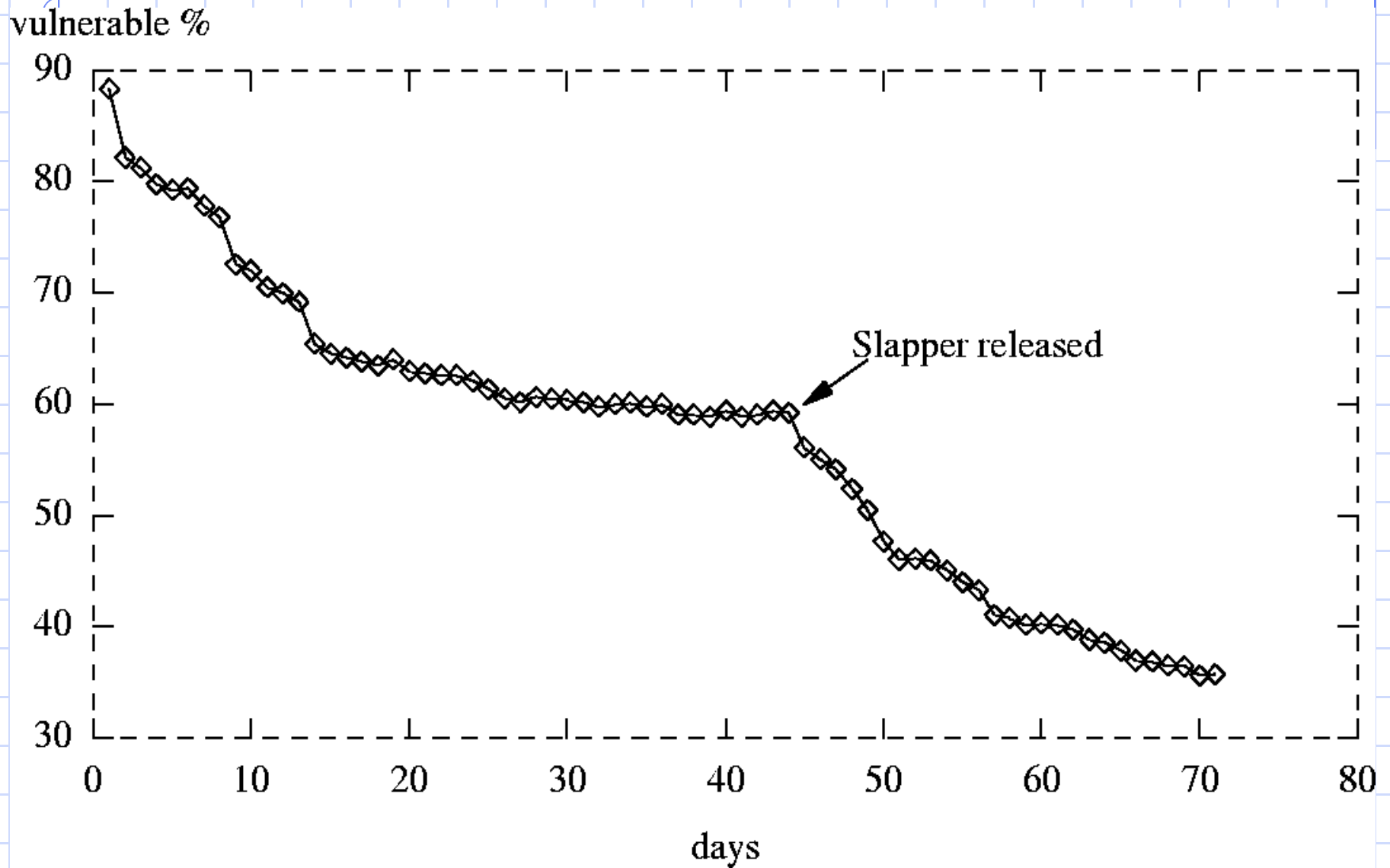
HSPs are more responsive

Current software version

OpenSSL

Apache

Fix deployment by time



Background

- ◆ Widely believed that users don't upgrade

- ◆ Anecdotal evidence

 - A lot of malware exploits "fixed" bugs

 - Lots of old versions of IE floating around

 - Netcraft: Many Apache users haven't upgraded

- ◆ Little hard data

 - Bellovin, Provos -- measured version number

 - Moore -- measured response to Code Red

Overview of the bugs

- ◆ Announced July 30, 2002 by Ben Laurie

- ◆ Buffer overflows in OpenSSL

 - Allowed remote code execution

- ◆ Affected software

 - Any OpenSSL-based server which supports SSLv2

 - Essentially everyone leaves SSLv2 on

 - mod_SSL, ApacheSSL, Sendmail/TLS, ...

 - Easy to identify such servers

 - Any SSL client that uses OpenSSL

OpenSSL flaws: a good case study

- ◆ A serious bug

 - Remotely exploitable buffer overflow

- ◆ Affects a security package

 - Crypto people care about security, right?

- ◆ In a server

 - Administrators are smarter, right?

- ◆ Remotely detectable

 - .easy to study

Questions we want to ask

◆ What fraction of users deploy fixes?

And on what timescale?

◆ What kind of countermeasures are used?

Patches

Available for all major versions

Often supplied by vendors

Upgrades

Workarounds

Turn off SSLv2

◆ What factors predict user behavior?

Methodology

- ◆ Collect a sample of servers that use OpenSSL

 - Google searches on random words

 - Filter on servers that advertise OpenSSL

 - This means `mod_ssl`

 - n=892

 - (890 after complaints)

- ◆ Periodically monitor them for vulnerability

Detecting Vulnerability

- ◆ Take advantage of the SSLv2 flaw

 - Buffer overflow in `key_arg`

- ◆ Negotiate SSLv2

- ◆ Use an overlong `key_arg`

 - The overflow damages the next struct field

 - `client_master_key_length`

 - `client_master_key_length` is written before it is read

 - So this is safe

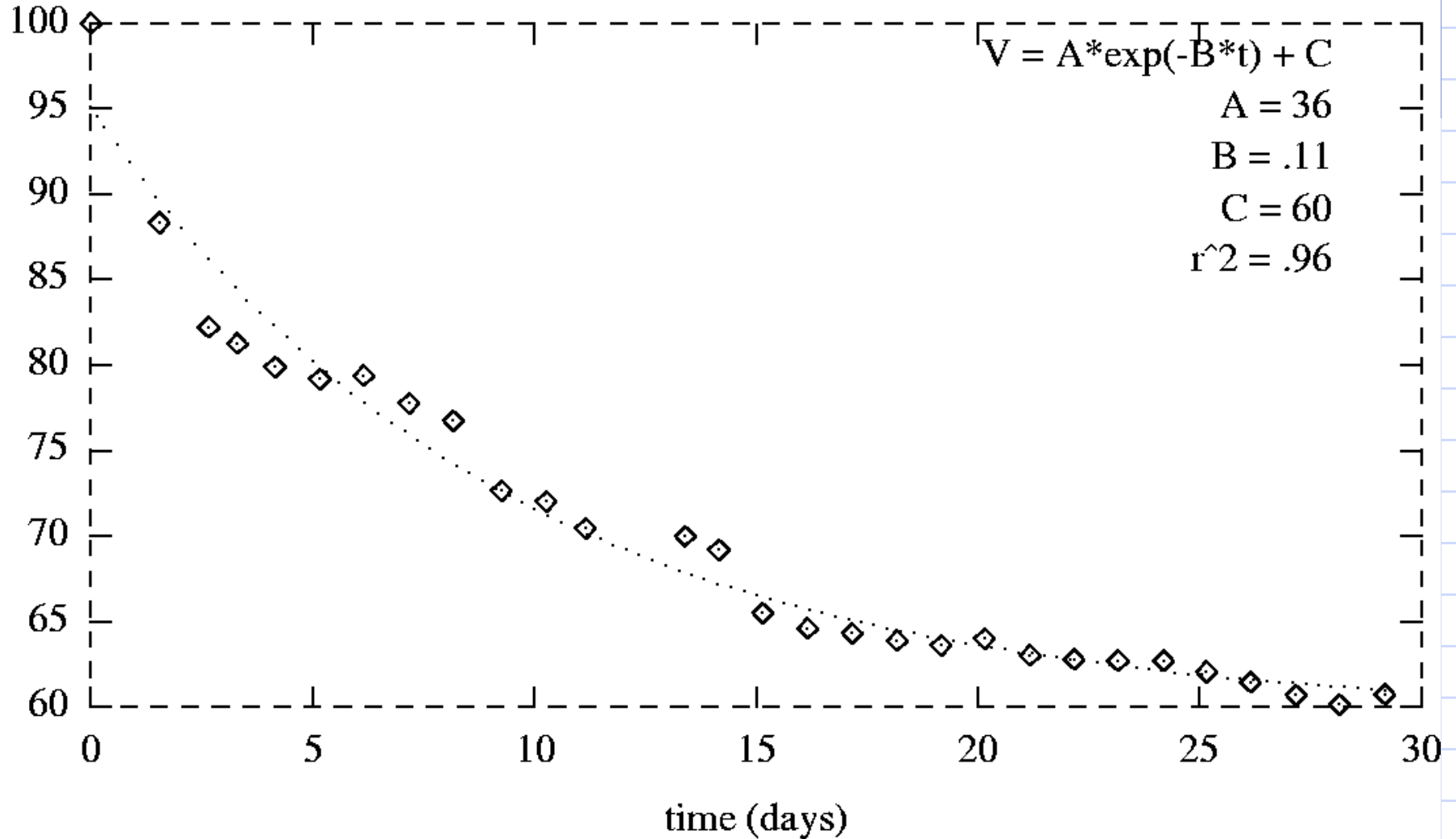
- ◆ This probe is harmless but diagnostic

 - Fixed implementations throw an error

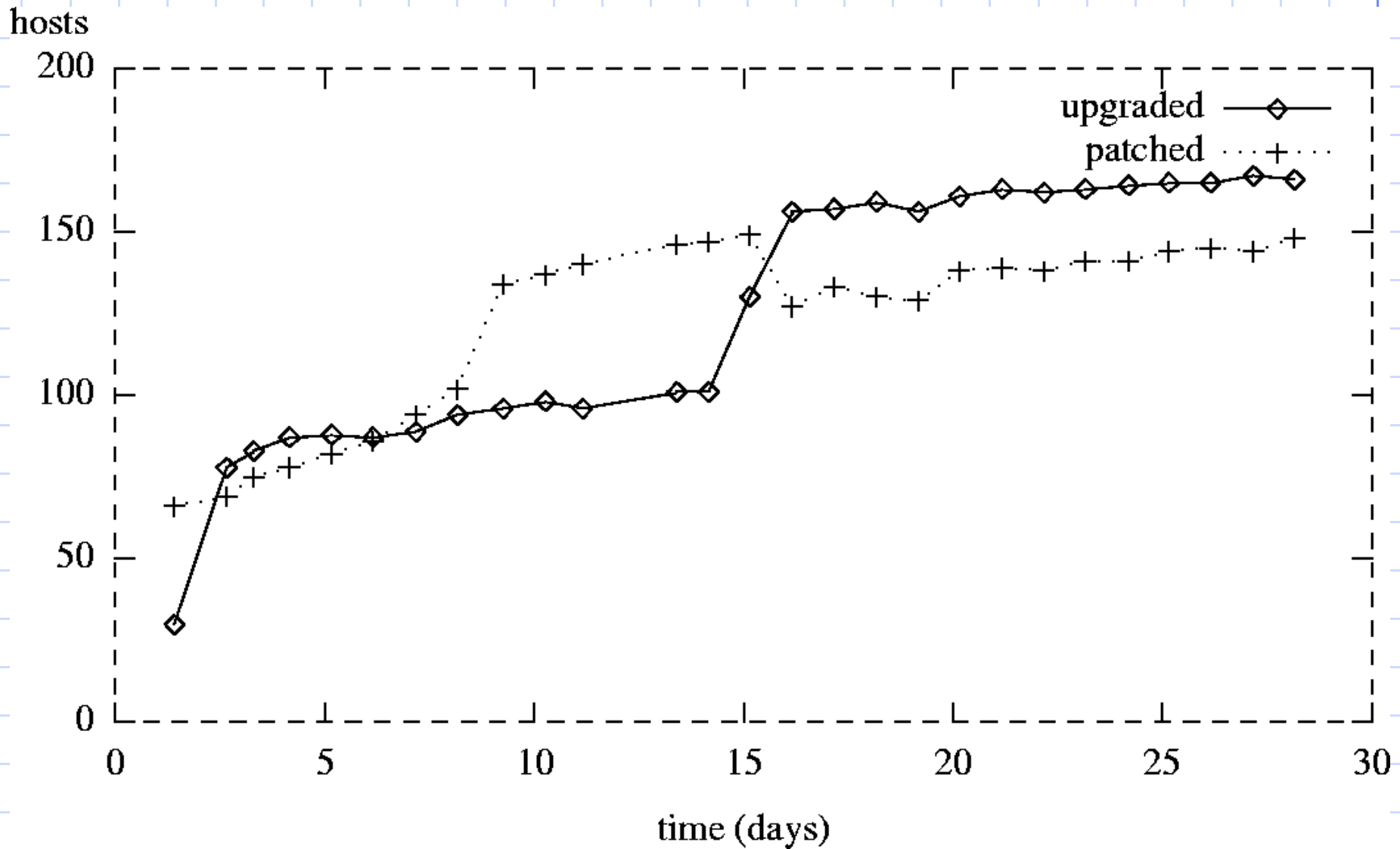
 - Broken implementations complete handshake

Response after bug release

vulnerable %



Kinds of fixes deployed



Why not use workarounds?

◆ Disabling SSLv2 is complete protection

It's easy

But essentially no administrator did it

Never more than 8 machines

Why not?

◆ Guesses...

Advisories unclear

Not all described SSLv2-disabling as a workaround

Some suggested that all OpenSSL-using applications were unsafe

It's fine if you just use it for crypto (OpenSSH, etc.)

Pretty easy to install patches

Anyone smart enough just used fixes

Potential predictors of responsiveness

- ◆ Who operates the server

 - HSPs

 - TLD

- ◆ Current software version

 - OpenSSL

 - Apache

- ◆ “Liveness”

 - Last modified date

 - Certificate validity

Is this an HSP?

- ◆ Use WHOIS to identify them

 - Assume that servers in the same netblk are an HSP
 - Recursively follow netblks assigned to RIRs

- ◆ This isn't perfect...

 - Sometimes blocks aren't delegated

 - Sometimes HSPs own blocks with different names

HSPs are more responsive

HSP Size (hosts)	% Vulnerable	# Hosts
1-4	0.71	673
5-15	0.50	46
15-30	.33	69
30-100 (Verio)	.15	69

- ◆ HSP servers behave differently from individuals

 - Attempts to fit them together don't work

- ◆ We need to split the dataset

 - And fit them separately

 - Breakpoint is somewhat arbitrary (we choose 1)

Version effects (logistic regression)

- ◆ Up to date versions are predictive for non-HSP servers
 - Significantly more likely to apply fixes
 - No real difference within OpenSSL 0.9.6
- ◆ No significant effect for HSPs

Predictor	Odds Ratio	95% CI	P
OpenSSL 0.9.6x (x<d)	2.82	1.42-5.60	0.003
OpenSSL 0.9.6d	4.44	1.86-10.6	0.001
Apache Up-to-date	2.39	1.48-3.86	0.000

Some other factors

◆ TLD

Reported by Moore et al.

A little tricky (lots of empty cells)

No effect seen (Chi-square $P=0.914$)

Some possibility that .edu is different in stratified model

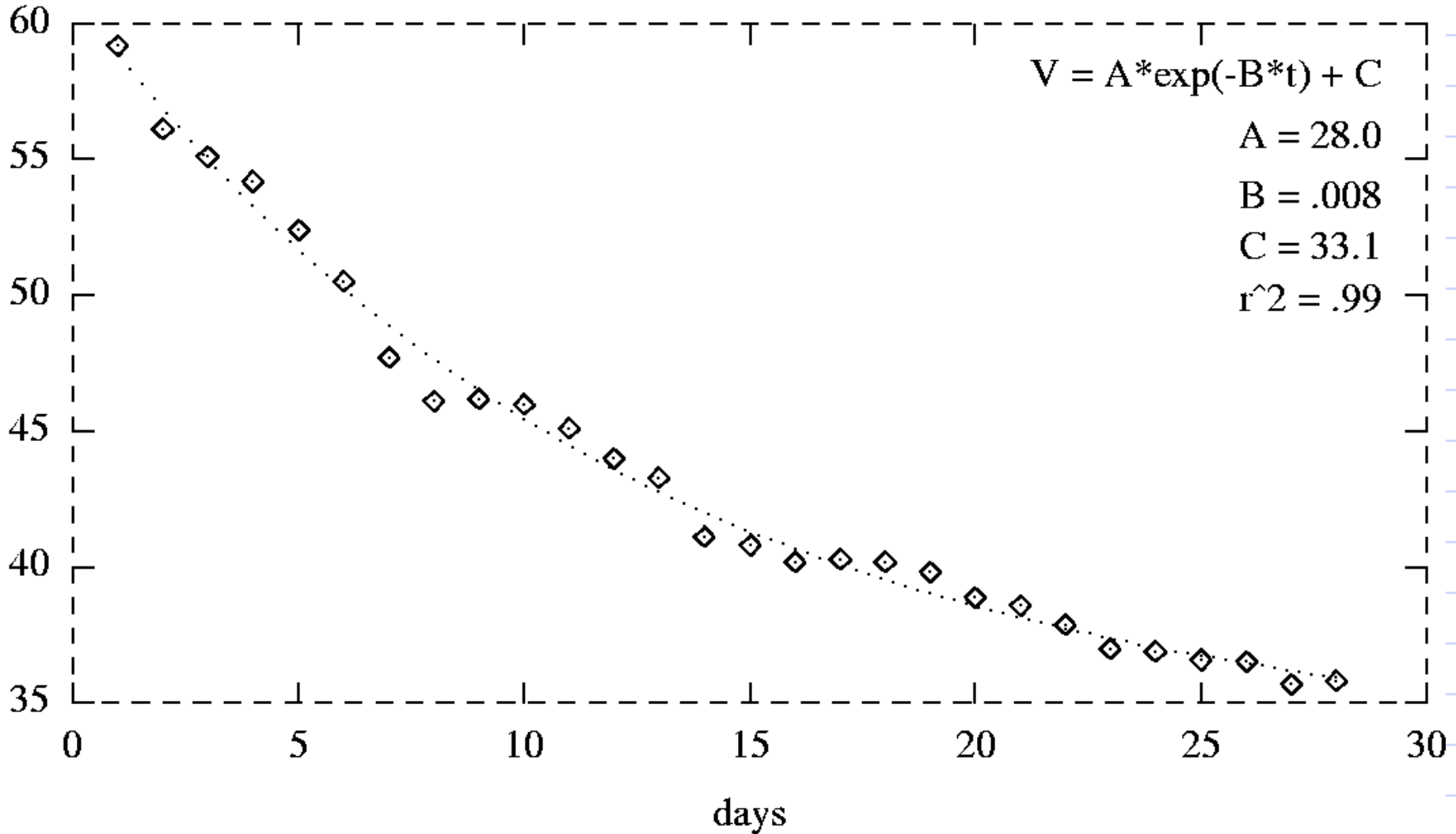
◆ Certificate status

May be an effect with independent servers

◆ “Last-modified” not relevant at all

Response after Slapper release

vulnerable %



Why so much post-worm response?

◆ People didn't hear the first time?

Not likely..published through the same channels

◆ Guesses

People are interrupt driven

People respond when they feel threatened

And deliberately ignore potential threats

The zombie problem

- ◆ 60% of servers were vulnerable at worm release
 - These servers can be turned into zombies
 - ..and then DDoS other machines
- ◆ Independent servers are less responsive
 - So they're harder to turn off
 - Try contacting hundreds of administrators
- ◆ Slapper wasn't so bad
 - Since Linux/Apache isn't a monoculture
 - And the worm was kind of clumsy

Policy Implications

- ◆ The window of vulnerability is really long
But the marginal window is short
- ◆ Don' t delay full disclosure by > 1 month
Everyone who is going to upgrade already has
- ◆ Full disclosure before fixes is bad
Marginal cost to attentive admins is very high
- ◆ How can we get people to upgrade?
Fine them?
Pay them?

Conclusions

- ◆ The situation is not good

 - A lot of machines are vulnerable

- ◆ Response to security bugs is bimodal

 - About a third of admins upgrade after the advisory

 - Another third after a worm is released

 - The rest not at all

- ◆ We need more research on why people do or do not fix

 - And how to motivate them to do so

More information

◆ <http://www.rtfm.com/upgrade.html>