

DRAFT

Optimal Time to Patch Revisited

Eric Rescorla
RTFM, Inc.
ekr@rtfm.com

1 Introduction

Whenever a vulnerability in a piece of system software is identified and a patch is released, administrators have to decide whether or not to apply the patch to their systems. In 2002, Beattie et al.[1] considered the question of the optimal waiting time to apply such a patch. This note revisits some aspects of that analysis.

2 Review of Prior Work

The paper under discussion consists of two separate parts. The first is a theoretical Bayesian decision analysis of the value of applying a given patch at time t . The second part is an attempt to empirically determine one of the parameters of the theoretical model. When doing econometric analysis of this type, there's always a danger of modelling only the parameters that you can easily measure, and some sensitivity analysis suggests that that may be the case here.

2.1 The Model

Beattie et al. apply standard cost/benefit analysis to the question of the expected value of applying a patch at time t . The cost function for applying a patch is given by

$$e_{patch}(t) = P_{fail}(t) \cdot e_{p.recover} + e_{patch} \quad (1)$$

- $P_{fail}(t)$ is the probability that a patch applied at time t will subsequently need to be reversed.
- $e_{p.recover}$ is the cost to recover from a failed patch.
- e_{patch} is the cost to apply a patch

The cost function for failing to apply a patch is given by

$$e_{nopatch}(t) = P_{breach} \cdot e_{breach} \quad (2)$$

where

- P_{breach} is the probability that the system will have been breached by time t .
- e_{breach} is the cost of a breach

In order to simplify the analysis, the authors make a number of assumptions.

Applying patches is free. The authors argue that the use of automated update tools makes the fixed cost of

applying a patch close to zero and therefore they assume that $e_{patch} = 0$.

The cost of breach is proportional to cost of recovery. Lacking a good model for the cost of recovery or the cost of breach, the authors assume that

$$e_{breach} = C \cdot e_{p.recover} \quad (3)$$

where C is an unspecified constant.

In this model, agents are assumed to be risk neutral and so the indifference point can be found by setting equation (1) equal to equation (2) and then solving for $t_{indifference}$. As equation (1) decreases monotonically, patching dominates not patching at any time $t > t_{indifference}$.

2.2 Empirical Results

Under the two simplifying assumptions previously specified, there are only three free variables:

- $P_{fail}(t)$
- $P_{breach}(t)$
- C (the cost proportionality constant)

The authors do not attempt to measure P_{breach} or C but do attempt to measure P_{fail} . The general approach is to examine a series of patches issued for various pieces of software and determine what fraction are recalled at any point in time. It is assumed that patches which have not been recalled at the end of the study period are never recalled. Figure 1 shows the number of issues which were resolved by time since patch release. (Note: data for this analysis were obtained by extracting them from the graphs in the original paper).

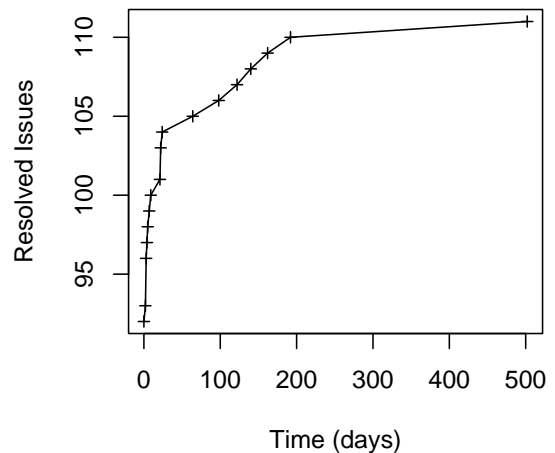


Figure 1 Resolved issues by time

Note that $< 20\%$ of the patches are ever recalled, meaning that over 80% of patches are presumably correct as initially issued. Thus, $P_{fail}(0) = .18$.

3 Problems with the Analysis

3.1 Assuming that patching is free

The authors argue that the cost to apply a patch is sufficiently small that it can be ignored. This assumption seems unwarranted. While it's true that some systems provide automated updating tools, others do not. In particular, changes to BSD systems often require system recompiles. Moreover, package updates do not always work smoothly. Finally, in many large installations, administrators deploy the patch first on sacrificial testbed machines before updating production machines. This consumes substantial resources.

Additionally, no data is currently available on the cost of recovery from bad patches ($e_{p.recover}$). The authors make some qualitative arguments that $e_{p.recover}$ is potentially high but don't appear to have examined the patches under study in order to determine what the likely In many cases, this cost will be small, simply the cost of installing the replacement patch, thus making $e_{p.recover} = e_{patch}$. Under these conditions, equation (1) is dominated by e_{patch} and therefore it is unsafe to set it to zero.

3.2 Boundary conditions

Beattie et al.'s Figure 1 shows the traditional pair of crossing cost curves. However, we need to consider the possibility that the cost curves do not cross. This can occur in two ways. First, consider what happens if the cost to patch is very high (such as when it requires recompiling the kernel) and the cost of penetration is very low, such as when the vulnerability is local or not very serious. In this case, the cost curves will never cross, as shown in Figure 2 and therefore it's never rational to patch. Recall that Beattie et al. assumed that $e_{patch} = 0$ and so did not consider this case—if $e_{patch} = 0$ then $e_{patch}(\infty) = 0$ and there must be a point at which the curves cross.

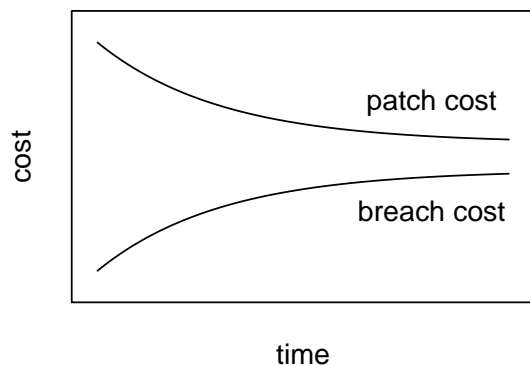


Figure 2 Patching too expensive

The second boundary condition occurs when the cost of patching is low by comparison to the expected cost of breach. This situation occurs when the seriousness of the vulnerability and the cost of breach are relatively high. In this case, $e_{patch}(t) < e_{breach}(t)$ for all values of t and it is rational to patch immediately, as shown in Figure 3.

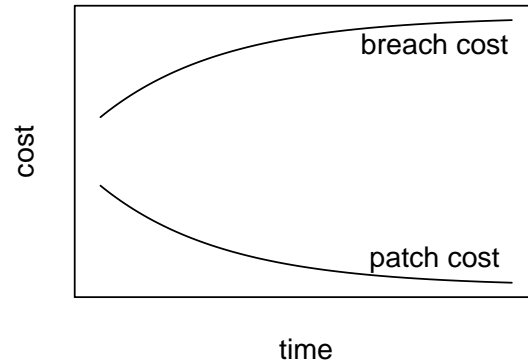


Figure 3 Patching dominates not patching

It's worth exploring briefly the conditions under which this analysis obtains. Recall that under the original analysis we assumed that $e_{patch} = 0$. Assume without loss of generality that $e_{p.recover} = 1$. In this case, $e_{p.recover}(t) \leq .18$ and $e_{patch}(t) < e_{nopatch}(t)$ provided that:

$$C \cdot e_{nopatch}(0) > .18 \quad (4)$$

How likely is this to occur? That largely depends on the availability of a day zero exploit. If we assume conservatively that $C = 5$ (that is that recovering from a penetration is 5x more expensive patch recover) than equation (4) is fulfilled as long as $P_{breach} > .05$. This number does not seem unreasonably high for a high profile site.

3.3 Timing Recommendations

Since data on P_{breach} is unavailable, Beattie et al. rely on the apparent inflection points at days 10 and 30 of the issue resolution curve. They state:

However, since $P_{breach}(t)$ is difficult to compute, the pragmatist may want to observe the knees in the curve depicted in Figure 7 and 9 and apply patches at either ten or thirty days

There are two problems with this approach. First, due to the highly restricted range of the $e_{p.recover}$ curve, the appropriate time to patch depends largely on the shape of the $e_{nopatch}$ curve, which is unknown. Second, there is insufficient data upon which to draw the conclusion that there are knees in the curve at days 10 and 30.

3.3.1 Dependency on cost of breach

As the previous discussion illustrates, $t_{indifference}$ depends primarily on $e_{nopatch}$. The reason for this is simply that the range of e_{patch} is very small because $P_{fail}(0) = .18$. Even if we assume conservatively that the maximum value of $e_{breach}(t)$ is equal to $e_{p.recover}$, the range of $e_{breach}(t)$ is 5x that of $e_{p.recover}$.

In order to illustrate this point, we now perform a sensitivity analysis for $t_{indifference}$ based on C and $P_{breach}(t)$. As before, we retain the original assumption that $e_{patch} = 0$ and set $e_{p.recover} = 1$. Following Browne [2], we assume that $P_{breach}(t)$ is linear with the \sqrt{t} :

$$P_{breach}(t) = I + S\sqrt{t} \quad (5)$$

Merging equations (2), (3), and (5) we get

$$e_{breach}(t) = C(I + S\sqrt{t}) \quad (6)$$

Figure 4 shows the value of $t_{indifference}$ under various values of C , and S ($I = 0$). For reference, the final column shows the fraction of hosts which have been penetrated at 1 year.

C	S	Days	Cost (indifference)	Pbreach (1 year)
1	0.001	178	0.013	0.019
1	0.005	90	0.047	0.095
1	0.01	37	0.06	0.191
1	0.03	11	0.099	0.573
2	0.001	147	0.024	0.019
2	0.005	37	0.06	0.095
2	0.01	21	0.091	0.191
2	0.03	5	0.134	0.573
10	0.001	37	0.06	0.019
10	0.005	6	0.122	0.095
10	0.01	3	0.173	0.191
10	0.03	1	0.3	0.573

Figure 4 Sensitivity analysis for indifference point

As can be seen from Figure 4, the optimal time to patch depends greatly on the probability of a breach and the cost of a breach. The reason for this variation is that the patch failure curve is very flat, as can be easily seen in Figure 5, which shows the cost curves for four sample values of C and S .

Note that the values we are exploring are quite conservative. The three lowest curves all assume that the cost of penetration is equal to the cost of a broken patch, and even the fastest growing of the three has a much lower penetration rate than Code Red [3]. Similarly, it's easy to imagine an attack which destroyed all data on your machine. This could easily be 10 times as expensive as a simple patch rollback. The top cost curve represents that case.

The result of all this is that the P_{fail} curve is of marginal utility for predicting the optimal time to patch. The shape and speed of the e_{breach} curve is much more important.

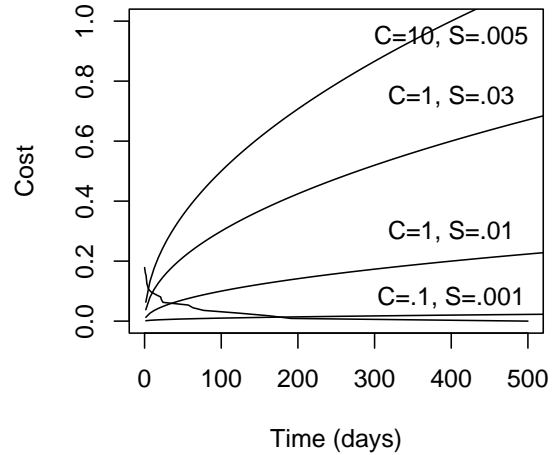


Figure 5 Some sample values for cost of breach

3.3.2 Inflection Points

In addition, there is insufficient data to draw conclusions about knees in the curve. The points in question deviate from the implied smooth curve by only one or two patches, which is well within the margin of error.

This is easily seen by examining Figure 6, which shows the Kaplan-Meier[4] survival curves for time to issue resolution.¹ The surrounding curves show the 95% confidence interval for the survival function, computed using the Greenwood formula. As is evident, the deviations that were interpreted as knees in the curve are small compared to the confidence interval. Therefore, there is insufficient data to conclude that days 10 and 30 represent knees in the curve.

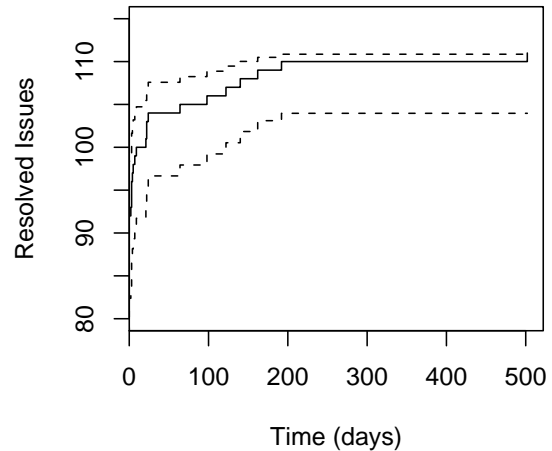


Figure 6 Kaplan-Meier Plot for Issue Resolution

1. Kaplan-Meier estimates and confidence intervals were performed using R [5].

Note that we have followed Beattie et al.'s lead in treating issues which did not result in patch recalls as "resolved" rather than right censored. However, treating the event as patch recall rather than issue resolution and therefore the patches which were not recalled as censored does not materially change the analysis.

References

- [1] Beattie, S., Arnold, S., Cowan, C., Wagle, C., Wright, C., and Shostack, A., "Timing the Application of Security Patches for Optimal Uptime," *Proceedings of LISA XVI* (2002).
- [2] Browne, H. K., Arbaugh, W. A., McHugh, J., and Fithen, W. L., "A Trend Analysis of Exploitations," *University of Maryland and CMU Technical Reports* (2000).
- [3] Moore, D., Shannon, C., and Claffy, K., "Code-Red: a case study on the spread and victims of an Internet worm," *Internet Measurement Workshop* (2002).
- [4] Kleinbaum, D., *Survival Analysis: A Self-Learning Text*, Springer, New York (1996).
- [5] Ihaka, R., and Gentleman, R., "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics*.