Measuring security with SecQua

Metricon 7.0-USENIX 2012

Constantinos Patsakis
Department of Computer Engineering and Maths
Universitat Rovira i Virgili
UNESCO Chair in Data Privacy
Tarragona, Catalonia

Contents

Introduction

Security metrics

A difficult and "itchy" topic set in the heart of Security. There is no straight answer from everyone. Several times we can say that we are more secure, but quantifying it remains we can say an open question, as there in not a widely accepted answer.

Why do we need quantification

- "Quantifying means identifying"
- "You cannot manage what you cannot measure"
- Take decisions
- Manage resources

Numbers can be misleading

- # of incidents,viruses/spam/attacks blocked
- "We block 95% of the launched attacks!"
  How powerfull is the rest 5%?
- "Our software has few vulnerabilities."
  What do they disclose?
- Statistics, the best way to tell a lie!

Measuring security

According to Geer, by measuring security, one must be able to answer to the following questions [1]:

- How secure am I?
- Am I better compared to my last checkpoint/year?
- Am I spending the right amount of money for security?
- How do I compare to my peers?
- What risk transfer options do I have?

What else?

- formal model
- Objective
- Change through time
- Inexpensive
- Obtainable
- Repeatable

Categorizing security metrics

So far several approaches have been proposed. If we would like to categorize them, they fall down to the following categories [2]:

- Standards, guidelines, and best practices research documents that provide processes, frameworks, and meta-models for security quantification.
- Automated tools focused on specific challenges that gather quantifiable data.
- Governmental research efforts that focus on specific aspects of IS security measurement.

Categorizing security metrics (continued)

- Industry research efforts that focus on specific aspects of IS security measurement.
- Data collected with various ways in order to be processed by a Certified Information System Auditor.
- Enumerations and scoring systems.
- Efforts made for producing categorizations or taxonomies.
- Legislative and regulatory directives.

Software as security "stock" (I)

By adding components to an information system, we make an investment. Like all investments it has a certain amount of risk attached to it as well a certain return.
We regard the security of an information system as the return that we have from combining several components. We assume that the security of each component changes everyday, as new vulnerabilities can be disclosed about it, or because of deprecation, in case we talk about physical components.

SOFTWARE AS SECURITY "STOCK" (II)

Each component has different value each day as stocks in stock market. Everything that is installed in an IS, from the hard disk to the firewall and the operating system, when viewed from the eyes of a manager are an economic investment that has to create an interest.
In this case is the increase of the security status.

Software as security "stock" (III)

The security status of an IS is the portfolio of these "stocks". The metric will try to "sum" the "price" of these "stocks". In the IS each component has different vulnerability distribution, has different impact on the overall security and is used for different amount of time, factors that should be taken into consideration.

Vulnerability patterns (I)

Several people believe that Friday 13th is not a "good day".
Does such concept exist in software security?
Systems are overloaded several days, e.g. payment systems at the end of month.
When are vulnerabilities disclosed for the software that I'm using? If for example they are disclosed every Monday, Tuesday is a "bad day"...

Vulnerability patterns (II)

A security metric should be able to point out these patterns. In many cases, it is not the day with most disclosed vulnerabilities the most "dangerous", as they might have minor impact, contrary to other days.

The metric

The core ideas of this metric have been presented in [3, 4] in the case of stochastic integration (part of this work is under review).
Here we illustrate the deterministic way.

Component weights

For each software component i we set the respective weight c_i by the following formula:

$$c_i = -\sum_{j=1}^{n}\sum_{k=1}^{m}(1 + pen \cdot log(1 + dt))t_{ik}e^{dt-k}w_j p_{ijk} log(p_{ijk})$$

Decompiling the weight formula (I)

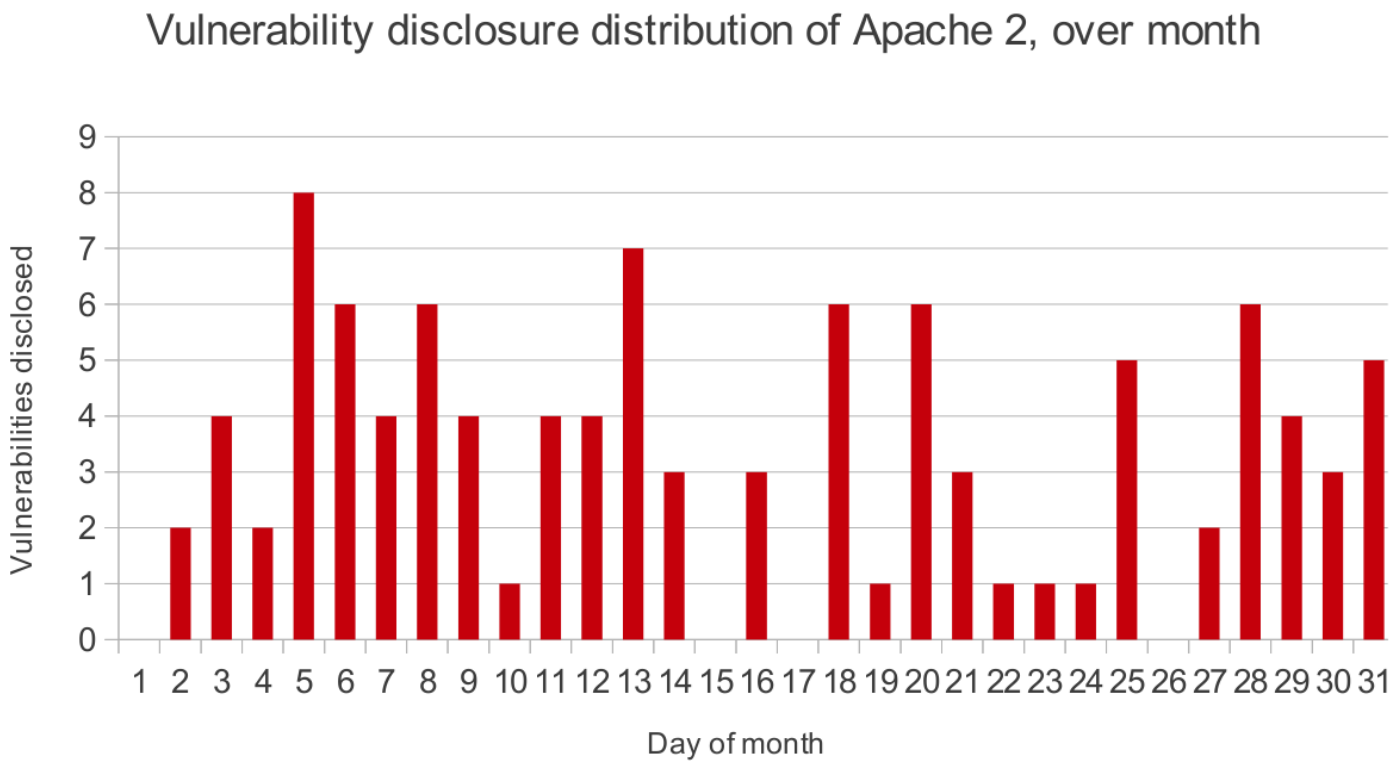
- n, the number of different impacts values.
- m, is the number of years that product i is in the market
- w_j is the weight attributed to each vulnerability impact (SecQua uses the CVSS score).
- pen is a constant declaring the penalty for using a discontinued product , (default installation of SecQua uses $pen = 1$).

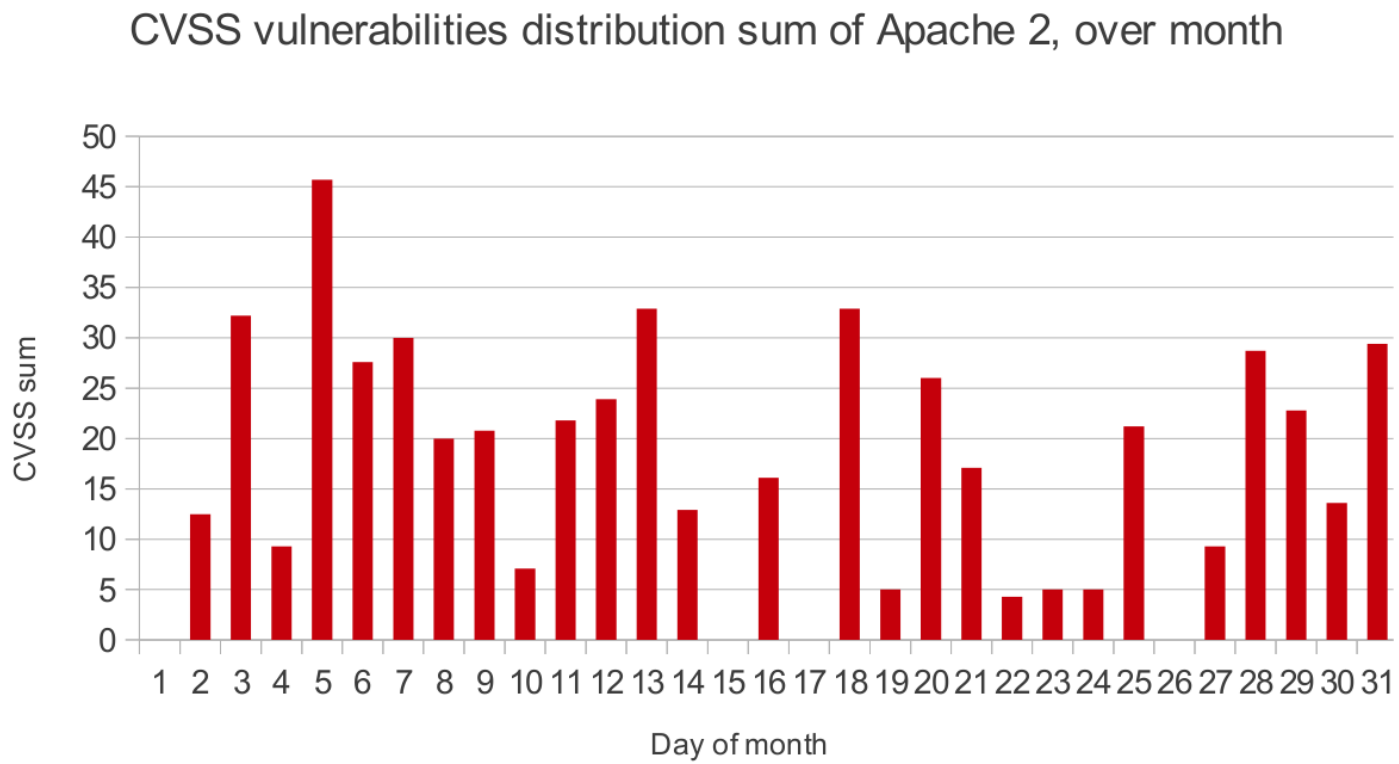
Decompiling the weight formula (II)

- $p_{ijk} = \frac{number\ vulnerabilities\ of\ impact\ j\ in\ year\ k}{total\ number\ vulnerabilities}$ for component i
- t_{ik} represents the percentage use of component i, k years ago.
- dt, the amount of years that the component is discontinued and does not receive updates.

Computing the impact of each component I

Having calculated for each component the respective c_i, we calculate the CVSS vulnerabilities distribution sum for the requested period.


Vulnerability disclosure distribution of Apache 2, over month
Vulnerabilities disclosed
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Day of month



For each day, we sum the CVSS scores.
3rd day of month is equally dangerous as 13th, 18th even if the disclosed vulnerabilities are not that many...

Computing the impact of each component (II)

We divide the daily CVVS sums with the total CVSS sum of each component. The resulting values range from 0 to 1 and sum up to one. For each product we exponentiate the respective values to the appropriate weight $1/c_i$.

Computing the impact of each component (III)

These values show how vulnerable the system is. To show how secure the IS is,we have to transform them, so we subtract them from number one (1=100% the totally secure system).

Putting them all together

We now have the impact of each component to the security of IS, according to the period we have selected (weekday/day/month/day of year), we calculate the product of the respective value for each component. Hence, we obtain the security level over the period.
But the result is not continuous...

Calculating the metric

To construct a continuous function out of these measurements, we use splines to connect the points.
The security level of the IS is now defined as:

$$SL = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(x)dx$$

where $f(x)$ is the continuous function that we have created from the splines.

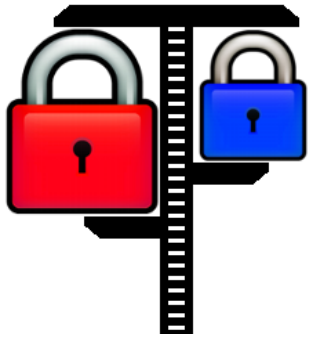
Advantages

The proposed metric provides:

- Unbiasedness.
- Change during time.
- Measure the security within any given time period.
- Allows a product to improve it's security status, old vulnerabilities affect less and less.

SecQua

SecQua



SecQua, is an open source project under GPL license, written in Python, which applies the aforementioned metric.
The source for the disclosed vulnerabilities is the National Vulnerability Database.
SecQua has a minimal GUI using wxWigdets and outputs graphs and LaTeXreports. The reports are for week days, days of month, months and days of years.

Examples

Configuration of IS_1

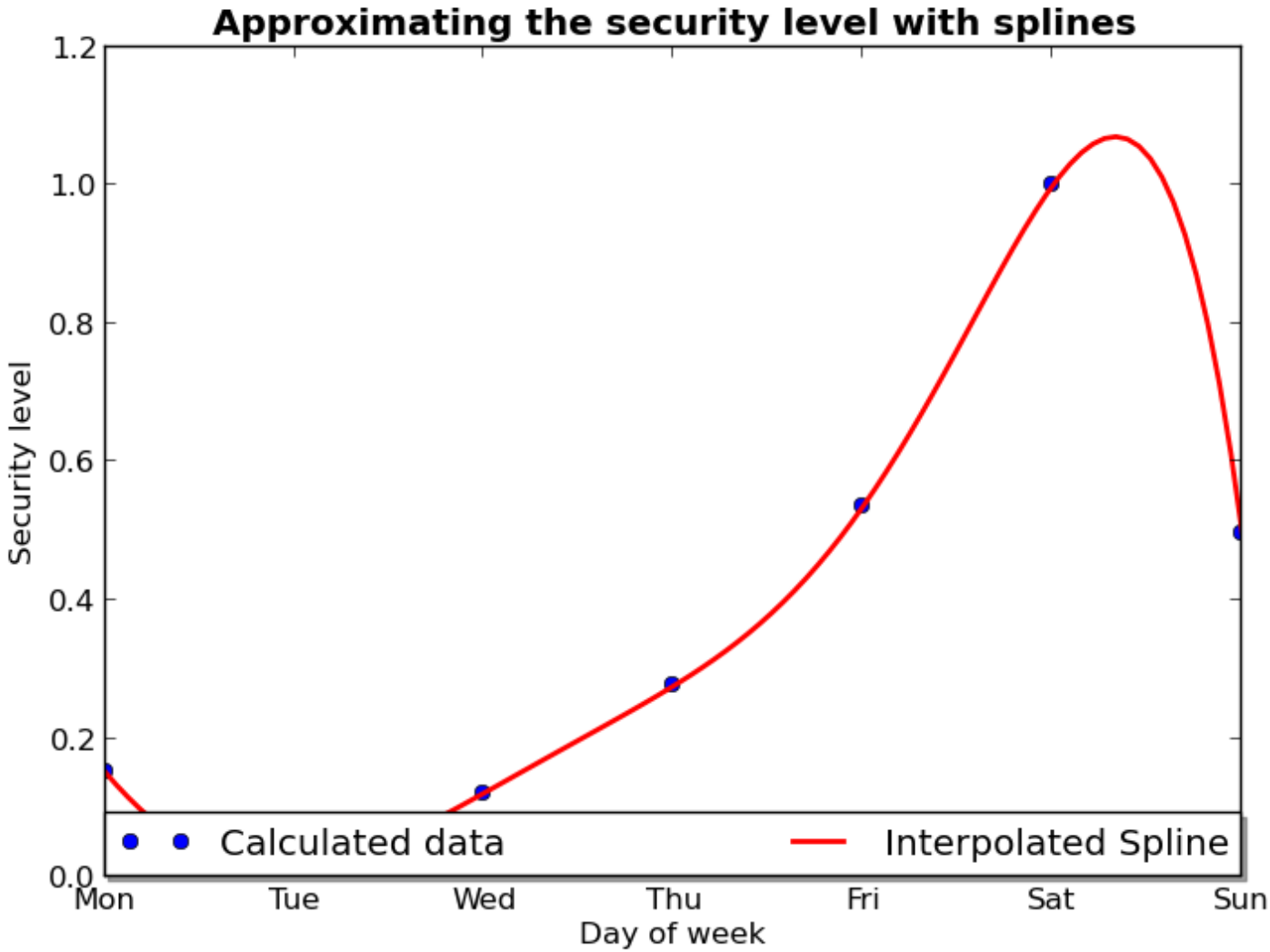
IS_1 consists of

- Windows server 2008
- IIS web server 7
- Oracle 11g

Measurements

	Windows Server 2008	IIS 7	Oracle 11g
c_i	6.57015322271	1.97284669437	0.430231551098
Mon	0.337232692575	0.44909640242	1.0
Tue	0.0604653464085	0.337373567122	0.543359728337
Wed	0.218679324266	0.5616937653	0.988751090289
Thu	0.358757002916	0.778569056429	0.989431590167
Fri	0.536303347655	1.0	1.0
Sat	1.0	1.0	1.0
Sun	0.495567781343	1.0	1.0

SecQua output for IS_1



Configuration of IS_2

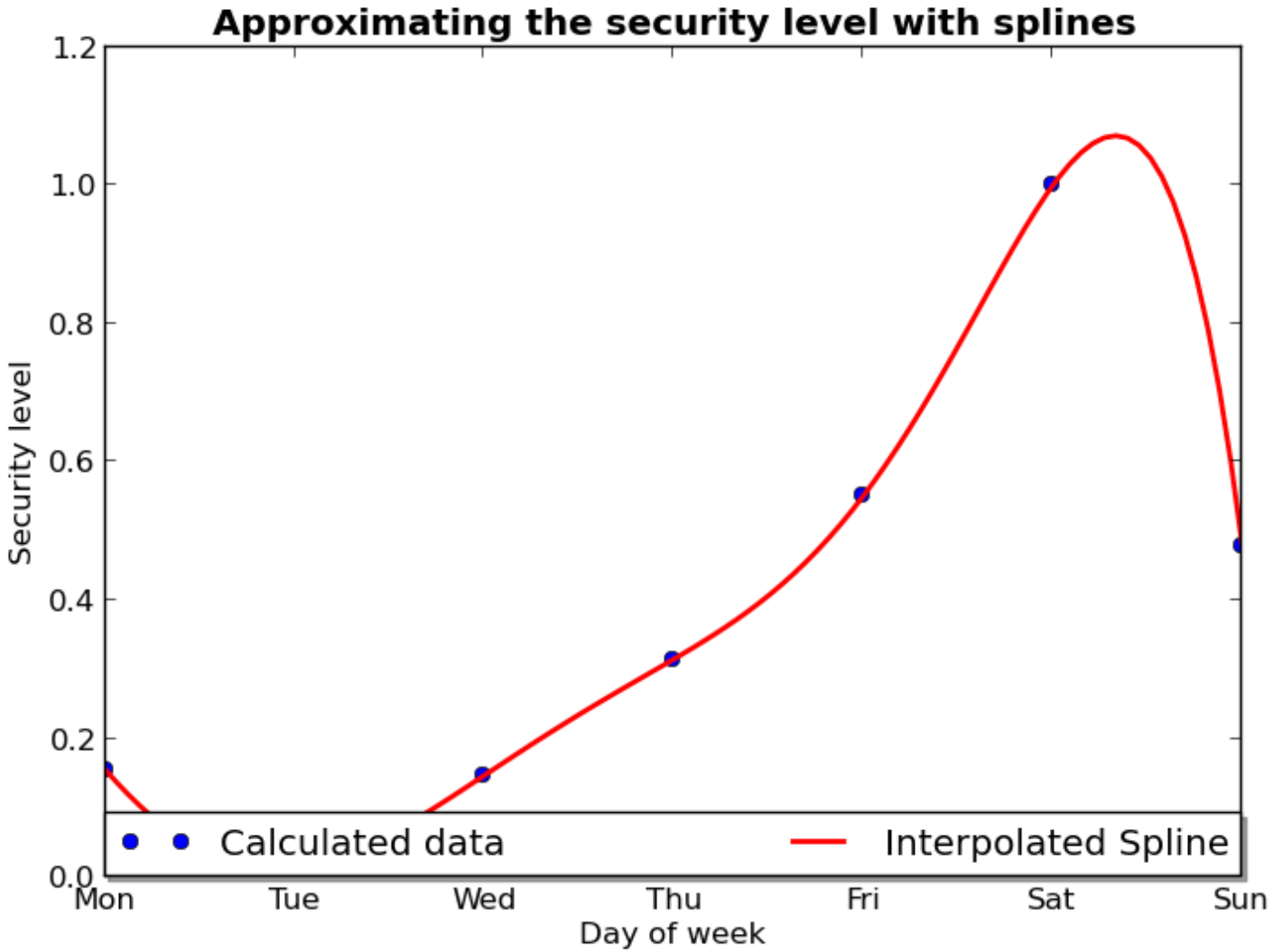
IS_2 consists of

- Windows server 2003
- IIS web server
- Oracle 11g

Measurements

	Windows Server 2003	**IIS 7**	**Oracle 11g**
c_i	5.68424479875	1.97284669437	0.430231551098
Mon	0.345482762215	0.44909640242	1.0
Tue	0.0731125858816	0.337373567122	0.543359728337
Wed	0.264228019695	0.5616937653	0.988751090289
Thu	0.40873166519	0.778569056429	0.989431590167
Fri	0.550884830157	1.0	1.0
Sat	1.0	1.0	1.0
Sun	0.478047360432	1.0	1.0

SecQua output for IS_2



Configuration of IS_3

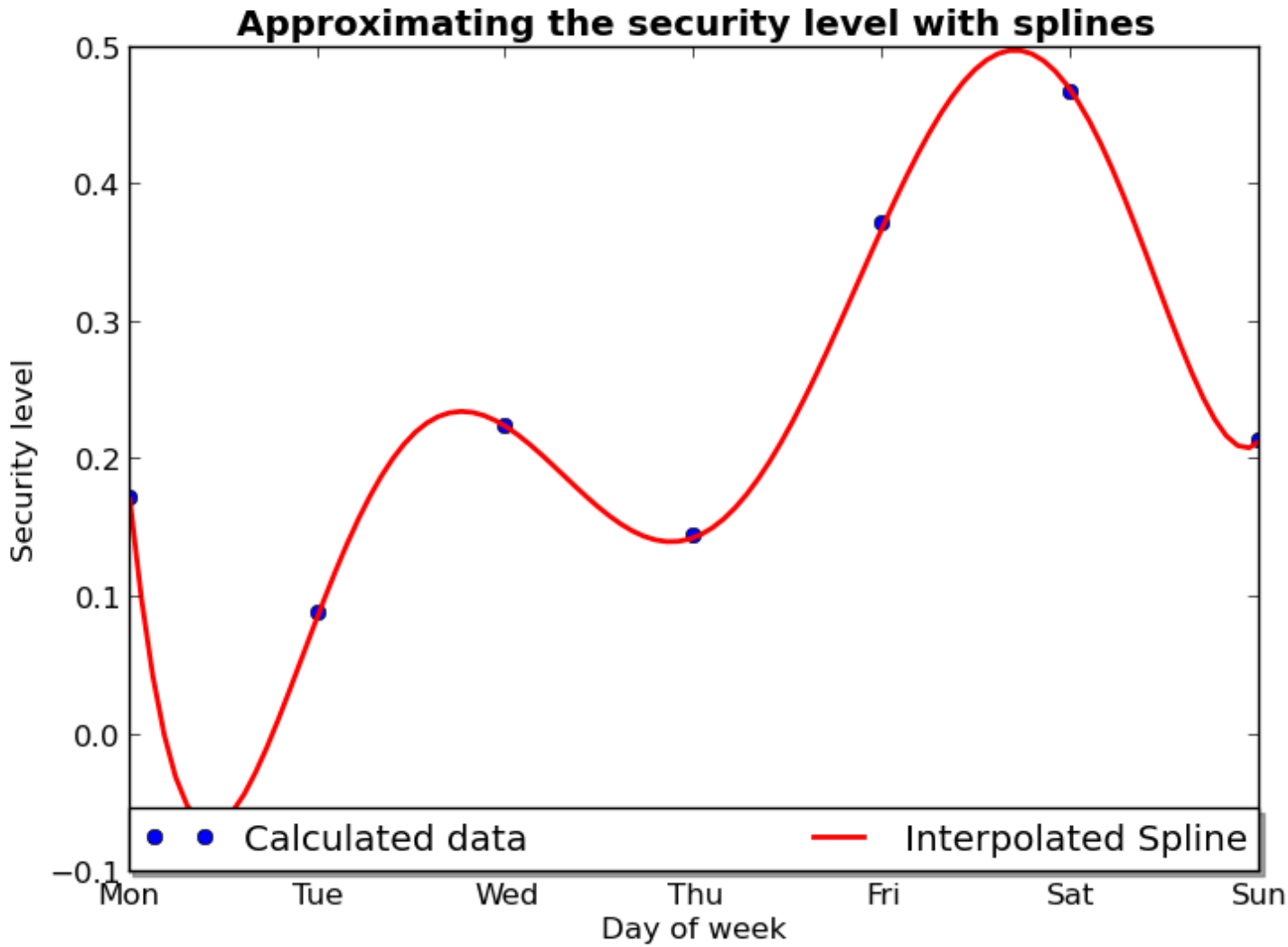
IS_3 consists of

- Gnu/Linux Kernel above 2.6.20
- Apache 2 web server
- Oracle 11g

Measurements

	Linux kernel $\geq$2.6.20	**Apache 2.2**	**Oracle 11g**
c_i	2.96581135411	3.20718763605	0.430231551098
Mon	0.415041555568	0.412682262653	1.0
Tue	0.386008233136	0.418199405458	0.543359728337
Wed	0.467996488081	0.483916624593	0.988751090289
Thu	0.439443830968	0.331002202049	0.989431590167
Fri	0.717896538379	0.516915726465	1.0
Sat	0.701411414652	0.666493844303	1.0
Sun	0.443239398249	0.482801821848	1.0

SecQua output for IS_3



Configuration of IS_4

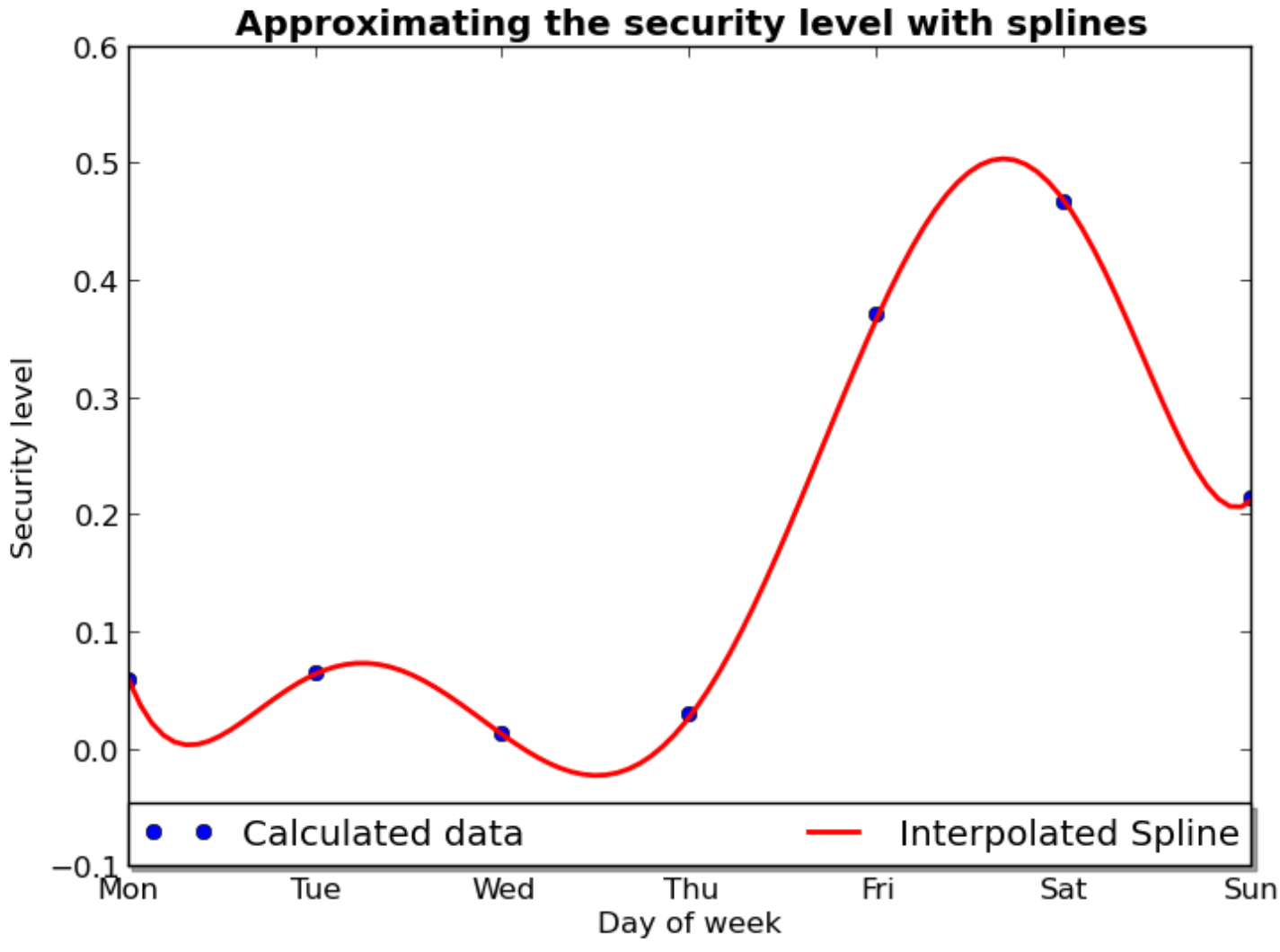
IS_4 consists of

- Gnu/Linux Kernel above 2.6.20
- Apache 2.2 web server
- MySQL 5.5

Measurements

	Linux kernel $\geq$2.6.20	**Apache 2.2**	**MySQL 5.5**
c_i	2.96581135411	3.20718763605	6.61564871894
Mon	0.415041555568	0.412682262653	0.34578441634
Tue	0.386008233136	0.418199405458	0.404384804723
Wed	0.467996488081	0.483916624593	0.0548968866281
Thu	0.439443830968	0.331002202049	0.205188632957
Fri	0.717896538379	0.516915726465	1.0
Sat	0.701411414652	0.666493844303	1.0
Sun	0.443239398249	0.482801821848	1.0

SecQua output for IS_4



Configuration of IS_5

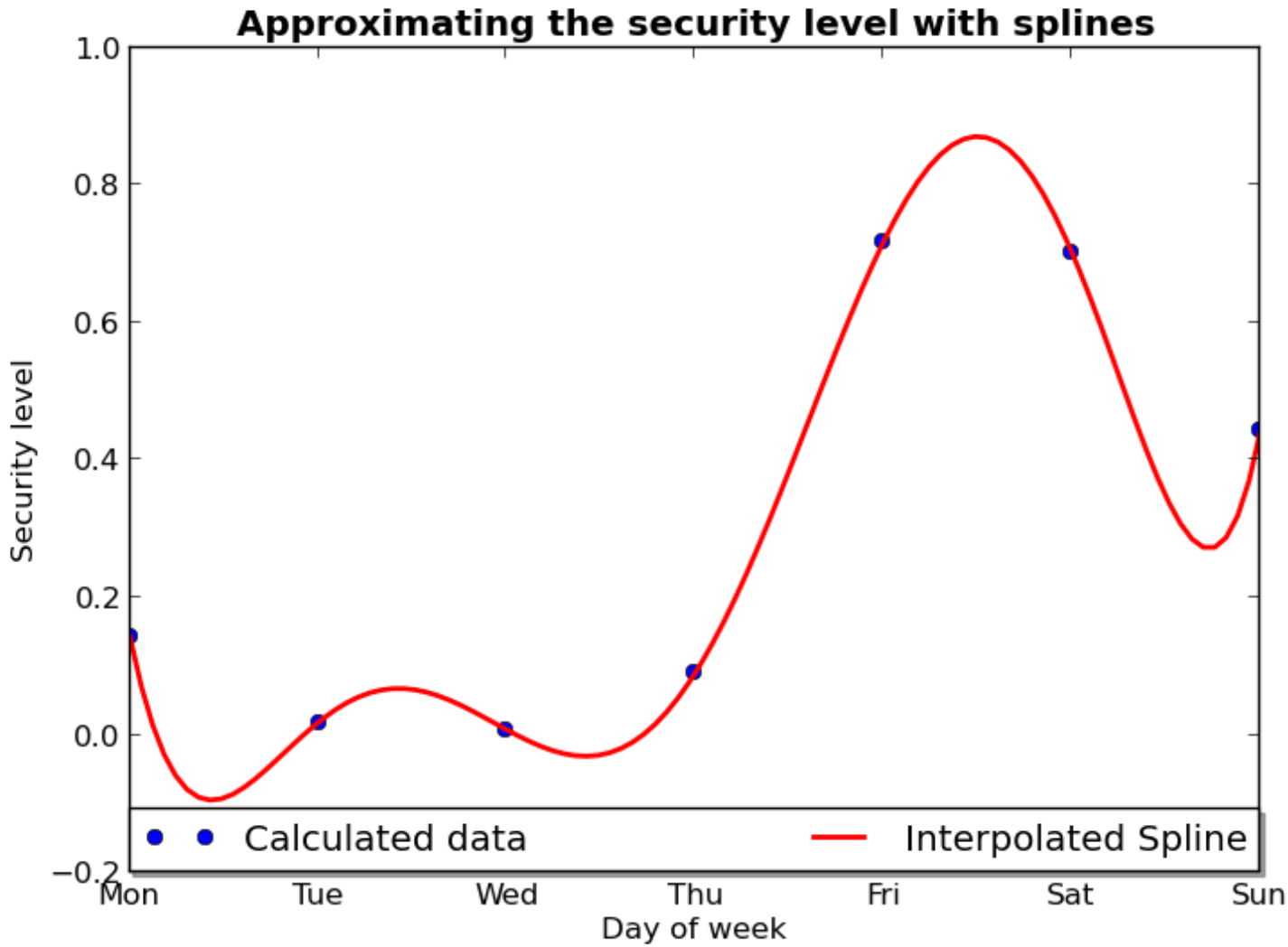
IS_5 consists of

- Gnu/Linux Kernel above 2.6.20
- nginx 10 web server
- MySQL 5.5

Measurements

	Linux kernel $\geq$2.6.20	**nginx 1**	**MySQL 5.5**
c_i	2.96581135411	3.21907538519	6.61564871894
Mon	0.415041555568	1.0	0.34578441634
Tue	0.386008233136	0.117441384104	0.404384804723
Wed	0.467996488081	0.290607319173	0.0548968866281
Thu	0.439443830968	1.0	0.205188632957
Fri	0.717896538379	1.0	1.0
Sat	0.701411414652	1.0	1.0
Sun	0.443239398249	1.0	1.0

SecQua output for IS_5



Configuration of IS_6

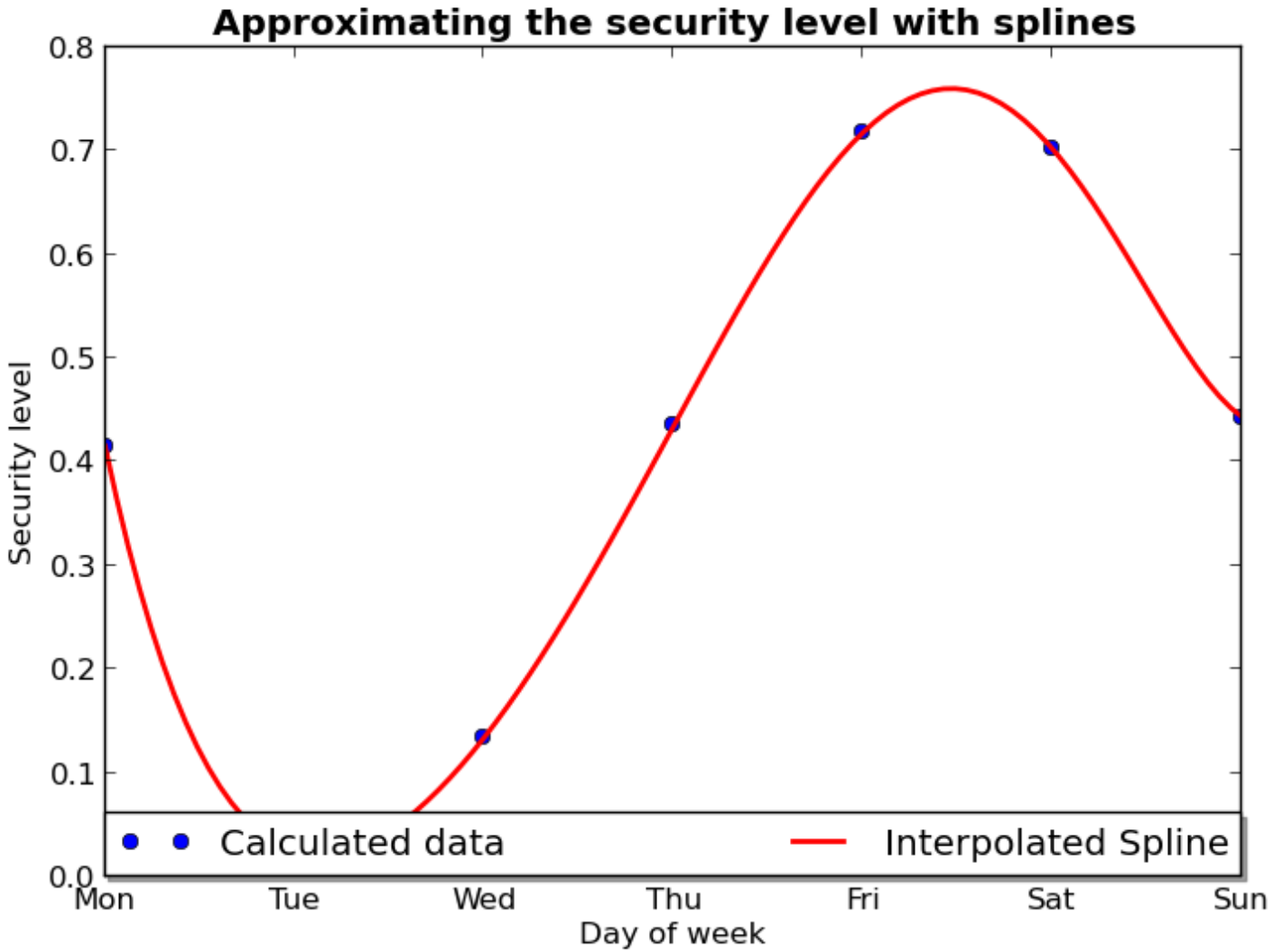
IS_5 consists of

- Gnu/Linux Kernel above 2.6.20
- nginx 10 web server
- Oracle 11g

Measurements

	Linux kernel $\geq$2.6.20	**nginx 1**	**Oracle 11g**
c_i	2.96581135411	3.21907538519	0.430231551098
Mon	0.415041555568	1.0	1.0
Tue	0.386008233136	0.117441384104	0.543359728337
Wed	0.467996488081	0.290607319173	0.988751090289
Thu	0.439443830968	1.0	0.989431590167
Fri	0.717896538379	1.0	1.0
Sat	0.701411414652	1.0	1.0
Sun	0.443239398249	1.0	1.0

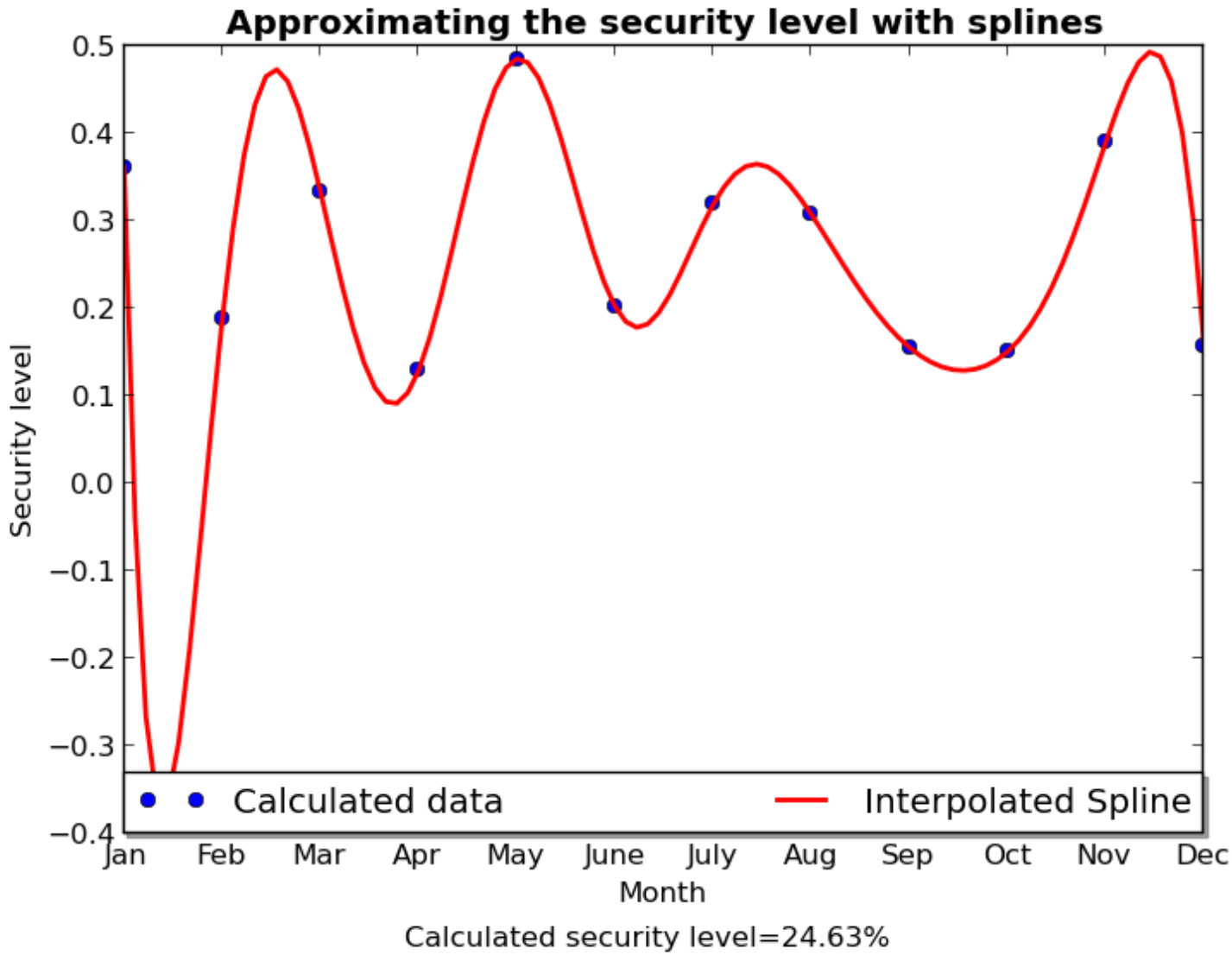
SecQua output for IS_6



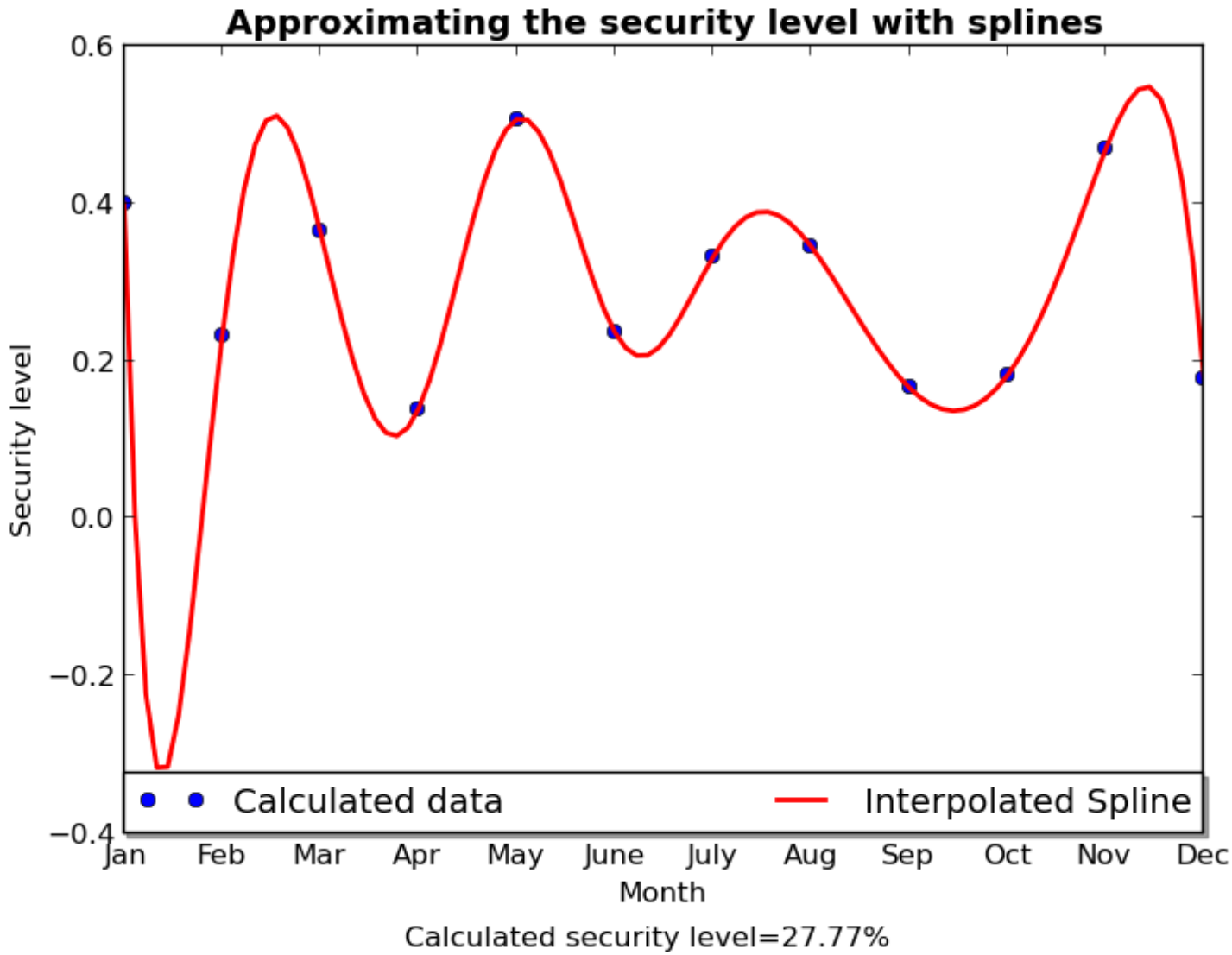
Calculated security level=39.74%

If we switch to view the results by month we have the following results.
Bear in mind that for proprietary software vulnerability disclosure is made at specific days a week most of the time.

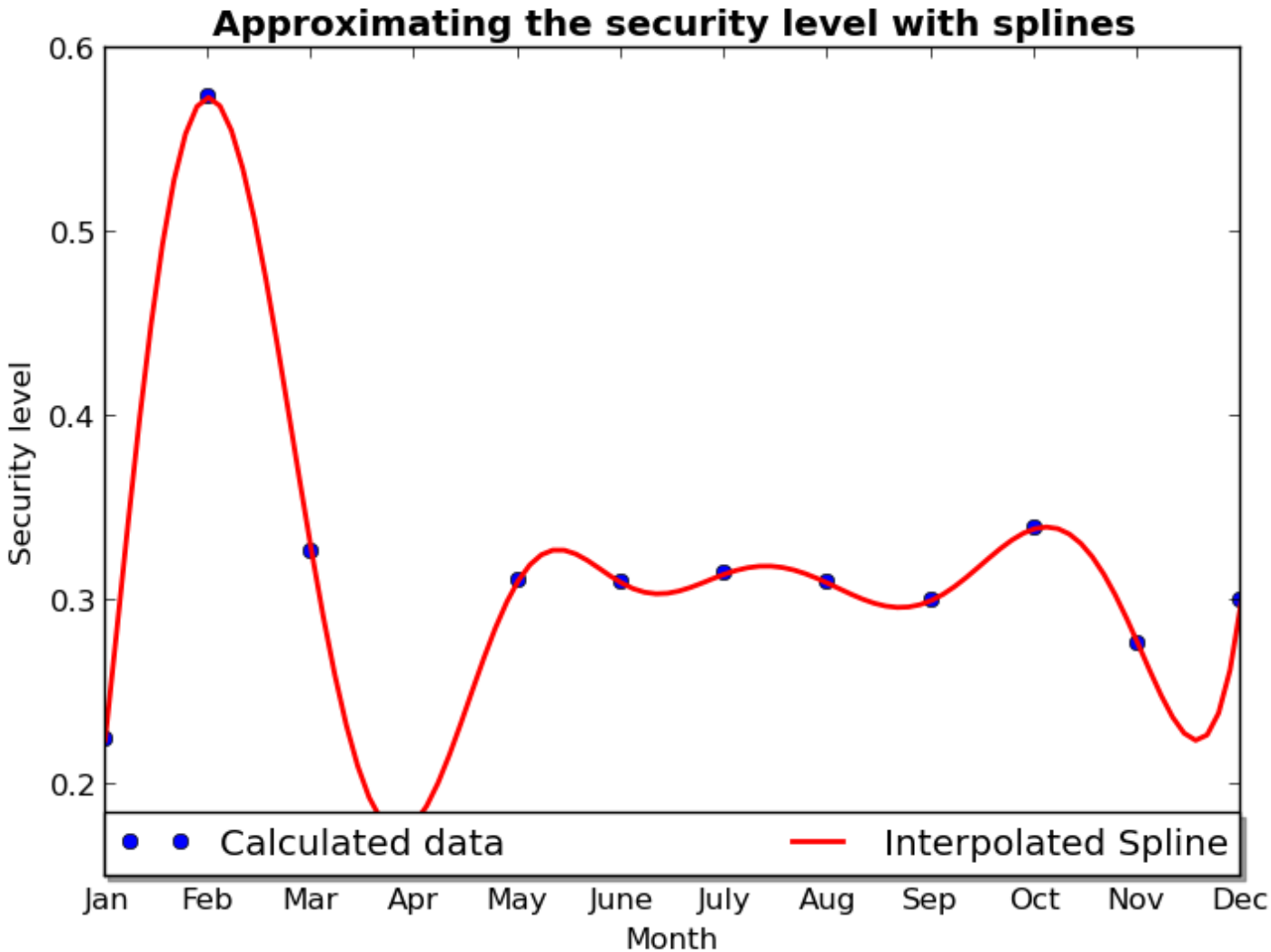
SecQua output for IS_1



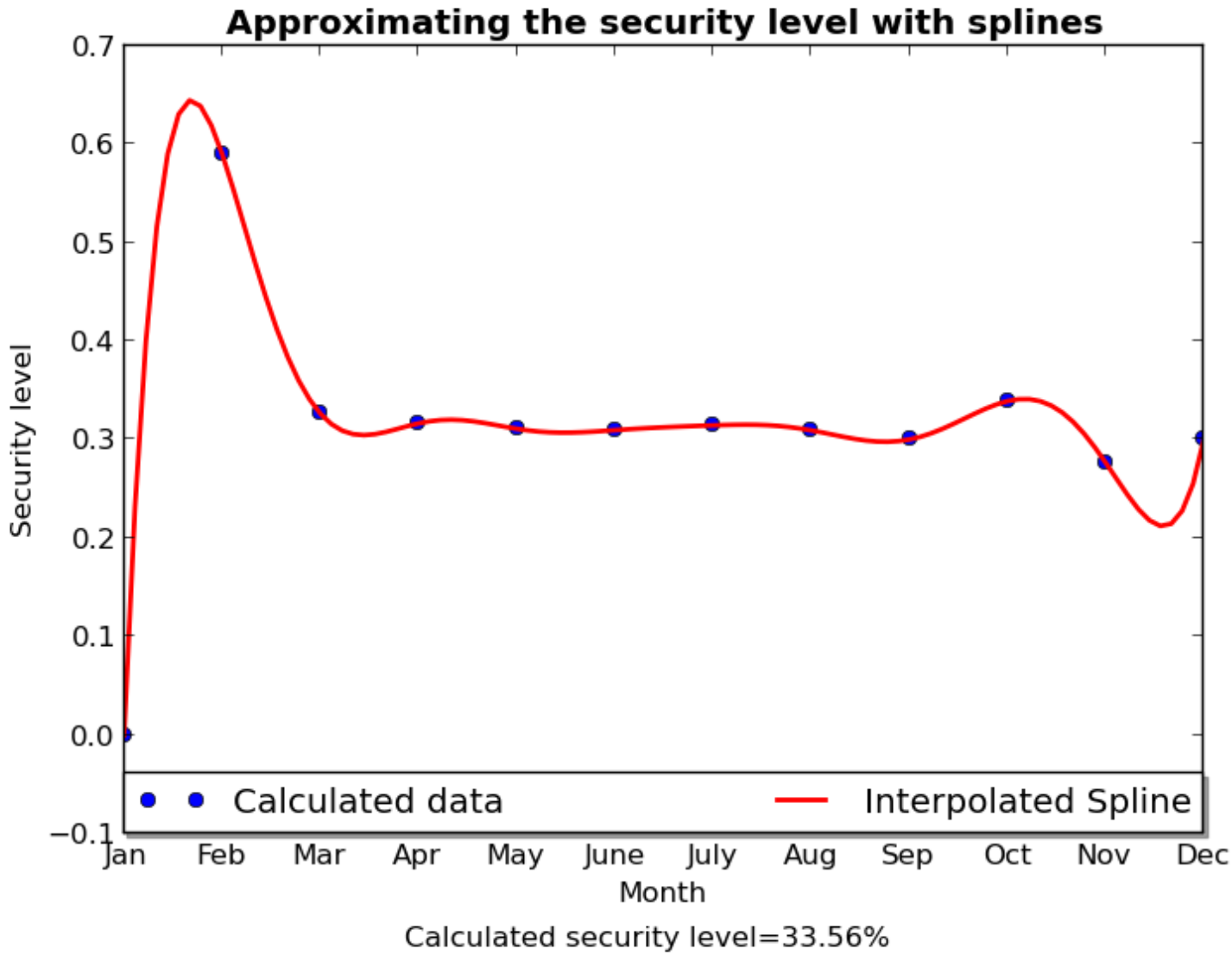
SecQua output for IS_2



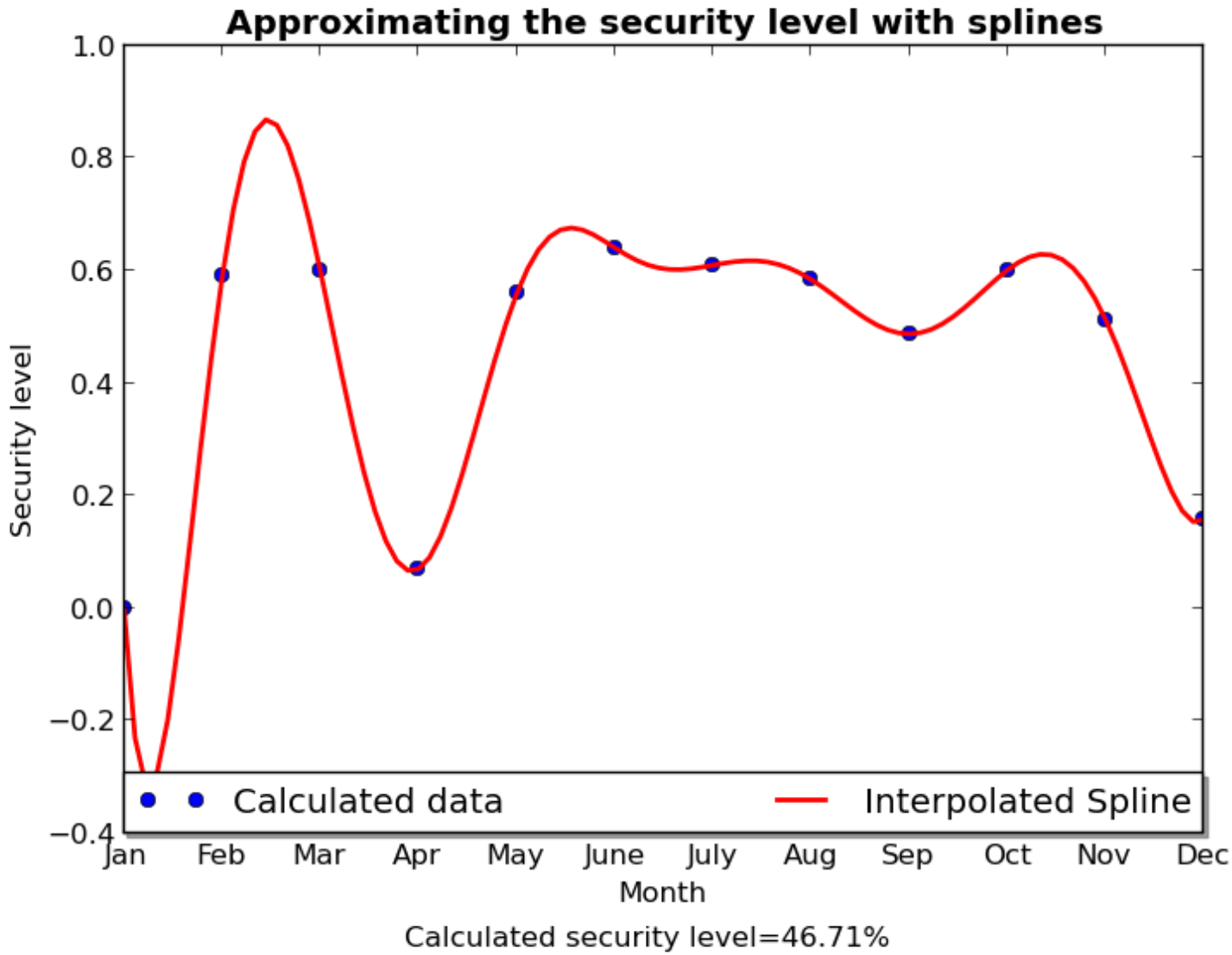
SecQua output for IS_3



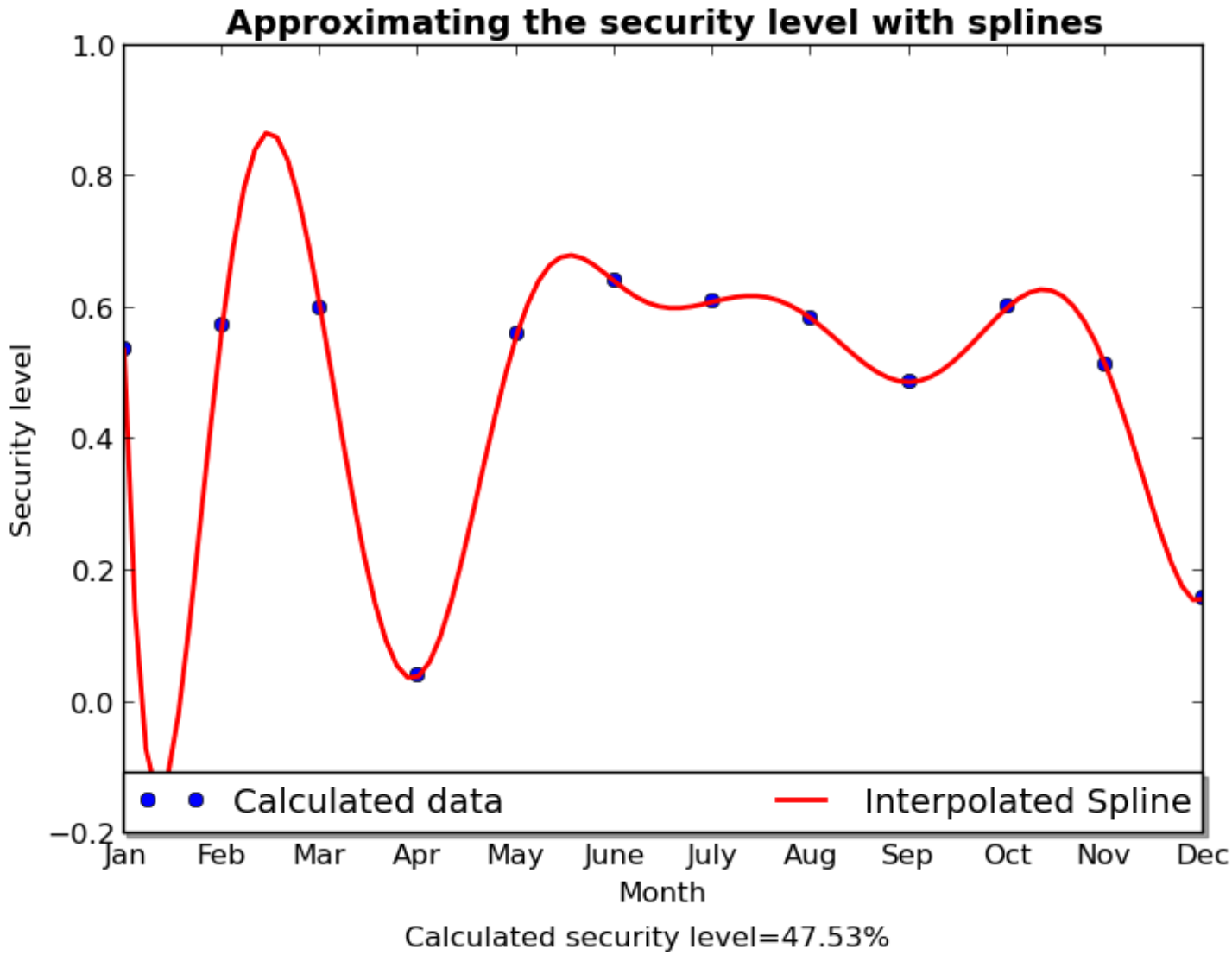
SecQua output for IS_4



SecQua output for IS_5



SecQua output for IS_6



What's next?

Extending SecQua

- Improve data entry... (use listctr, add components preview)
- Support for other vulnerability databases
  OSVDB for example, if it manages to stay open...
- Provision for patching date.
  Has not been used because it isn't stored in most vulnerabilities.
- Users/Admins/Managers feedback!
  Apply the metric, compare your results and get back to us.
- Ideas to improve!

Thank you

Thanks for attending!

Q&A time...

For more on SecQua please visit its page on SourceForge

`https://sourceforge.net/projects/secqua/`

or email:

`kpatsak@gmail.com`

References

D. Geer, K. Hoo, and A. Jaquith.
Information security: Why the future belongs to the quants.

IEEE Security and Privacy, July/August, 2003.

IATAC.
Measuring cyber security and information assurance, May 8, 2009.

C. Patsakis, G Chondrocoukis, D. Mermigas, and S. Pirounias.
The role of weighted entropy in security quantification.
In *International Conference On Information Security And Artificial Intelligence (ISAI 2010)*, 2010.

C. Patsakis, D. Mermigas, S. Pirounias, N. Alexandris, and E. Fountas.
Towards a formalistic measuring of security using stochastic calculus.

In *IEEE ICCSIT 2010, 2010 3rd IEEE International Conference on Computer Science and Information Technology.*, 2010.