Security Principles for Public-Resource Modeling Research

David Stainforth[1], Andrew Martin[2], Andrew Simpson[2], Carl Christensen[2],
Jamie Kettleborough[3], Tolu Aina[1], and Myles Allen[1]
[1]*Atmospheric Physics, Oxford University (UK),* [2]*Oxford University Computing Laboratory,*
[3]*Rutherford Appleton Laboratory, Didcot, Oxfordshire, UK*
d.stainforth1@physics.ox.ac.uk, [Andrew.Martin,Andrew.Simpson,Carl.Christensen]@comlab.ox.ac.uk
J.A.Kettleborough@rl.ac.uk, tolu@atm.ox.ac.uk, m.allen1@physics.ox.ac.uk

Abstract

Large-scale distributed computing projects have many security concerns due to their public and often "open" nature. Climate*prediction*.net *(CPDN) is taking the concept of public-resource, high-throughput Grid computing a stage further, by using it for a major piece of modeling research. The aim is to harness the spare CPU cycles of potentially millions of individual users' PCs to run a massive ensemble of climate simulations. In doing so it has been faced with a range of security and integrity issues beyond those encountered by previous projects but likely also to confront similar initiatives in the future. This paper introduces the project and its software architecture, and outlines a threat model for such situations, including threats to participants, threats to the experiment and threats to other stakeholders. It goes on to discuss how these threats have been addressed, with the procedures presented expected to form a valuable foundation for an increasing number of similar projects.*

1. Introduction

Climate*prediction*.net (CPDN) is taking the concept of public-resource, high-throughput Grid computing (already demonstrated by projects such as SETI@home) a stage further, by using it for a major piece of modeling research. The overall objectives of the climate*prediction*.net project ([1] – [4]) are as follows: to harness the power of idle home and business PCs to provide the first fully probabilistic 50-year forecast of human induced climate change, to enhance public understanding of climate modeling and the nature of uncertainty in climate prediction, and to demonstrate the potential of public-resource distributed computing for Monte Carlo ensemble simulations of chaotic systems.

The quantification of uncertainty in climate predictions requires of order one to two million integrations of a complex climate model [5]. This is beyond the scope of conventional supercomputing facilities but could be achieved using the Grid concept of "high-throughput computing" [6] which has been demonstrated by a number of "public-resource" projects such as SETI@Home ([7], [8]). The principle is to utilize idle processing capacity on PCs volunteered by businesses and the general public.

This project differs from previous projects in a number of ways including: the volume and management of the distributed data generated, the scale of the data collection problem, the complexity, granularity and duration of the computational task, and the necessity to maintain long-term participant interest. These distinctive aspects create a number of security challenges and raise a variety of issues regarding experimental and data integrity, above and beyond the experience of other Internet-based distributed computing projects.

This paper presents a threat model for the project, along with the security features developed to cope with such threats. It is anticipated that the results presented will serve as an initial attempt at a generic template for managing security in future public-resource high-throughput Grid applications.

The paper first introduces the climate*prediction*.net project and its software architecture in Sections 2 and 3. In Section 4 the security and experimental integrity features are described, focusing first on threats to stakeholders then on threats to participants and finally threats to the experiment itself. Finally, in Section 5, the contribution of the paper is reviewed.

2. Motivation for climate*prediction*.net

Recent studies have shown that *atmosphere-ocean general circulation models* (AOGCMs) are capable of simulating some large-scale features of present-day climate and recent climate change with remarkable accuracy (e.g. [9]). These models contain, however, large numbers of adjustable parameters, many of whose values are poorly constrained by the available data on the processes they are supposed to represent and which are known, individually, to have a significant impact on simulated climate. The practical question is therefore: to what extent might a different choice of parameters or parameterizations result in an equally realistic simulation of 20th century climate yet a different forecast for 21st century climate change? (See [5].) This is acknowledged by the Inter-Governmental Panel on Climate Change (IPCC) as one of the key outstanding priorities for climate research [10].

Currently available supercomputing resources would be completely inadequate for this task ([3], [4]). However, the continuing increase in computing capacity of the average PC has meant that AOGCMs, which until recently could only be sensibly run on supercomputers, can now be run on commonly available desktop machines. This opens up the possibility of carrying out the above research by using high-throughput public-resource computing, which has been demonstrated by a number of projects such as SETI@home ([7], [8]), FightAIDS@home [12] and Compute Against Cancer [13]. The most successful have been those that stimulate the public's imagination and interest, and in such situations tremendous computing capacity has been accessed at relatively low cost.

3. Project Challenges and Architecture

A suitable architecture for climate*prediction*.net requires a design that builds on the experience of related projects while addressing the challenges peculiar to this project. Below is a summary of related work, the system architecture and the issues that have influenced its design; concentrating on the aspects that have relevance for security features. A more complete description is given in [4].

3.1. Related Work

As mentioned above, the fundamental nature of climate*prediction*.net is similar to several previous projects, but it differs in the way in which resources are utilised. On CPDN, each simulation takes one to three months of elapsed time to run, therefore long-term commitment on the participants' behalf is essential. Furthermore, large volumes of data will be produced, perhaps up to one gigabyte on each machine, giving petabytes in total. Although all of this data is potentially useful, it cannot affordably be gathered in one place with present technology and costs.

The problems are therefore similar to other Grid and DataGrid projects, such as those related to the Large Hadron Collider (LHC) ([17]), but in contrast, here the data is produced in a massively distributed fashion and needs to be gathered together into a server based, distributed archive for analysis. Bandwidth issues for home PCs mean that most of it will nevertheless remain on the participants' machines with only the critical data being collected centrally. Consequently efficient post-processing of model-produced data is essential, with as much analysis as possible being distributed, to minimise the volume of data collected. In the longer term, dynamic updating of the post-processing software will be a critical aspect of experimental analysis for projects of this nature.

3.2. The Challenges

There are a number of factors that create particular problems when addressing security issues for any public-resource modeling project of this nature. Foremost is the requirement that the software be suitable for use on machines with infrequent, low speed, modem-based Internet connections. This comes about from the need to be inclusive in the participation base in order to raise awareness and stimulate more informed public debate on the associated scientific issues. In addition, there is the problem of implementing a complex model in a simply installable package on a popular operating system such as Windows. In particular, unlike previous public-resource projects, the UM comes with a series of input files and produces a large number of output, diagnostic files. The architecture must cope with the possibility that it may be desirable to replace any of the input files, and possibly the model executable itself, during the registration process in which the participant is allocated a particular simulation within the experimental design. The resulting volume of data is large so it is necessary to have a design that allows for an expandable number of data "upload" servers on which the collected data can be held.

The integrity of this data, and issues of cataloguing, metadata, provenance data, analysis and visualization of this distributed archive, pose additional problems; some of them in common with other DataGrid projects ([17], [18]). Many of the tools necessary for this will

be required as much for participants as for researchers, because the public – particularly schools and colleges – will want to undertake their own "mini-research" projects. Indeed this is a major aim of the project, as well as being necessary to maintain commitment and interest. Consequently it is also necessary to provide visualization tools with which they can keep track of their own simulation; something that raises additional software challenges.

4. Security

The security issues fall into three groups as shown in Table 1. Threats to the participants and the experiment are, of course, closely related since the success of the experiment depends very largely on the goodwill of the participants. If the client software were to lead to damage to participants' machines, or if that were widely reported as a possibility, many participants might withdraw.

4.1. Threats to Stakeholders

The model software is a valuable piece of proprietary code donated by the UK Meteorological Office, representing many years of research and development effort. It is used not only for climate research but also for commercial activities, including weather forecasting. It therefore has substantial intrinsic value and consequently its owners require that it remain closed source and that participants agree to a licence restricting its use to that arranged by climate*prediction*.net. In practice, setting up the model for other uses is a highly specialized task involving a complex user interface and a wide range of additional scripts, even when the source code is available. But this is still a threat that should not be taken lightly. Furthermore, it may be more critical in other projects with simpler but just as valuable code.

4.2. Threats to Participants

The chief threat to participants is that they will unwittingly become a host to some "malware." The principal means of guarding against this is by digitally signing the distributed package using a commercial certificate, identifiable and accepted by the most commonly used browsers. Of course for this to be effective the private key must be secure and the code verifiable, so the software is compiled (and the self-extracting installation package constructed and signed) on a stand-alone machine used only for this purpose. Strict access control procedures are employed to ensure the integrity of this machine.

When the package is downloaded from the Web, a browser with appropriate security settings checks the certificate and indicates that the package has been signed by climate*prediction*.net. The participant can therefore be reassured that the software he/she is installing has not been tampered with. For those who do not have their browser security settings set sufficiently high to check for digital certificates, or who may receive the package from other sources (e.g. CD-ROMs or from friends or co-workers), two further levels of security are also critical. The certificate can be validated "post-download" (or on the CD-ROM, email attachment, etc) via Windows security features to check the digital certificate. Also, the MD5 checksum of the valid CPDN package is published on the website. Instructions are given to users on the website on how to check their CPDN installation package.

An important design principle is that the client always initiates communications, thus avoiding the risk that the package could facilitate unauthorised machine access. There are no server processes running on TCP ports. The client basically acts as a "web browser" and performs an HTTP POST (over port 80 to an upload server, or port 443 (HTTPS/SSL) to the central server). Furthermore, the distributed package is designed so that installation makes minimal changes to the PC set-up and can be uninstalled easily and completely; important factors for some potential users.

A further area of security for participants is in respect of personal data. It is desirable for the project to collect limited information on participants and their machines in order to allocate simulations optimally according to machine capabilities and bandwidth. This personal data is also used to provide personalization on their web-based user page (results, leaderboard, etc). Whilst not particularly sensitive, this data falls within the definition of "Personal Data", so that European privacy law [19] requires adequate organisational and technical means to be employed in maintaining its accuracy and privacy.

Therefore, climate*prediction*.net employs secure sockets (HTTPS/SSL) to protect personal data in transit, using a commercially-signed server certificate to authenticate the registration server. Of course, security of the registration database itself is even more critical. This is held on a firewalled subnet and standard database and server security measures will be employed, such as limited and logged login access, restricted client connections via ipchains, iptables etc.

4.3. Threats to the Experiment

Perhaps the most difficult threat to counter is that to experimental integrity. This can be separated into three areas as shown in Table 1.

4.3.1. Prevention of Tampering. Significant threats could arise from participants tampering with the run they have been assigned. For instance they could attempt to alter the parameters of their run, return data generated by a different run, modify the model, falsify final data files, or impersonate the client to overwhelm the servers.

The risk of false data being returned accidentally can be minimised and the efforts of those who might try to return false data intentionally can be reduced, by appropriate design measures. However, two factors mitigate against reliably preventing the malicious return of any false results: participant involvement is voluntary and communication is only required during registration and data upload (see Sections 3.2 and 4). Consequently, there is no means of preventing the code being reverse engineered and replaced by an alternative version that produces false data but either appears authentic to the servers or is only used between server communications.

In practice, however, producing plausible but false or biased results would be extremely difficult and the experience of SETI@Home has been that reverse engineering can be easily discouraged because it is mostly done to help rather than hinder the project. Nevertheless, it is important not just that the results are correct but that they are demonstrably correct and not influenced by partisan organisations. The maintenance of overall experimental integrity is therefore ensured by three mechanisms:

1) Initial Condition Ensembles: The experimental design [3] requires that all simulations be repeated several times using different initial model conditions (see Figure 4, ref [3]). This is known as an initial condition (IC) ensemble and is used to increase the signal-to-noise ratio of the results, given the chaotic nature of the system. The members of each IC ensemble can be compared and used to identify suspicious results. Most modified runs are likely to be obvious on superficial automated inspection but some may appear to be plausible but questionable. These uncertain runs can simply be repeated identically.

2) Simulation repetition: A certain percentage (order 10%) of simulations are simply being repeated identically. This will provide an indication of the scale of any tampering problem. However, because the simulations take so long it is not sensible to repeat every simulation identically several times, as is done in other projects. This would be a substantial waste of computing resources.

3) Tampering prevention: As described above, it appears to be impossible to implement a fail-safe mechanism to prevent tampering. On the other hand, if tampering is trivially easy many participants may change their set-up just to get more interesting and extreme results, perhaps to visualize and share with friends. This ignores the relevance of the wider experiment. For instance, increasing the levels of critical humidity for cloud formation could be easily achieved by changing a clearly labelled parameter in a FORTRAN namelist.

The client has been set up so that the initial package includes a list of all files along with their checksums. The model has been adapted so that before any file is opened, the checksum is calculated and compared with the value stored in the list. After any file is closed the checksum is recalculated and if it has changed the new value is stored for future use. New diagnostic files are added to the list as they are created. A count is kept of the number of failed checksum comparisons, as an indication of possibly suspicious data. The checksums of diagnostic files no longer accessed by the model are stored for use as verification that no changes are made before they are used in post-processing and data upload.

Initially an MD5 hashing algorithm was proposed for the checksum. However, although this is cryptographically secure it is also widely available so it would be relatively easy for any reasonably computer literate individual to change a file and put the relevant MD5 checksum into the file list, thus defeating the security measure. It has therefore been decided to use a simpler but proprietary checksum algorithm. Although it will be possible to deduce this algorithm by reverse engineering the code, this is non-trivial for the average PC owner. Indeed it may be beneficial to publish the algorithm, thus removing any kudos associated with "cracking the code". In any case, executables to calculate the checksum will not be as widely available as those for MD5. The hope is therefore that this mechanism will prevent casual tampering while not launching an unwinnable war with those who like to break systems.

4.3.2. Data corruption in transit. At the end of each simulation a subset of the diagnostic files produced is returned to one of the upload servers. But how does the upload server know whether the files being offered come from a participating client? This problem is dealt with by the client contacting the central server and indicating that its simulation has finished, and then providing a list of files and related checksums which it

has ready for return. The central server then checks the details of the client's globally unique identifier (GUID) and the associated simulation. If they match, the list of files is digitally signed by the central server and sent back to the client along with a URL for a suitable upload server. The client then contacts its assigned upload server and provides the digitally signed file list and the files themselves. The digital signature is checked by the upload server to verify that the files come from a client that has been verified by the central server. The checksums can be used to demonstrate the integrity of the files upon transfer from the client to the server.

4.3.3. Server and Database Corruption. The servers are potentially subject to all of the common attacks: attempts to change Web pages, denial of service, theft of personal data, corruption of stored data files, and so on. Of course the data generated and the experimental database of participants and diagnostic file metadata, has great intrinsic value. They therefore need to be protected against not just malicious attack but also accidental loss, corruption or deletion. As indicated in Section 4.2, standard database and server security measures have been implemented and the database has been located on a firewalled subnet.

Access to the complete range of diagnostic files will be made available to researchers using the Earth System Grid II software [18] and other proprietary software developed by the project (in design). It is anticipated that the results of interest to most participants can be made available securely using standard Web techniques involving HTML and CGI scripting. The question of optimal caching of the most frequently accessed results, both for researchers and the wider public, is an issue currently being researched.

5. Conclusions

This paper represents a summary of the crucial aspects of security planning and development for climate*prediction*.net. One of its aims is to reassure participants, and the scientific community, that we have explored the issue from many perspectives, and produced a reliable, secure system. Of course, it could be argued that this is a risky strategy as the paper could be seen as a manual on how to break the project. However, we feel that if public-resource computing is to continue to capture the public's imagination, issues of security need discussion and review by a wide group of experts. They will certainly receive it in the press and on the Web, so academic input can only be beneficial.

The core of our integrity measures is to discourage casual tampering whilst avoiding an unwinnable "arms race" against determined opponents. We hope that there will be too little challenge in breaking the project's software for it to be interesting to try. The experience of other projects seems to support this *laissez-faire* approach. The social value of such projects limits the number of adversaries interested in breaking them; however there are always those that would do it because it is "fun" or because they mistakenly think that there are political motives behind the project, with which they disagree, etc.

Nevertheless, if the results are to be trustworthy we must not only put in place measures of suitable strength but also collect data on the efficacy of those procedures and demonstrate overall experimental integrity. We believe that the measures outlined above will enable us to meet these goals and will provide a structure for security planning in future public-resource high-throughput modeling applications. As always, we invite you to visit climateprediction.net and run the client for yourself!

6. Acknowledgements

The authors wish to thank David Anderson for his insights from the experiences of the SETI@Home and BOINC teams. Also, thanks to the UK Met Office for their support of the project; as well as DTI and NERC who are major sources of funding of the CPDN project.

7. References

[1] M. R. Allen. Do-it-yourself climate prediction, *Nature*, 401:627, 1999.

[2] J. A. Hansen, M. R. Allen, D. A. Stainforth, A. Heaps, & P. Stott. Casino-21: Climate Simulation of the 21st Century, *World Resource Review 13*, 2001.

[3] D. A. Stainforth, J. Kettleborough, et al. Distributed Computing for Public Interest Climate Modeling Research, *Computing in Science and Engineering 4*(3): 82-89, 2002.

[4] D. A. Stainforth, J. Kettleborough, et al Climate*prediction*.net: Design Principles, *14th IASTED International Conference on Parallel and Distributed Computing Systems*, 2002.

[5] M. R. Allen and D. A. Stainforth (2002). "Towards Objective Probabilistic Climate Forecasting", Nature, 419:228

[6] I. Foster and C. Kesselman, (eds). *The Grid: Blueprint for a new computing infrastructure*, Chapter 2 (Morgan Kaufman, 1999).

[7] W.T. Sullivan III, D. Wertheimer, S. Bowyer, J. Cobb, D. Gedeye & D. Anderson. New major SETI project. *Proc. of the 5th Intl. Conference on Bioastronomy*.

[8] E. Korpela, D. Werthimer, D. Anderson, et al. SETI@home – Massively distributed computing, *Computing in Science and Engineering*, 3 (1): 78-83 Jan-Feb 2001.

[9] P. Stott, S. Tett, G. Jones, et al. External Control of Twentieth Century Temperature Variations by Natural and Anthropogenic Forcings, *Science 290*, 2000, 2133-2137.

[10] Houghton, J.T. et al. *Climate Change 2001, The Scientific Basis*, Cambridge Univ Press, Cambridge 2001.

[11] K.D. Williams, C.A. Senior & J.F.B. Mitchell, Transient climate change in the Hadley Centre models: The role of physical processes. *Journal of Climate, 14* (12): 2659-2674 2001.

[12] URL: http://www.fightaidsathome.org/

[13] L. Hand, Computing in Cancer Research, *The Scientist* 15(1), 2001.

[14] C. Gordon, C. Cooper, C.A. Senior et al, The Simulation of SST, Sea Ice Extents and Ocean Heat Transports. *Climate Dynamics 16*, 147-168, 2000.

[15] P. M. Cox, R. A. Betts, C. D. Jones et al, Acceleration of Global Warming Due to Carbon-Cycle Feedbacks in a Coupled Climate Model. *Nature*, 408:184, 2000.

[16] M. J. P. Cullen, The Unified Forecast/Climate Model, *Meteorological Magazine*, 122 (1449), 1993.

[17] URL:http://lhcgrid.web.cern.ch/lhcgrid/

[18] URL:http://www.earthsystemgrid.org/

[19] Data Protection Act, 1998. URL: http://www.legislation.hmso.gov.uk/acts/acts1998/19980029.htm

Threat:	**Form of threat:**	**Form of security**
To: Participants **By:** 3rd party via project servers and software.	Unexpected cost of participation – software having unwanted effects.	• Digitally signed software package. • Standard server security. • Info. on checking digital signatures. • Client initiated communications.
By: 3rd party via project servers	Release of personal data.	• Secure sockets. • Standard database and security measures. • Database on firewalled subnet.
To: The experiment. **By:** Participants	Tampering with simulation or data.	• Repeat simulations. • Initial condition ensembles. • Checksum monitoring of files.
By: 3rd party	Data corruption in transit.	• Secure sockets. • Digitally signed file list & checksums.
By: 3rd party	Server & Database corruption.	• Standard database and security measures. • Database on firewalled subnet.
To: Other stakeholders – e.g owner of the model. **By**: 3rd party.	Commercial use or proprietary code.	• Participant licence agreement. • Closed source model code.

Table 1: Threats and security measures.