Análise de vulnerabilidades e incidentes de segurança em grades de computação voluntária

Miriam von Zuben[1,2], **Marco Aurélio de Amaral Henriques**[2]

[1]Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
Núcleo de Informação e Coordenação do Ponto BR

[2]Departamento de Engenharia de Computação e Automação (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
UNICAMP

miriam@cert.br, marco@dca.fee.unicamp.br

Abstract. *The Volunteer Computing Grids, besides having many traditional security problems inherent to information systems, also have their own challenges due to questions like the anonimicity of the volunteers, falsification of results and credits and malicious projects. This article presents an analysis of the main security challenges faced by these systems, presents the attacks which they are subjected to and some measures to prevent them. A record of past incidents and the vulnerabilities found in the main projects are also commented.*

Resumo. *As Grades de Computação Voluntária, além de apresentar vários dos problemas tradicionais de segurança inerentes aos sistemas informação, possuem seus próprios desafios a serem enfrentados como anonimato dos voluntários, falsificação de resultados e de créditos e projetos mal-intencionados. Este artigo apresenta uma análise dos principais desafios de segurança enfrentados por estes sistemas, os ataques a que eles estão sujeitos e algumas contra-medidas para evitá-los. É apresentado também um histórico dos incidentes ocorridos e das vulnerabilidades descobertas nos principais sistemas.*

1. Introdução

A redução dos preços dos computadores e a facilidade de conexão advinda da Internet causou um grande aumento da popularidade dos computadores pessoais. Atualmente a Internet é utilizada por cerca de 24% da população mundial e estima-se que entre os anos de 2000 e 2008 sua taxa de uso tenha crescido cerca de 342% [IWS 2009] e que haja aproximadamente 570 milhões de computadores conectados a Internet [ISC 2008]. No Brasil a tendência de crescimento nas taxas de acesso também se mantém em alta. Entre os anos de 2007 e 2008 a taxa de domicílios com acesso a Internet passou de 17% para 20%, e o acesso por meio de banda-larga passou de 50% para 58% [NIC.br 2008].

O poder computacional dos computadores também sofreu um grande avanço nos últimos anos. Em 1993 o poder de processamento do computador mais rápido era de 59.7 GFlops, em 2003 este valor já se aproximava de 100 TFlops e em 2008 já ultrapassava a barreira do petaflop [Top500 2008]. Devido a este avanço usuários domésticos podem

adquirir hoje, a baixo custo, computadores com taxas de processamento antes disponíveis apenas em supercomputadores. Para alguns usuários toda esta capacidade de processamento é essencial; porém a grande maioria dos computadores pessoais possui seu poder de processamento desperdiçado, sendo utilizado na maior parte do tempo para a execução de tarefas rotineiras como edição de textos, visualização de páginas Web e programas de troca instantânea de mensagens, que não requerem grande capacidade de processamento.

Enquanto de um lado há uma grande quantidade de poder computacional sendo desperdiçado, por outro lado há uma grande carência de poder computacional para a resolução de problemas como simulações de fenômenos naturais, previsão numérica de tempo, modelagem de exploração de recursos minerais e de física nuclear. Para resolver estes problemas muitos pesquisadores investem uma parte significativa de seus recursos de pesquisa na compra ou aluguel de dispendiosos sistemas de computação paralela.

Os avanços tecnológicos aliados as necessidades de processamento e a subutilização dos computadores culminaram com o surgimento dos sistemas de computação em grade (*grid*), cuja finalidade é agrupar computadores distribuídos como um único e grande computador virtual e ser utilizado na execução de aplicações paralelas e distribuídas que necessitem alto poder de processamento e armazenamento. Uma categoria destes sistemas são os sistemas de computação voluntária que correspondem a um tipo de computação distribuída que utiliza os recursos doados por voluntários que, deliberadamente, cedem os ciclos ociosos de seus computadores para contribuir para projetos humanitários, científicos, matemáticos e de grande apelo emocional.

Os sistemas de grade trouxeram grandes desafios relacionados a segurança, escalabilidade, interoperabilidade, tolerância a falhas e volatilidade da rede. Os sistemas de computação voluntária, além de herdarem estes desafios, possuem suas próprias questões como anonimato dos participantes, proteção aos dados dos sistemas, dos projetos e dos computadores voluntários, necessidade de não interferir no funcionamento normal dos computadores voluntários e implementação de mecanismos para controle de créditos. Além disto, o grande diferencial destes sistemas está na grande quantidade de participantes envolvidos, em diferentes países e diferentes domínios administrativos, o que os torna vulneráveis a atacantes e permitem a rápida disseminação de problemas de segurança.

Neste artigo apresentamos as vulnerabilidades e os incidentes de segurança a que estes sistemas estão sujeitos e as contra-medidas que podem ser utilizadas para minimizá-los. Na seção 2 serão conceituados os sistemas de grade os sistemas de computação voluntária, seus componentes e sua evolução. Na seção 3 serão apresentadas as principais ameaças a que estes sistemas estão sujeitos e as contra-medidas que podem ser aplicadas para evitá-las. Na seção 4 serão apresentados alguns sistemas e o histórico de vulnerabilidades e incidentes de segurança por eles enfrentados. Na seção 5 faremos uma análise dos dados levantados na seção 4 e, finalmente, na seção 6 serão apresentadas as conclusões.

1.1. Trabalhos relacionados

Os novos desafios de segurança apresentados pelos sistemas de grade tem sido objetivo de estudo de vários autores. Em [Chakrabarti et al. 2008] é apresentada uma taxonomia dos principais problemas encontrados referentes a infraestrutura, ao gerenciamento e a arquitetura destes sistemas. Em [Martin and Yau 2007] são apresentados alguns modelos de segurança em grade incluindo dois estudos de casos, um baseado na arquitetura

GSI (*Grid Security Infrastructure*) e outro aplicado a sistemas de grade de computação voluntária, especificamente o projeto climateprediction.net. Em [Stainforth et al. 2004] é detalhada a arquitetura de software deste projeto e são apresentadas algumas das ameaças enfrentadas pelos voluntários e pelos projetos.

2. Grades

O termo grade pode ser definido como um tipo de sistema paralelo e distribuído que permite o compartilhamento, seleção e agregação, de forma dinâmica, de recursos autônomos geograficamente distribuídos, de acordo com a disponibilidade, capacidade, desempenho, custo e necessidades de qualidade de serviço do usuário [Buyya 2002]. A computação em grade envolve a utilização de recursos dispersos em diferentes domínios administrativos, cada qual com suas próprias regras e políticas. Estes recursos podem ser, por exemplo, capacidade de processamento, capacidade de armazenamento e equipamentos específicos.

Desde a sua definição os sistemas de grade sofreram um rápido e contínuo desenvolvimento. Várias aplicações surgiram e várias áreas de pesquisa desenvolveram seus próprios conceitos específicos, de acordo com a área de atuação. É possível hoje encontrar Grades Computacionais, Grades de Dados e Grades de Serviços, entre outras.

As Grades Computacionais correspondem a sistemas que possuem grande capacidade computacional agregada e que permitem o compartilhamento de ciclos de processamento. O objetivo destes sistemas é a integração de recursos computacionais dispersos para prover aos seus usuários uma maior capacidade combinada de processamento. Estes recursos podem ser explorados por meio da execução de uma aplicação em qualquer máquina da grade, do particionamento de uma aplicação em tarefas executadas paralelamente e por meio de múltiplas execuções de uma tarefa em diferentes máquinas da grade.

Grades de Computação Voluntária (*Volunteer Computing Grids*) [Sarmenta 2001], também chamados de Grades de Computação Filantrópica ou Oportunista, são uma subcategoria das Grades Computacionais e correspondem a um tipo de computação distribuída que utiliza os recursos computacionais doados por voluntários que, deliberadamente, cedem os ciclos ociosos de seus computadores para contribuir para determinados projetos. Os projetos executados neste tipo de grade possuem como características a possibilidade de serem divididos em múltiplas tarefas executadas de forma independente e a necessidade de pouca comunicação, pouca interação, grande poder computacional e/ou grande capacidade de armazenamento.

O sucesso de execução dos projetos está diretamente ligado a quantidade de voluntários participantes. Para isto, além da necessidade de ter forte apelo público, alguns projetos incentivam a participação por meio de esquemas de recompensa baseados em créditos. Os créditos correspondem a medidas numéricas atribuídas a um voluntário em recompensa a quantidade de computação executada por seu computador que pode ser contabilizada utilizando como parâmetros o tempo de CPU, o espaço de armazenamento no disco e as taxas de transferência da rede. Os créditos obtidos com a participação nos projetos são individuais porém é possível aos voluntários se associarem a times para obterem estatísticas agregadas. Outras razões que levam a participação voluntária são: vontade de participação em causas científicas, desejo de contribuição para o avanço de um campo específico de estudo, ligação emocional com projetos de descoberta de cura de doenças, benefícios pessoais e desejo de reconhecimento.

Os componentes básicos destas grades são:

Voluntários: pessoas que doam o poder de processamento e/ou a capacidade de armazenamento de seus computadores, denominados **computadores voluntários**, para a participação em um ou mais projetos de pesquisa científica;

Pesquisadores: pessoas que desenvolvem projetos de pesquisa científica e utilizam os sistemas para executá-los;

Projetos ou Aplicações: sistemas independentes de computação distribuída que utilizam as capacidades da grade para serem executados;

Sistemas de grade ou *middlewares*: plataformas de softwares que oferecem um ambiente homogêneo e robusto aos pesquisadores e tornam transparentes as características de heterogeneidade do ambiente grade. Permitem o compartilhamento dos computadores voluntários para a participação em vários projetos.

O esquema básico de relacionamento existente entre estes componentes é apresentado na Figura 1. Para participar da grade um voluntário deve instalar em seu computador o software cliente disponível na página Web do sistema de grade. Este software, ao ser executado, verifica se há atualizações disponíveis e obtém a lista de projetos. O voluntário pode selecionar então com qual projeto deseja colaborar e, após autenticar-se, passa a receber as tarefas deste projeto. Após as tarefas serem executadas os resultados são enviados ao servidor do projeto e novas tarefas são recebidas. O servidor do projeto calcula a quantidade de créditos atribuída a cada voluntário e envia estas informações ao servidor do sistema de grade que disponibiliza em sua página Web estatísticas acumuladas referentes a participação dos voluntários e de seus times.

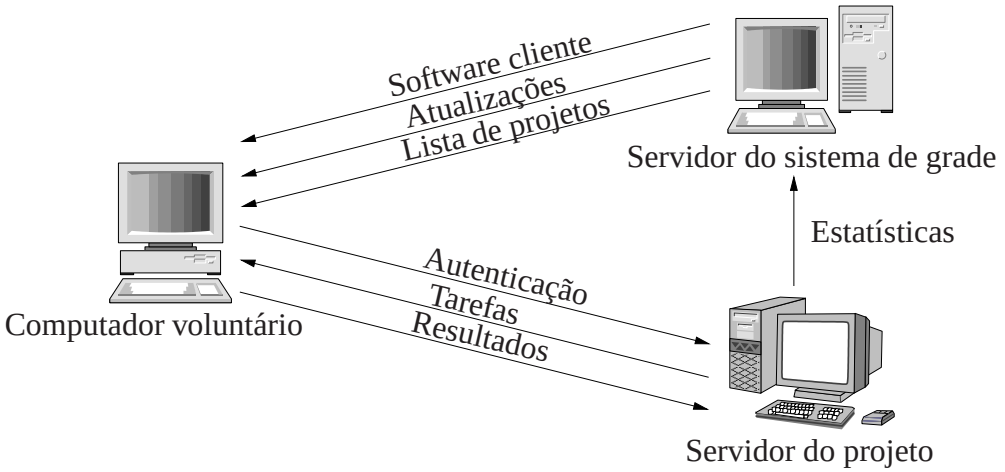

Figura 1. Componentes de grades de computação voluntária

Os sistemas de grade de computação voluntária podem ser divididos em duas gerações. A primeira geração corresponde a sistemas, como o GIMPS [GIMPS 2009], o Distributed.net [Distributed.net 2009a] e o SETI@home Classic [Berkeley 2009], que possuem estrutura monolítica e são desenvolvidos para a execução de projetos específicos, onde computação científica e infraestrutura de computação distribuída são combinadas em um único programa. A segunda geração corresponde a sistemas, como o BOINC [BOINC 2009b], o XtremWeb [XtremWeb 2009] e o JoiN [Yero et al. 2005], que oferecem uma infraestrutura de computação distribuída independente da computação científica.

3. Segurança

Nesta seção apresentaremos as principais vulnerabilidades e incidentes de segurança aos quais os sistemas de grade de computação voluntária estão expostos e algumas contra-medidas para minimizá-los ou evitá-los.

Podemos conceituar uma vulnerabilidade de segurança como uma falha de projeto, implementação ou configuração de um software ou sistema operacional que, quando explorada por um atacante, resulta em um incidente segurança. Um incidente de segurança corresponde a qualquer evento adverso, confirmado ou sob suspeita, relacionado à segurança de sistemas de computação ou de redes de computadores [CERT.br 2006]. A possibilidade de ocorrência de um incidente de segurança pode ser eliminada ou mitigada através da implementação de contra-medidas.

3.1. Vulnerabilidades

Utilizando como base a classificação apresentada pelo CWE (*Common Weakness Enumeration*), projeto desenvolvido pelo MITRE [MITRE 2009], podemos citar as principais vulnerabilidades de um sistema de informação: utilização de senhas fracas ou compartilhadas (CWE-521), falta de atualização de softwares (CWE-671), configuração incorreta de aplicativos (CWE-16), ausência de *firewalls* (CWE-654), alteração de software (CWE-513) e hardware (CWE-2), execução de serviços desnecessários nas máquinas (CWE-272), dados transferidos em aberto (CWE-319), softwares com erros de programação (CWE-17) e instalação de softwares inseguros (CWE-319).

As grades de computação voluntária, além de herdarem estas vulnerabilidades, introduzem novos aspectos problemáticos como ampla distribuição, existência de múltiplos domínios administrativos, heterogeneidade dos computadores participantes, controles de recompensa por créditos, anonimato dos voluntários e grande quantidade de projetos. Além disto, a maior quantidade de recursos, usuários e projetos envolvidos aumenta a quantidade e a velocidade de disseminação destes problemas.

Algumas das vulnerabilidades introduzidas por estas grades advêm da fragilidade das relações de confiança existentes entre seus componentes. Voluntários necessitam de mecanismos que garantam que, ao oferecer seus computadores, não estarão comprometendo sua privacidade, a integridade de seus dados e a disponibilidade de seus recursos. Por outro lado, pesquisadores precisam confiar na exatidão dos resultados obtidos e na garantia de sigilo dos dados do projeto. Os sistemas de grade precisam lidar com grande quantidade de pesquisadores e necessitam confiar nos projetos por eles desenvolvidos sob pena de ter sua reputação comprometida.

3.2. Incidentes de segurança

A exploração das vulnerabilidades pode resultar nos seguintes incidentes de segurança:

- Comprometimento de senhas: por meio de senhas compartilhadas, fracas, trafegadas em texto claro ou obtida através da instalação de *keyloggers*;
- Interceptação de dados: técnicas como *sniffer* e *man-in-the-middle* podem resultar em acesso não autorizado a informações e na modificação de uma comunicação de rede;
- Erros de softwares: a exploração de falhas de codificação como *buffer-overflow*, *race-condition* e falta de verificação de argumentos permite elevação de privilégios e acesso não autorizado;

- Softwares inseguros: programas obtidos de fontes duvidosas podem conter códigos maliciosos e executar operações indevidas;
- Técnicas de varreduras: ferramentas como Nmap e Nessus permitem a atacantes descobrir portas de serviços ativos e explorar suas vulnerabilidades;
- Negação de serviço: interrupção de funcionamento de serviços de um computador ou sistema, por meio da saturação de seus recursos;
- Vazamento de informações: furto de dados de voluntários, pesquisadores e projetos;
- Falsificação de resultados: envio de resultados falsos pelos computadores voluntários com objetivo de atrapalhar os resultados gerais dos projetos;
- Falsificação de créditos: falsificação do tempo de processamento gasto na execução de uma tarefa, pela instalação indevida do software cliente e pela produção de resultados;
- Furto de arquivos dos projetos: computadores voluntários podem furtar arquivos de dados dos projetos, recebidos como base para a execução das tarefas;
- Abuso dos computadores voluntários pelos projetos: disponibilização pelos sistemas de grade de projetos que, intencionalmente ou acidentalmente, abusam dos computadores voluntários, tornando-os inoperantes, deletando arquivos ou acessando informações;
- Distribuição de projetos maliciosos: disponibilização de projetos maliciosos adicionados ao sistema de grade por meio de invasão de seu servidor;
- Instalações indevidas: instalação não autorizada do software cliente.

3.3. Contra-medidas

A possibilidade de ocorrência destes incidentes de segurança pode ser mitigada ou evitada através da implementação, separada ou em conjunto, de contra-medidas. As contra-medidas apresentadas a seguir referem-se a segurança dos computadores (voluntários e servidores), aplicações e sistemas de grade, e podem ser utilizadas tanto nos sistemas de grade de computação voluntária como em sistemas de informação em geral.

1. Instalação de *firewalls*: permite reduzir informações disponíveis externamente sobre uma rede ou computador, bloquear ataques a vulnerabilidades e a liberação controlada de acesso externo aos serviços;
2. Desativação de serviços desnecessários: a redução do número de serviços executados na máquina reduz a possibilidade de existência de vulnerabilidades não resolvidas;
3. Utilização de canais seguros: permite garantir, via criptografia, a autenticidade, privacidade, integridade e não-repudiação da informação trafegada;
4. Auditoria de logs: o registro de eventos de segurança, exceções e atividades de usuários permite o rastreamento de problemas e a identificação da autoria de uma ação;
5. Instalação de correções de segurança: permite eliminar vulnerabilidades conhecidas e já corrigidas pelos fabricantes de softwares;
6. Técnicas de programação segura: a utilização de técnicas de programação segura e de ferramentas de verificação de código, como Flawfinder e ITS4, permitem que falhas no desenvolvimento dos softwares sejam detectadas e corrigidas;
7. Política de *download*: softwares devem ser baixados apenas de fontes confiáveis e com verificação de integridade por meio de algoritmos de *hash* como os da família SHA;
8. Anti-vírus: instalação e atualização de softwares anti-vírus;
9. Política de senhas: implementação de políticas que definam regras para formação, proteção e uso de senhas;
10. Política de acessos: implementação de políticas de privilégio mínimo;

11.Técnicas de replicação e métodos de validação: utilização de técnicas de *majority voting* e *spot-checking* [Sarmenta 2002] permitem a detecção de tentativas de falsificação de resultados e créditos;
12.Restrição de tamanho de arquivos: utilização de mecanismos, como certificados de *upload* [BOINC 2009c], permitem restringir o tamanho de arquivos de dados recebidos e assim evitar saturação de recursos;
13.Utilização de *sandbox* baseados em contas: utilização de mecanismos, como *sandbox* ou *jail*, evita o acesso indevido a dados dos computadores voluntários;
14.Seleção criteriosa de projetos: participação em projetos de código aberto e de instituições bem conhecidas;
15.Assinatura de código: permitem a checagem da autenticidade de arquivos;
16.Inclusão de passos manuais na instalação do software cliente: a necessidade de intervenção manual evita instalações automáticas;
17.Instalação automática de atualizações: permite que novas versões do software cliente sejam automaticamente atualizadas.

A Tabela 1 relaciona quais contra-medidas podem ser aplicadas a quais tipos de incidentes de segurança.

Incidente de segurança	Contra-medidas
Comprometimento de senhas	3, 4, 8, 9
Interceptação de dados	3
Erros de softwares	4, 6, 17
Softwares inseguros	4, 7
Técnicas de varreduras	1, 2, 4, 5,
Negação de serviço	1, 2, 4, 5, 12, 17
Vazamento de informações	3, 4, 5, 8, 10
Falsificação de resultados	3, 4, 11
Falsificação de créditos	3, 4, 8, 11
Furto de arquivos dos projetos	2, 3, 4, 5
Abuso dos computadores voluntários pelos projetos	4, 7, 13, 14
Distribuição de projetos maliciosos	15
Instalações indevidas	16

Tabela 1. Incidente de segurança e Contra-medidas

Devido à sua abrangência mundial e a velocidade com que um incidente pode se alastrar utilizando o ambiente grade algumas contra-medidas preventivas, não específicas e de vital importância, podem ser tomadas:

- Disponibilização de canais efetivos de comunicação que permitam que vulnerabilidades e incidentes, tanto no sistema de grade como nos projetos, sejam reportados e rapidamente tratados como, por exemplo, disponibilização de endereços de email ou formulários em páginas Web e a criação de grupos de discussão;
- Criação de procedimentos de resposta a incidentes que permitam uma ação rápida;
- Contato com grupos de segurança nacionais que podem agilizar o contato com os demais países.

4. Vulnerabilidades e Incidentes de segurança

Nesta seção apresentaremos o histórico das principais vulnerabilidades e incidentes de segurança envolvendo as grades de computação voluntária. As vulnerabilidades descritas

basearam-se em dados públicos contidos no dicionário CVE (*Common Vulnerabilities and Exposures*) e referem-se apenas a vulnerabilidades dos sistemas de grades incluindo suas interfaces Web, projetos e softwares clientes. Os incidentes descritos são baseados em pesquisas feitas na Web, em grupos de discussão, listas de emails e anúncios nas páginas dos projetos.

4.1. SETI@home Classic

O SETI@home Classic [Berkeley 2009] é um experimento científico iniciado em 1999 na Universidade da Califórnia, em Berkeley, com objetivo de utilizar a capacidade ociosa de processamento dos computadores voluntários para busca de inteligência extraterrestre. Em 2005 foi migrado para o BOINC e passou a ser denominado SETI@home/BOINC. É considerado o projeto de computação voluntária mais popular e um dos primeiros a se difundir na rede. Atualmente possui cerca de 1 milhão de voluntários, 56 mil times e 2.3 milhões de computadores voluntários, distribuídos por cerca de 250 países [BOINC 2009a].

Vulnerabilidades

- *Buffer overflow* no software cliente que, se instalado com o bit setuid ligado, permite a usuários locais executar códigos arbitrários. Tipo: erros de softwares. Impacto: acesso a conta de usuários; violação de confidencialidade, integridade e disponibilidade; vazamento de informações e negação de serviço (CWE-120) [CVE-2001-1553 2001] (VUL-1);
- *Buffer overflow* no software cliente que permite a atacantes remotos interromper a sua execução e executar códigos arbitrários, por meio de respostas longas recebidas de servidores clonados. Tipo: erros de softwares. Impacto: acesso a conta de usuários; violação de confidencialidade, integridade e disponibilidade; vazamento de informações e negação de serviço (CWE-120) [CVE-2003-1118 2003] (VUL-2);
- Scripts de inicialização do software cliente executam programas pertencentes aos usuários com privilégios de root, o que permite que usuários elevem seus privilégios por meio da modificação destes programas. Tipo: erros de softwares. Impacto: comprometimento de root; violação de confidencialidade, integridade e disponibilidade; vazamento de informações e negação de serviço (CWE-264) [CVE-2004-1115 2004] (VUL-3).

Incidentes de Segurança

- Alteração da freqüência do processador de computadores voluntários fez com que cálculos errados fossem gerados ocasionando falsos resultados. Tipo: alteração de hardware. Impacto: falsificação de resultados [Anderson 2001] (INC-1);
- Alterações no software cliente para que fosse executado mais rapidamente, para que gerasse falsos resultados e para aumentar a quantidade de trabalho realizado. Tipo: alteração de software. Impacto: falsificação de créditos e resultados [Anderson 2001] (INC-2);
- Instalações indevidas em máquinas de universidades e locais de acesso público, sem autorização prévia dos donos dos computadores. Tipo: instalação indevida. Impacto: falsificação de créditos [Anderson 2001] (INC-3);

- Desenvolvimento de códigos maliciosos que baixam e instalam o software cliente e o configuram para executar tarefas utilizando a conta do seu desenvolvedor. Tipo: instalação indevida. Impacto: falsificação de créditos, exposição do computador voluntário [McAfee 2001] (INC-4);
- Falha no mecanismo de controle de créditos que utilizava o endereço de email do voluntário, informado na criação de sua conta no sistema, e que permitia a atribuição indevida de créditos. Tipo: erros de softwares. Impacto: falsificação de resultados [SETI@home 2001] (INC-5);
- Colaboradores do sistema receberam um email onde era informada uma página Web que disponibilizava para *download* arquivos que continham informações pessoais obtidas por meio do envio de mensagens falsas de resultados aos servidores do SETI@home. Juntamente com os resultados era enviado um arquivo com informações pessoais do colaborador. Caso estas informações estivessem desatualizadas o servidor enviava em resposta as informações atualizadas. Tipo: erros de softwares. Impacto: falsificação de créditos e vazamento de informações [SETI@home 2001] (INC-6).

4.2. BOINC

A idéia de expandir o poder de processamento oferecido pelo SETI@home para projetos de outras áreas de pesquisa fez com fosse desenvolvido, em 2005, o BOINC (*Berkeley Open Infrastucture For Network Computing*) que é uma plataforma aberta para projetos de processamento distribuído, desenvolvido pela Universidade da Califórnia, em Berkeley. Entre os projetos executados, além do SETI@home podemos citar: World Community Grid (pesquisa de curas para doenças e causas humanitárias), Einstein@home (busca por pulsares) e ClimatePrediction (estudos sobre mudanças climáticas). Atualmente o sistema possui cerca de 1.7 milhão de voluntários, 81 mil times e 4 milhões de computadores voluntários, distribuídos por cerca de 277 países [BOINC 2009a].

Vulnerabilidades

- *Cross-site-scripting* (XSS) no fórum do sistema que permite a atacantes remotos injetar Web Scripts arbitrários ou códigos HTML. A exploração desta vulnerabilidade coloca em risco a integridade das máquinas voluntárias que venham a acessar este fórum através da execução de códigos maliciosos introduzidos pelo atacante além de comprometer a imagem e a confiabilidade do projeto. Tipo: erros de softwares. Impacto: modificação não autorizada (CWE-79) [CVE-2007-4899 2007] (VUL-4);
- Falta de checagem do valor de retorno da função "*RSA_public_decrypt*" do OpenSSL que permite a atacantes remotos contornar a validação de cadeias de certificados por meio de assinaturas mal-formadas do SSL/TLS. Tipo: erros de softwares. Impacto: vazamento de informações (CWE-287) [CVE-2009-0126 2009] (VUL-5).

Incidentes de Segurança

- Reportado no grupo de discussão do sistema que uma pessoa que não havia instalado o BOINC estava recebendo mensagens referentes a execução de arquivos deste projeto. Após análises constatou-se que a máquina havia sido infectada por um vírus que instalava automaticamente o BOINC e executava as tarefas em nome de um colaborador específico. Tipo: instalação indevida. Impacto: falsificação de créditos, exposição do computador voluntário [BOINC 2006] (INC-7).

4.3. Distributed.net

O sistema Distributed.net [Distributed.net 2009a] surgiu em 1997 com o objetivo de utilizar a capacidade ociosa de processamento dos computadores voluntários para a quebra, via método de força bruta, de algoritmos de criptografia. O projeto inicial foi a participação em um desafio proposto pela RSA de quebrar o algoritmo RC5 de 56 bits. Outros desafios resolvidos foram: DES, RC5 de 64 bits, OGR-24, OGR-25 e OGR-26. Seus projetos atuais são o RC5 de 72 bits e o OGR-27 que juntos possuem cerca de 84 mil participantes e 5.4 mil times.

Vulnerabilidades

Não foram localizadas vulnerabilidades deste sistema na base do CVE.

Incidentes de Segurança

- Instalações indevidas do software cliente feitas por um técnico de computação em máquinas de uma universidade [Harrison 2001] (INC-8);
- Vários códigos maliciosos foram detectados que instalavam indevidamente o software cliente com o objetivo de aumentar a participação de seus desenvolvedores nos projetos. Os primeiros casos foram detectados em 1997 porém desde lá já houve cerca de 12 diferentes ameaças detectadas. Tipo: instalação indevida. Impacto: falsificação de créditos, exposição do computador voluntário [Distributed.net 2009b] (INC-9);
- Detecção de instalação de um arquivo malicioso de projeto, não desenvolvido e nem distribuído pela equipe do sistema, que alterava a identificação do colaborador para o do seu desenvolvedor nos resultados enviados ao servidor do projeto. Tipo: softwares inseguros. Impacto: falsificação de créditos [McNett 1998] (INC-10);
- Coleta de endereços de email da base de dados de estatísticas do projeto e utilização destes emails para envio de spam. Tipo: exposição de dados. Impacto: vazamento de informações [Nasby 1999] (INC-11).

5. Análise das vulnerabilidades e incidentes de segurança

Analisando o histórico das vulnerabilidades encontradas pelos três sistemas analisados SETI@home Classic (Tabela 2), BOINC (Tabela 3) e Distributed.net (Tabela 4) podemos observar que:

- Os tipos de vulnerabilidades encontradas concentram-se nas categorias de erros de programação, tanto no software cliente quanto em fórum Web, relacionados a *buffer overflow*, falta de checagem de parâmetros e falta de checagem de retorno de funções;
- Os impactos que afetam diretamente os voluntários são: acesso a contas de usuários; violação de confidencialidade, integridade e disponibilidade; vazamento de informações e comprometimento de root, sendo este último considerado o de maior risco;
- Negação de serviço, nestes casos, corresponde a possibilidade de interromper a execução do software cliente. Nos sistemas de grade, diferentemente do que ocorre nos demais sistemas, este impacto pode ser considerado de risco baixo pois assemelha-se a interrupção que ocorre quando um voluntário retorna o uso do computador e a execução deste software é automaticamente interrompida;

- Modificação não autorizada ocorrida no fórum Web do sistema de grade permite a alteração de seu conteúdo e a inserção de código malicioso que pode comprometer a segurança dos participantes do fórum;
- Considerando o BOINC (que inclui o SETI@home Classic) e o Distributed.net conjuntamente tivemos cerca de 1.784 milhão de voluntários e 4.2 milhões de computadores expostos as vulnerabilidades apresentadas (como o Distributed.net apresenta a quantidade de voluntários porém não apresenta a quantidade de computadores consideramos para cálculo a média de computadores por voluntário no sistema BOINC que é de 2.35);
- A implementação de atualizações automáticas, sem necessidade de interferência do voluntário, minimiza bastante o grau de risco de uma vulnerabilidade ser largamente explorada pois facilita a atualização dos computadores voluntários.

Sistema:	SETI@home Classic
Qtde de voluntários:	1 milhão
Qtde de computadores	2.3 milhões
Vulnerabilidades:	VUL-1, VUL-2 e VUL-3
Tipo:	erros de programação no software cliente
Impacto:	acesso a conta de usuários; violação de confidencialidade, integridade e disponibilidade; vazamento de informações e negação de serviço
Incidentes:	INC-1, INC-2, INC-3, INC-4, INC-5 e INC-6
Tipos:	alteração de hardware, alteração de software e instalação indevida
Impactos:	falsificação de resultados, falsificação de créditos, exposição do computador voluntário e vazamento de informações

Tabela 2. Histórico: SETI@home Classic

Sistema:	BOINC
Qtde de voluntários:	1.7 milhão
Qtde de computadores	4 milhões
Vulnerabilidades:	VUL-4 e VUL-5
Tipo:	erros de programação no software cliente e no fórum Web do sistema
Impacto:	modificação não autorizada e vazamento de informações
Incidentes:	INC-7
Tipos:	instalação indevida
Impactos:	falsificação de créditos e exposição do computador voluntário

Tabela 3. Histórico: BOINC

Sistema:	Distributed.net
Qtde de voluntários:	84 mil
Qtde de computadores	200 mil (calculada)
Vulnerabilidades:	não encontradas
Incidentes:	INC-8, INC9, INC10 e INC-11
Tipos:	instalação indevida, softwares inseguros e exposição de dados
Impactos:	falsificação de créditos, exposição do computador voluntário e vazamento de informações

Tabela 4. Histórico: Distributed.net

Analisando o histórico de incidentes de segurança encontrados observamos que:

- A grande maioria dos incidentes relacionou-se a tentativas de burlar os mecanismos de controle de crédito;
- A falsificação de créditos ocorreu por meio de instalações indevidas, softwares inseguros, alteração de hardware, alteração de software, falsificação de resultados e exploração de erros de softwares;
- As instalações indevidas de softwares clientes por meio de códigos maliciosos, além da falsificação de créditos, permitem a exposição do computador voluntário pois interrompem a execução dos softwares anti-vírus deixando o computador exposto a outras ameaças;
- Instalações indevidas causam danos aos donos dos computadores pois há várias instituições cuja política de uso não permite que seus computadores participem deste tipo de sistema e a execução indevida do software cliente pode representar uma desobediência a estas políticas trazendo punições como cancelamento da conta ou demissão;
- A inclusão de passos manuais na instalação do software cliente diminui a possibilidade de ocorrer instalações indevidas automatizadas pelos códigos maliciosos;
- A falsificação de resultados, além de poder ser utilizada para a falsificação de créditos, causa grandes danos aos projetos pois compromete a confiabilidade de seus resultados.

Apesar da quantidade de vulnerabilidades e incidentes reportados ser relativamente baixa a principal preocupação encontra-se na possibilidade de expansão em grande escala destes problemas de segurança devido a grande quantidade de participantes envolvidos. Por isto uma questão crucial para o sucesso dos sistemas de grade de computação voluntária está na velocidade com que os desenvolvedores dos sistemas conseguem lançar correções e na eficiência dos mecanismos de atualizações automáticas.

6. Conclusões

Os sistemas de computação voluntária oferecem grande capacidade computacional e permitem o desenvolvimento de pesquisas em áreas científica, climática, humanitária e computacional, entre outras. Sem a utilização desta capacidade estas áreas dificilmente conseguiriam evoluir tão rapidamente. O apelo trazido por estas pesquisas e a facilidade de participação fez aflorar o desejo de voluntariado das pessoas. Porém ao oferecer seus computadores, estas pessoas acabam sujeitas a problemas de segurança que podem colocar em risco sua privacidade e a integridade de seus dados.

Estes sistemas, além de herdarem os problemas de segurança dos sistemas de informação, precisam lidar com seus próprios desafios relacionados a grande quantidade de participantes, ao anonimato destes participantes e a existência de diferentes projetos. Para minimizar estes riscos é necessária a implementação de um conjunto de contramedidas, além de medidas preventivas como a disponibilização de canais de comunicação de problemas e o apoio de grupos de segurança que auxiliem e facilitem a contenção de um provável incidente.

O principal diferencial das grades de computação voluntária perante os demais sistemas está justamente na quantidade de voluntários e computadores voluntários participantes o que aumenta a possibilidade de incidentes em escala global e o transforma em um ambiente atrativo para atacantes. Como forma de incentivar a participação de grandes quantidades de voluntários e de mantê-los interessados, os sistemas de computação voluntária implementam mecanismos de créditos. Com o passar de tempo verificou-se

que o espírito competitivo, resultado do desejo de obtenção de altas posições nos rankings de colaboradores e times, fez com que surgissem vários incidentes de segurança. Para conseguir seus objetivos atacantes utilizam falhas de implementação dos sistemas e a disseminação de códigos maliciosos que, entre outras coisas, efetuam instalações dos softwares do sistema nas máquinas infectadas e os configuram para gerar créditos em seu nome.

Este artigo procurou demonstrar as principais ameaças enfrentadas por estes sistemas e as contra-medidas que podem ser aplicadas como forma de minimizá-las. Procurou demonstrar também o histórico das vulnerabilidades surgidas e dos incidentes de segurança ocorridos. A análise destes históricos nos permite identificar quais os problemas reais enfrentados pelos sistemas e utilizar este passado como forma de evitar que os mesmos tipos de erros ocorram novamente no futuro.

Referências

Anderson, D. (2001). Wired Cheaters Bow to Peer Pressure. http://www.wired.com/science/discoveries/news/2001/02/41838.

Berkeley (2009). University of California - The Search of Extraterrestrial Intelligence Project. http://setiathome.berkeley.edu/.

BOINC (2006). Do we have a Boinc virus? http://setiathome.berkeley.edu/forum_thread.php?id=27739&nowrap=true.

BOINC (2009a). All Projects Stats.com. http://www.allprojectstats.com/.

BOINC (2009b). Open-source software for volunteer computing and grid computing. http://boinc.berkeley.edu/.

BOINC (2009c). Security issues in volunteer computing. http://boinc.berkeley.edu/trac/wiki/SecurityIssues.

Buyya, R. (2002). Grid Computing Info Centre: Frequently Asked Questions. http://www.gridcomputing.com/gridfaq.html.

CERT.br (2006). *Cartilha de Segurança para Internet*. Núcleo de Informação e Coordenação do Ponto BR.

Chakrabarti, A., Damodaran, A., and Sengupta, S. (2008). Grid computing security: A taxonomy. *IEEE Security and Privacy*, 6(1):44–51.

CVE-2001-1553 (2001). Vulnerability Summary for CVE-2001-1553. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2001-1553.

CVE-2003-1118 (2003). Vulnerability Summary for CVE-2003-1118. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2003-1118.

CVE-2004-1115 (2004). Vulnerability Summary for CVE-2004-1115. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2004-1115.

CVE-2007-4899 (2007). Vulnerability Summary for CVE-2007-4899. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-4899.

CVE-2009-0126 (2009). Vulnerability Summary for CVE-2009-0126. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-0126.

Distributed.net (2009a). http://www.distributed.net.

Distributed.net (2009b). Trojans, worms and viruses. http://www.distributed.net/trojans.php.

GIMPS (2009). Great Internet Mersenne Prime Search. http://www.mersenne.org/.

Harrison, A. (2001). Is Distributed Computing A Crime? http://www.securityfocus.com/news/300.

ISC (2008). Internet Domain Survey. http://ftp.isc.org/www/survey/reports/2008/07/.

IWS (2009). Internet World Stats - Usage and Population Statistics. http://www.internetworldstats.com/stats.htm.

Martin, A. and Yau, P.-W. (2007). Grid security: Next steps. *Inf. Secur. Tech. Rep.*, 12(3):113–122.

McAfee (2001). W32/Hadra@M. http://vil.nai.com/vil/content/v_99108.htm.

McNett, D. (1998). MacOS Meggs RC5 Security Advisory. http://lists.distributed.net/pipermail/announce/1998/000049.html.

MITRE (2009). Common Weakness Enumeration. http://cwe.mitre.org/.

Nasby, J. (1999). Spam to distributed.net team members. http://lists.distributed.net/pipermail/announce/1999/000071.html.

NIC.br (2008). TIC Domicílios e Usuários - Pesquisa sobre o Uso das Tecnologias da Informação e da Comunicação no Brasil. http://www.cetic.br/indicadores.htm.

Sarmenta, L. F. G. (2001). *Volunteer Computing*. PhD thesis, MIT Department of Electrical Engineering and Computer Science.

Sarmenta, L. F. G. (2002). Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems*, 18(4):561–572.

SETI@home (2001). Security Issues. http://arstechnica.com/archive/2001/0501-1.html.

Stainforth, D., Martin, A., Simpson, A., Christensen, C., Kettleborough, J., Aina, T., and Allen, M. (2004). Security principles for public-resource modeling research. In *WETICE*, pages 319–324.

Top500 (2008). Top 500 Supercomputer sites - TOP500 List. http://www.top500.org/list/2008/11/100.

XtremWeb (2009). The Open Source Platform for Desktop Grids. http://www.xtremweb.net/.

Yero, E. J. H., de Oliveira Lucchese, F., Sambatti, F. S., von Zuben, M., and Henriques, M. A. A. (2005). JOIN: the implementation of a Java-based massively parallel grid. *Future Gener. Comput. Syst.*, 21(5):791–810.