Paulo Simões, Edmundo Monteiro
Editores

Coimbra, 19 de Novembro de 2008

SINO2008

Actas da 4ª Conferência Nacional sobre Segurança Informática nas Organizações

Actas da 4ª Conferência Nacional sobre Segurança Informática nas Organizações

SINO2008

Novembro de 2008

Editores

Paulo Simões
Departamento de Engenharia Informática
Pólo II da Universidade de Coimbra
3030-290 Coimbra, Portugal
E-mail: psimoes@dei.uc.pt

Edmundo Monteiro
Departamento de Engenharia Informática
Pólo II da Universidade de Coimbra
3030-290 Coimbra, Portugal
E-mail: psimoes@dei.uc.pt

ISBN: 978-989-96001-0-2

Design Gráfico da Capa: FBA

Prefácio

A 4ª Conferência Nacional sobre Segurança Informática nas Organizações (SINO 2008) realizou-se no dia 19 de Novembro de 2008, na Universidade de Coimbra.

A SINO 2008 foi organizada pelo Departamento de Informática da Universidade de Coimbra e pelo Centro de Informática e Sistemas da Universidade de Coimbra, com o apoio do CERT-IPN e com o patrocínio da Microsoft e da ANACOM (Autoridade Nacional de Comunicações).

Esta foi a 4ª edição desta conferência, que teve anteriormente lugar na Covilhã (SINO 2005), em Aveiro (SINO 2006) e em Lisboa (SINO 2007). Na sequência das edições anteriores, a SINO 2008 pretende constituir um fórum privilegiado de discussão das principais questões inerentes à segurança informática, congregando investigadores académicos, consultores na área de segurança, administradores de sistemas e tecnologias de informação, empresas e outras instituições de algum modo relacionadas com esta temática. A SINO 2008 promoveu assim os seguintes objectivos: contribuir para a divulgação da investigação científica e tecnológica realizada em Portugal no domínio da Segurança Informática, proporcionando um espaço de debate sobre tendências, metodologias e soluções para melhorar e promover a segurança de sistemas de informação e redes de comunicação; promover a partilha de experiências e práticas de adopção de modelos, técnicas, metodologias e tecnologias de segurança; contribuir para a criação de sinergias entre a Indústria, as Universidades e os Centros de Investigação; contribuir para a educação, sensibilização, divulgação e melhoria da percepção da importância do papel decisivo da segurança para os sistemas informáticos, nas suas diversas facetas.

O programa da SINO 2008 foi composto por três apresentações convidadas (duas provenientes da área empresarial e uma oriunda da área académica) e duas sessões técnicas. Foram submetidos 9 artigos científicos, tendo sido seleccionados 6 para apresentação nas sessões técnicas. Gostaríamos de agradecer aos autores dos artigos submetidos e aos oradores convidados. Agradecermos também a contribuição da Comissão de Programa, pelo seu papel no processo de revisão e selecção dos artigos submetidos, e aos diversos voluntários que ajudaram à organização do evento.

Coimbra, Novembro de 2008

Paulo Simões
Edmundo Monteiro
(Editores)

Comissão Permanente

André Zúquete, Universidade de Aveiro
Edmundo Monteiro, Universidade de Coimbra
Henrique João Domingos, Universidade Nova de Lisboa
Mário Freire, Universidade da Beira Interior
Nuno Neves, Universidade de Lisboa

Comissão de Programa

André Zúquete, Universidade de Aveiro
Alexandre Santos, Universidade do Minho
Carlos Ribeiro, IST-Tagus Park
Edmundo Monteiro, Universidade de Coimbra (Presidente)
Henrique Domingos, UNL
Henrique Santos, Universidade do Minho
João Barros, Universidade do Porto
Joaquim Arnaldo Martins, Universidade de Aveiro
José Luís Oliveira, Universidade de Aveiro
José Augusto Legatheaux Martins, UNL
Manuel Barbosa, Universidade do Minho
Mário Freire, Universidade da Beira Interior
Miguel Pupo Correia, Universidade de Lisboa
Nuno Ferreira Neves, Universidade de Lisboa
Paulo Simões, Universidade de Coimbra
Paulo Sousa, Universidade de Lisboa
Pedro Veiga, FCCN e Universidade de Lisboa
Rui Aguiar, Universidade de Aveiro
Simão Melo de Sousa, Universidade Beira Interior
Vítor Santos, Microsoft Portugal

Comissão Organizadora

Edmundo Monteiro, Universidade de Coimbra
Paulo Simões, Universidade de Coimbra (Presidente)

Conteúdo

A Token-based Reputation Framework

Ricardo Godinho[1], Carlos Ribeiro[1]

[1] Instituto Superior Técnico – Taguspark
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo, Portugal
{ricardo.godinho, carlos.ribeiro}@tagus.ist.utl.pt

Resumo

Num ambiente distribuído, os nós da rede apresentam a necessidade de obter recomendações dos seus pares, no sentido de inferir uma avaliação acerca da reputação de uma determinada entidade. Genericamente, o modo como estes nós recolhem informação de *feedback* conduz a um elevado número de interrogações na rede. No seguimento da investigação efectuada, este artigo introduz um modelo bidireccional, utilizando o conceito de lista de recomendação inversa, que está habilitado a determinar valores de confiança, através de uma quantidade mínima de requisições na rede. Após a concepção deste modelo, adoptou-se uma noção designada de confiança diversificada, com o intuito de evitar as intenções maliciosas de alguns dos pares da rede. Utilizando esta noção, cada par é julgado de acordo com a sua capacidade de recomendação e habilidade no fornecimento de serviços ou recursos. Os resultados experimentais permitem obter conclusões em relação ao modelo bidireccional e à noção de confiança diversificada. Primeiramente, são evidenciadas as vantagens significativas, em termos de nós interrogados, daquele modelo. Seguidamente, a noção de confiança diversificada é confrontada perante um conjunto de ameaças onde se demonstra que em determinados cenários a rede não é significativamente perturbada pelo mau comportamento dos seus constituintes.

1 Introdução

As redes P2P são sistemas distribuídos descentralizados onde cada participante possui responsabilidades equivalentes. Nomeadamente, permitem que determinado utilizador actue simultaneamente como cliente e servidor. Neste sentido, é intuitiva a percepção de que a viabilidade de uma rede P2P depende essencialmente do nível de colaboração de cada um dos seus constituintes.

Com o intuito de evitar a degradação da rede através do funcionamento malicioso de algumas das suas entidades, a adopção de um esquema de reputação e confiança é essencial. Estes esquemas, tendem a punir as entidades que operam de uma forma prejudicial para a rede ou visam a incentivar os nós ao comportamento cooperativo, fazendo uso das noções de reputação e confiança. A reputação de uma entidade resulta da confiança que os vários pares têm nessa entidade. No sentido de averiguar se determinado agente é digno de confiança, cada nó deverá obter a reputação desse mesmo agente através de requisições efectuadas aos seus nós conhecidos.

Os esquemas de reputação e confiança actualmente mais relevantes, apesar de garantirem uma melhoria no funcionamento dos sistemas distribuídos, apresentam-se reféns de algumas condicionantes, como é o caso da arquitectura de rede. Concretamente, a maioria dos sistemas de reputação e confiança apenas se direcciona para situações onde existe o recurso a mecanismos centralizados ou onde a organização de rede é do tipo estruturada. Por outro lado, os sistemas vocacionados a operar em redes descentralizadas e não estruturadas, como é o caso da Gnutella [10], deparam-se com dificuldades acrescidas aquando da obtenção de um recurso ou de um valor de confiança para determinado nó. Nessa rede, a informação é recolhida dos seus pares através de técnicas de inundação. Tal situação pode conduzir a uma elevada sobrecarga em termos de quantidade de tráfego que circula na rede. Além disso, este ambiente ao fazer uso de um TTL (*Time To Live*) para evitar a propagação infinita de requisições, pode levar a que em situações de baixa densidade de partilha, o nó solicitador não obtenha a informação que desejaria.

Assim, o primeiro objectivo deste artigo passa por conseguir obter um valor de reputação de uma entidade através de uma solução que minimize o número de nós interrogados, conseguindo obter um maior número de caminhos entre a origem e o destino, através de um menor número de *hops*. Para esse efeito, começou-se por

desenvolver soluções de cálculo baseadas em recomendações de nós conhecidos. Numa primeira fase, apenas foram consideradas recomendações no sentido directo. De seguida, o valor de reputação é calculado recorrendo somente a recomendações inversas. Por último, ao se agregar ambos os tipos de recomendações, desenvolveu-se um modelo bidireccional que considera recomendações directas e inversas na mesma solução.

Foi primordial utilizar um sistema que permitisse avaliar as várias soluções desenvolvidas com o intuito de determinar aquela que se aproxima de uma situação óptima. Por permitir soluções centralizadas e descentralizadas, optou-se pela utilização do EigenTrust [1]. Considerando este sistema como meio de comparação, os resultados demonstram a vantagem significativa, em termos de nós interrogados, de uma solução que utilize simultaneamente ambos os modos de recomendação.

No modelo de procura bidireccional, a rede torna-se mais sensível aos nós que de forma maliciosa submetem *feedback* desonesto ou incorrecto. Nesta linha, o segundo objectivo deste trabalho passa por redefinir a noção de confiança, considerando um valor associado à capacidade de estabelecer recomendações e outro destinado à capacidade no fornecimento de recursos ou serviços. É esta delimitação do alcance de confiança que permite enfraquecer o impacto de recomendações mal intencionadas de nós maliciosos. Com base nesta noção de confiança diversificada, os resultados demonstram que a rede se pode tornar praticamente indiferente às recomendações de índole maliciosa.

No presente artigo, começa-se por descrever os ataques e fragilidades, bem como as propriedades e componentes dos sistemas de reputação e confiança (secção 2). A secção 3 corresponde à descrição e caracterização dos sistemas de reputação actualmente mais relevantes, onde se dá ênfase ao EigenTrust. A quarta secção descreve os algoritmos desenvolvidos, partindo da contextualização do modelo bidireccional e da noção de confiança diversificada. A quinta secção apresenta os resultados experimentais obtidos, antes de se proceder à conclusão do artigo na secção 6.

Sendo este artigo resultante de uma dissertação de mestrado, considerações ou resultados adicionais podem ser obtidos através da consulta do documento integral da mesma.

2 Sistemas de Reputação e Confiança

As pessoas interagem diariamente entre elas. Comunicam com familiares, amigos, vizinhos ou outros contactos. Este tipo de relação constitui uma rede, designada de rede social. Nas redes sociais, as pessoas tendem a confiar mais em amigos do que em pessoas totalmente desconhecidas. Neste âmbito, é importante introduzir os conceitos de reputação e confiança. A confiança é por vezes baseada na reputação. Uma pessoa que apresente boa reputação é mais digna de confiança do que outra pessoa cuja reputação seja negativa. Nas redes de computadores, construir um esquema semelhante, baseado na reputação e confiança, é estimulante. O maior desafio prende-se com o anonimato. Ao contrário das redes sociais, os utilizadores das redes de computadores não se observam mutuamente, o que pode conduzir a uma falta de confiança naquilo que terceiros possam dizer [2]. Sendo assim, é necessário um mecanismo de suporte ao estabelecimento de relações de confiança, isto é, sistemas baseados em reputação e confiança [3]. No entanto, existem inúmeras técnicas que podem ser exploradas, com o intuito de danificar o seu funcionamento, quer seja para tirar partido de alguma situação ou simplesmente comprometer o sistema.

Nos sistemas P2P os nós podem ser classificados como egoístas ou maliciosos. Um exemplo concreto de nós egoístas é o caso dos *freeriders*. Estes nós usam recursos sem oferecer nada em troca [6]. Um nó que actue de forma traiçoeira é outro tipo de comportamento indesejado. Alguns pares apresentam um funcionamento correcto durante um certo período de tempo. Após esta fase, tendem a agir de uma forma mal intencionada. Esta técnica tem especial efeito no caso em que uma reputação elevada corresponde a privilégios adicionais [4]. O *whitewashing* consiste num par que de forma voluntária abandona o sistema P2P, voltando a associar-se posteriormente, com uma identidade nova, de forma a libertar a má reputação associada à sua identidade anterior [5]. Em sistemas P2P, é possível que nós mal intencionados enviem acusações falsas, ou forneçam relatórios falsos de forma a afectar um par inocente [6].

Em muitas situações, o dano provocado por um conjunto de nós mal comportados, é significativamente maior do que se estes actuassem isoladamente. Estes nós conspiram contra um nó alvo, tendo como intenção influenciar a opinião externa acerca desse nó. Em esquemas de reputação, a actuação em conluio é extremamente difícil de ser detectada [6].

Idealmente, um sistema de reputação e confiança deve respeitar um conjunto de propriedades [6]. A primeira corresponde ao tipo de *feedback*. A confiança sobre um nó pode basear-se no *feedback* positivo ou

negativo, dos outros nós em relação a esse. É desejável que um esquema de confiança contemple vários tipos de *feedback*. A segunda propriedade corresponde à comunicação e armazenamento. Existe a necessidade que haja um *trade-off* entre o custo de trocar demasiada informação e obter um valor de confiança credível. Do mesmo modo, é fundamental um *trade-off* entre a quantidade de informação de *feedback* que deve ser armazenada para avaliar a credibilidade da transacção e as implicações desse armazenamento em termos de ocupação de memória. A última propriedade consiste no anonimato. Os sistemas de reputação e confiança tendem a proteger a identidade do nó que submete *feedback*. A propriedade de anonimato visa preservar o par que envia *feedback*, de qualquer tipo de retaliação.

Além do respeito pelas propriedades apresentadas, genericamente, a concepção de esquemas de reputação envolve três componentes principais [4]. A primeira componente consiste na recolha de informação. Para determinar o grau de confiança, é fundamental que o sistema tenha a capacidade de recolher informação acerca do histórico comportamental dos vários utilizadores. A recolha de informação pode ser efectuada individualmente por cada par ou através dos vários pares, com base no seu conjunto de experiências. A segunda componente é o *scoring* e *ranking* da reputação. Tendo a informação do histórico de transacções sido recolhida, procede-se ao cálculo de um determinado *score* para a reputação do par pretendido. Isto pode ser feito por um par, por uma entidade centralizada ou eventualmente, pelo conjunto de todos os pares pertencentes ao sistema. Geralmente o valor do *score* corresponde à reputação e é obtido através de uma função geral de *score*. A terceira e última componente do sistema consiste nas acções resultantes. Além de permitir a escolha de nós cooperativos na execução de uma transacção específica, os esquemas de reputação e confiança podem ser usados para incentivar os nós a colaborarem com a rede. Por outro lado, devem punir os nós cujo funcionamento é incorrecto.

3 Trabalho Relacionado

O sistema EigenTrust atribui a cada nó um valor único e global de confiança, com base na história de *uploads* desse nó. Um nó tem em consideração os valores globais de confiança aquando da decisão de um *download* e a rede utiliza esses valores para identificar e isolar nós maliciosos. Este sistema permite que um nó da rede esteja habilitado a calcular a reputação de qualquer entidade, de um modo totalmente descentralizado. Assim, cada nó apresenta uma visão local de confiança que se traduz pelo número de transacções satisfatórias e insatisfatórias que foram estabelecidas com as demais entidades. Essa visão traduz-se no valor s_{ij} que representa a opinião do nó *i* em relação a *j*: $sij = sat(i,j) - insat(i,j)$. Estes valores locais de confiança são normalizados de forma a evitar que um agente atribua, de forma maliciosa, um valor consideravelmente baixo ou elevado a um outro agente da rede:

$$cij = \frac{\max(sij, 0)}{\sum_j \max(sij, 0)}$$

Baseando-se na ideia de confiança transitiva, os valores c_{ij} podem ser agregados. O nó *i* de forma a obter o valor de confiança de *k*, deverá interrogar todos os seus *j* nós conhecidos. Essas opiniões são pesadas pelo valor que *i* ostenta de *j*:

$$tik = \sum_j cij\ cjk$$

Considerando como exemplo uma situação em que um nó *i* pretende obter um valor de confiança para *k*, através da opinião dos seus vizinhos *m*, *n* e *p*: $tik = cim\ cmk + cin\ cnk + cip\ cpk$. Se, segundo *i*, os nós *m*, *n* e *p* são muito reputáveis, esse facto tem repercussões nos valores cim, cin e cip, sendo que as suas opiniões serão mais consideradas. Em contrapartida, a um nó que se atribua um peso pouco relevante, implicará uma menor consideração pela sua opinião.

Paralelamente, o sistema EigenTrust sugere um algoritmo centralizado para determinar valores de reputação de qualquer constituinte da rede. Considera-se que os valores c_{ij} são representados sob a forma de uma matriz C, isto é, C corresponde à matriz $[c_{ij}]$. Por outro lado, definindo $\overrightarrow{ci}$ como o vector local de confiança que contém as opiniões c_{ij} de todos os nós com quem *i* interagiu, pode-se obter o vector de confiança $\overrightarrow{ti}$. O vector $\overrightarrow{ti}$ apresenta o valor de confiança tik: $\overrightarrow{ti} = C^T \overrightarrow{ci}$. Esta equação reflecte apenas a visão de *i* e dos seus nós conhecidos. Com o intuito de incluir um número significativamente alargado de opiniões e assumindo a matriz C como aperiódica e irredutível, é possível obter o vector $\vec{t}$ do modo seguinte: $\vec{t} = (C^T)^x \overrightarrow{ci}$. Para x iterações elevadas, a potência da matriz C tende a estabilizar num determinado valor. Além disso, se cada nó *i* executar o

cálculo apresentado, $\overrightarrow{ti}$ converge para o mesmo valor em todos esses nós. Concretamente, $\overrightarrow{ti}$ corresponde ao vector próprio esquerdo da matriz C. Por outro lado, cada elemento tj do vector $\vec{t}$ quantifica o valor total de confiança atribuído ao nó *j* por parte da rede.

No seguimento das noções enunciadas, e descartando a natureza distribuída das redes P2P, surge um algoritmo centralizado, designado de Basic EigenTrust, para o cálculo dos valores globais de confiança. Assume-se que uma determinada entidade central tem conhecimento dos valores locais de confiança de toda a rede. Esse conhecimento traduz-se pela matriz C. Por outro lado, o algoritmo propõe que é possível efectuar a substituição do vector $\overrightarrow{ci}$ por um vector $\vec{e}$, que representa uma distribuição uniforme de probabilidades sobre os nós da rede, sem colocar em causa a convergência do algoritmo.

Além do EigenTrust, foram considerados outros sistemas de reputação. Todos os esquemas de reputação e confiança analisados estão caracterizados, de acordo com a recolha de informação, *score* e *ranking* da reputação e as acções resultantes, na tabela 1.

Tabela 1 – Caracterização dos sistemas de reputação e confiança.

Sistemas de Reputação	Recolha de Informação	*Score* e *Ranking* da Reputação	Acções Resultantes
EigenTrust	Os *score managers* são responsáveis por calcular a reputação de um par. Um *score manager* é localizado com base numa DHT.	O cálculo da reputação de um nó é efectuado por um conjunto de nós (*score managers)*.	O nó, de entre aqueles que apresentam reputação mais elevada, é seleccionado de forma probabilística.
XRep [8]	A recolha da informação de reputação é efectuada através das mensagens de *Poll*, ao qual os nós respondem com *PollReply*.	Além da reputação dos pares, é atribuído um valor aos objectos. Agrupa os vários votantes pelo seu endereço IP.	O nó que apresente a maior reputação é contactado afim de obter o recurso pretendido.
Credence [9]	Um cliente consulta directamente os seus pares para obter a votação acerca de um determinado objecto.	Os clientes avaliam os votos dos seus pares com o intuito de determinar a credibilidade dos mesmos. São atribuídos pesos aos votos.	Os votos determinam a autenticidade de um objecto. Um objecto é seleccionado dependendo da votação recolhida.
PeerTrust [11]	A informação de reputação é recolhida através da componente *Data Locator*.	O modelo de confiança do PeerTrust baseia-se em transacções recentes para o cálculo do nível de confiança de um par.	A selecção é feita de forma descentralizada. Cada nó decide, através do *Trust Manager*, se um determinado par é confiável.
P2PRep [12]	Tal como os outros sistemas, o P2PRep interroga os seus pares, através de esquemas de *flooding*.	Interroga os seus pares para obter a reputação de um determinado nó. Pode atribuir pesos à opinião desses pares.	Selecciona interagir com o par que tenha sido alvo de uma opinião mais favorável.
TrustGuard [13]	Recolhe informação acerca da reputação de um par, através do *Trust Evaluation Engine*.	O *Trust Evaluation Engine* reúne *feedback* através do protocolo de *overlay*. Engloba um mecanismo de provas de transacção, para se basear somente em transacções efectuadas.	O utilizador solicita a execução de uma acção e o sistema determina se a transacção deve ser efectuada ou não, de acordo com a reputação calculada.

4 Trabalho Desenvolvido

A extensão de uma relação de confiança para fora das partes para as quais foi criada é possível através do conceito de confiança transitiva. Genericamente, a confiança transitiva pode ser descrita do seguinte modo: se a entidade A confia em B, e B confia em C, então A pode obter uma opinião de C através da recomendação exercida por B. A recomendação de B em relação a C designa-se de recomendação directa.

Por outro lado, é possível que uma entidade estabeleça uma relação de confiança com uma outra entidade, igualmente baseada em recomendações, porém sistematicamente diferente. Considere-se que cada nó apresenta não só a lista de todos os agentes que conhece, mas igualmente a lista de todos aqueles que o conhecem. Neste âmbito, o nó A, ao invés de interrogar B, poderia abordar directamente C. O nó C facultaria a A, a lista de todos os nós que apresentam uma opinião de C. Por último, o nó A decidiria confiar em C de acordo com essa recomendação recebida. A essa recomendação designa-se de recomendação inversa.

Nos exemplos apresentados, a entidade A elabora a sua opinião em relação a C através de recomendações recebidas de outras entidades. Na primeira situação a recomendação é feita pelo nó B (recomendação directa), para o qual existe uma relação de confiança directa por parte de A. Em contrapartida, na segunda abordagem é o próprio nó C que indica a A os nós sobre os quais ele se deve basear de modo a formular uma opinião acerca

de C (recomendação inversa). Em ambos os casos, é o caminho A-B-C que permite a A obter uma visão de C. Primeiramente, B é o nó interrogado enquanto que no exemplo seguinte passa a ser C.

Nas situações descritas anteriormente, a origem obtém um valor de confiança com base num caminho que é determinado pelas recomendações que vão sendo recebidas da rede, quer sejam directas ou inversas. Normalmente, a utilização isolada de apenas um destes tipos de recomendação conduz a um elevado número de interrogações na rede. Esta situação é sobretudo preocupante em redes de dimensão elevada, onde no limiar se tem a necessidade de interagir com a totalidade dos nós da rede. No sentido de limitar o número de interacções na rede na obtenção de um valor de confiança, considerou-se a utilização simultânea de ambos os modos de recomendação: directa e inversa. Este modelo bidireccional, para um mesmo *hop*, permite obter um maior número de caminhos, isto é, mais informação útil para o cálculo da reputação de um nó, interrogando uma quantidade bastante menor de nós. Esta conclusão é intuitiva se for dada relevância às seguintes considerações: numa solução directa procura-se estabelecer caminhos entre os vários nós conhecidos pela origem, ou sucessivamente conhecidos de conhecidos, e determinado destino; no caso de uma solução inversa, com base nas recomendações, nós conhecedores do destino, ou sucessivamente conhecedores de conhecedores, pretendem alcançar caminhos até à origem; na solução bidireccional a origem questiona os seus nós no sentido de obter um conjunto de recomendações directas que lhe permita calcular a reputação do destino. Caso não seja possível, passa a considerar a informação no sentido inverso. Assim, seguidamente, pretende-se encontrar caminhos entre os nós conhecidos da origem e os nós conhecedores do destino. Indefinidamente, são considerados mais nós, conhecidos de conhecidos e conhecedores de conhecedores, até se obter a reputação do destino. Neste caso, deixa de haver um ponto único em alguma das extremidades dos caminhos;

O conceito de opinião que está por base às relações de confiança merece uma análise cuidada. De uma forma simplificada, a opinião consiste num valor de entre uma escala previamente definida (ex.: [0;1]). Esse valor numérico traduz a visão externa dos demais acerca de um nó alvo. Tipicamente, um valor elevado indica que o nó é digno de confiança e um valor baixo corresponde à conclusão contrária.

A visão apresentada anteriormente engloba num valor único o grau de confiança de um nó. Neste sentido, o *score* de confiança calculado para um determinado destino representa apenas o seu nível de cooperação com a rede, tanto no fornecimento de serviços e recursos, como nas recomendações que estabelece acerca de outros nós. Contudo, se um nó é correcto a prestar serviços, não é dado adquirido que também o seja a recomendar bons prestadores de serviço. Neste sentido, considera-se relevante definir uma separação da noção de confiança em termos de confiança na recomendação e confiança na prestação de serviços ou recursos.

A delimitação da confiança é a forma encontrada no sentido de punir os nós cujo comportamento é prejudicial para a rede. Sem esta diversificação, nós que facultassem recomendações desonestas ou forjadas não seriam alvo de qualquer retaliação. É por este motivo, que a confiança diversificada é importante, especialmente em modelos que façam uso de recomendações inversas. Numa outra perspectiva, este tipo de confiança permite ainda introduzir uma visão mais realista do comportamento que as várias entidades que integram um sistema distribuído podem apresentar.

O trabalho desenvolvido apresenta-se organizado em 3 fases distintas: modelo bidireccional, confiança diversificada e algoritmos de lista de recomendação.

4.1 Modelo Bidireccional

O trabalho desenvolvido assenta na existência de uma rede composta por um conjunto de nós, aos quais está associado um identificador único. Para esse efeito é necessária a execução de um protocolo designado de IdleProtocol. Cada nó conserva uma lista de recomendação directa e uma lista de recomendação inversa. A lista de recomendação directa de um nó x apresenta os identificadores únicos dos nós conhecidos por x e o respectivo valor de confiança que lhes está associado. Em contrapartida, a lista de recomendação inversa indica os identificadores únicos de todos os nós que conhecem x bem como os valores de confiança atribuídos pelos mesmos a x.

De seguida, considera-se a existência de uma entidade, central e externa à rede, que recolhe todos os valores locais de confiança (listas de recomendação directa dos diversos nós) e executa o algoritmo Basic EigenTrust. Este algoritmo possibilita que a entidade central tome conhecimento do valor global de confiança que cada nó da rede apresenta. De forma equivalente, para obter localmente o valor global de confiança de um determinado nó alvo, cada nó deverá executar o protocolo a que se designa de TransitiveTrust. A implementação deste protocolo contempla três soluções distintas, todas baseadas na noção de confiança transitiva. Numa primeira fase foram desenvolvidas soluções que apenas recorrem a recomendações no sentido directo. Posteriormente,

passou a considerar-se soluções que calculam a reputação do destino utilizando recomendações inversas dos vários nós da rede. Ambas as soluções apresentadas requisitam um elevado número de nós de modo a obter um valor de reputação. Procedendo à união de soluções directas e inversas num único modelo, a quantidade de nós interrogados diminui substancialmente.

A natureza distribuída de um sistema P2P deve evitar a utilização de entidades centralizadas. Nesse sentido, a rede serve-se da entidade central para ganhar autonomia. Numa primeira fase calcula-se o valor global de confiança com base na entidade central como termo de comparação para detectar a convergência das várias soluções. Paralelamente, esta entidade funciona como mecanismo de validação, ao permitir registar o número de caminhos e de nós interrogados até se alcançar a tal convergência. É este registo que legítima o modelo bidireccional como a solução óptima de entre as soluções desenvolvidas.

Finda a operação centralizada, procede-se a um treino que permite descartar a presença da entidade central no exercício do cálculo. Por esse motivo, é de considerar a seguinte divisão: algoritmos de recomendação com entidade central e algoritmos de recomendação sem entidade central;

4.2 Confiança Diversificada

No caso de confiança diversificada, o valor de confiança único é substituído por um valor de confiança associado à recomendação e um valor de confiança associado ao serviço.

Revela-se importante referir que os valores de confiança são normalizados (a soma de todos esses valores, no caso do sentido directo da lista de recomendação de um nó, é igual a 1). Cada um desses valores é a visão local que determinado nó apresenta em relação aos demais. Intuitivamente, aos olhos desse nó, um valor mediano de confiança corresponde a:

$$valor\ mediano\ de\ confiança = \frac{1}{\#\ \text{total de nós conhecidos por parte do nó}}$$

Tipicamente, um nó i atribui a um conhecido j um valor superior ao valor mediano se este for digno de confiança. Por outro lado, se ao j for atribuído um valor inferior ao valor mediano é um indício de que este não merece confiança. São as diferentes combinações de atribuição de valores que traduzem o modo como determinado nó coopera com a rede.

A confiança diversificada enfatiza o comportamento heterogéneo dos vários pares constituintes da rede. Foram estabelecidos quatro tipos de nós: A, B, C e D; sendo que apenas o nó tipo A é de natureza cooperativa. Um nó do tipo A é hábil a exercer recomendações e a prestar serviços, contrastando com os nós do tipo D que são inábeis em ambas as situações. Por outro lado, um nó do tipo B recomenda mal mas presta bons serviços. Por fim, um nó do tipo C é óptimo a recomendar, mas não presta serviços de uma forma correcta. Assim, um nó do tipo A ou do tipo C mantém nas suas listas de recomendação de confiança diversificada valores correctos. Em contrapartida, um nó do tipo B ou do tipo D, ao ser mal intencionado nas suas recomendações, indica valores incorrectos de forma a sacrificar o funcionamento da rede. Os valores atribuídos por parte dos vários nós da rede são apresentados na tabela 2.

Tabela 2 – Tipos de nós: atribuição de valores.

	Valor atribuído à habilidade de recomendar				Valor atribuído à habilidade de fazer serviços			
	A	**B**	**C**	**D**	**A**	**B**	**C**	**D**
A	+	-	+	-	+	+	-	-
B	-	+	-	+	-	+	+	+
C	+	-	+	-	+	+	-	-
D	-	+	-	+	-	-	+	+

\+ valor superior ao valor mediano de confiança.
\- valor inferior ao valor mediano de confiança.

O valor concreto da atribuição feita a um nó reflecte a importância real desse mesmo nó. Isto é, se por exemplo em termos de confiança na prestação de serviços, o nó i indicar que j apresenta um valor pouco inferior ao valor mediano e se i indicar ainda que um outro nó k ostenta um valor muito inferior ao valor mediano, então conclui-se naturalmente que segundo i, o nó k é pior a prestar serviços do que j. Esta evidência serve de suporte ao conceito de valor de punição. Neste contexto, definem-se níveis de punição de acordo com a grandeza concreta dos valores atribuídos. Uma punição severa consiste em conceder valores significativamente baixos, enquanto que uma punição fraca representa a atribuição de valores não muito

inferiores ao valor mediano. De um modo complementar, e fruto da normalização das listas, os nós que não são alvo de punição vêem o seu valor de confiança aumentado.

A utilização dos conceitos de confiança diversificada é materializada através do desenvolvimento de um protocolo ao qual se designa de TrustDiversity, contemplando as três soluções de recomendação previstas neste artigo. Para a execução efectiva deste protocolo, cada nó interessado em obter a reputação do destino indica o número de caminhos que deverá estabelecer entre si e o tal nó alvo. Um caminho representa uma opinião em relação ao destino. Logicamente, um maior número de caminhos corresponde a um maior conjunto de opiniões. O nó de origem agrega as opiniões recebidas, sendo que estas são pesadas de acordo com a habilidade de recomendar dos nós que devolvem tal *feedback*. No sentido de exemplificar esta situação, considere-se uma derivação do exemplo apresentado no ponto 3. Um nó i pretende obter um valor de confiança para o serviço prestado por k, através da opinião dos seus vizinhos m, n e p: $confiança\ de\ i\ no\ serviço\ de\ k = cim\ cmk + cin\ cnk + cip\ cpk$. Os valores cmk, cnk e cpk representam a confiança que os nós m, n e p apresentam na habilidade para prestar serviços por parte de k. As recomendações dos vizinhos de i são influenciadas pela própria confiança que i deposita na capacidade que estes têm em contribuir com *feedback* (cim, cin e cip).

Após a execução do protocolo TrustDiversity, dependendo do número de caminhos definidos, o nó agrega um conjunto de opiniões, favoráveis ou desfavoráveis, em relação ao destino que pretende avaliar em termos de reputação. É nesta fase que é necessário tomar a decisão de confiar ou não confiar no serviço prestado pelo nó de destino. Para esse efeito, considera-se um bloco designado de Decision Factory. Tendo como ponto de partida as recomendações recolhidas pelo nó, esta entidade toma uma decisão de acordo com os critérios seguintes: se o número de opiniões favoráveis é superior ao número de opiniões desfavoráveis, o nó a avaliar é assumido como confiável; se pelo contrário, a maioria das opiniões indica uma baixa reputação para o destino, não se lhe deve depositar confiança; no caso do número de opiniões favoráveis ser igual ao número de opiniões desfavoráveis, deve avaliar-se o quão forte são as opiniões negativas e as opiniões positivas e optar-se por quem exerce mais influência. No sentido de determinar se a decisão tomada é efectivamente correcta, contemplou-se a existência de uma entidade de validação. Após a decisão resultante do bloco Decision Factory, tal entidade tem por objectivo averiguar se o nó de origem verá as suas expectativas defraudadas de acordo com a natureza do nó de destino.

4.3 Algoritmos de Lista de Recomendação

As noções apresentadas ao longo deste artigo culminam na concepção de um conjunto de algoritmos cujo intuito é alcançar os objectivos propostos. Na sua totalidade, foram desenvolvidos nove algoritmos, três para cada um dos tipos de recomendação: directa, inversa e bidireccional; Os algoritmos de recomendação encontram-se divididos em três secções: algoritmos centralizados de confiança única, descentralizados de confiança única e de confiança diversificada. Os algoritmos centralizados têm como condição de paragem a convergência com uma entidade central. Foram implementados três algoritmos distintos: i) algoritmo de recomendação directa de confiança única com entidade central; ii) algoritmo de recomendação inversa de confiança única com entidade central; iii) algoritmo de recomendação bidireccional de confiança única com entidade central;

Tendo como ponto de partida o registo do número de caminhos e do número de nós interrogados, provenientes da execução dos algoritmos de lista de recomendação com entidade central, é possível proceder à implementação de soluções descentralizadas para cada um dos tipos de recomendação. Para esta fase, os algoritmos centralizados foram modificados de forma a permitir um modo de operação independente da entidade central. Assim, os algoritmos de lista de recomendação sem entidade central devem ser alvo de um treino prévio resultante das soluções centralizadas. Um nó, sendo conhecedor do número de caminhos ou de nós que deverá interrogar, à medida que agrega recomendações da rede, consegue determinar, através dos algoritmos descentralizados, se a informação que dispõe é suficiente para fazer uma avaliação correcta do destino pretendido. As três soluções algorítmicas desenvolvidas para esta fase foram: i) algoritmo de recomendação directa de confiança única sem entidade central; ii) algoritmo de recomendação inversa de confiança única sem entidade central; iii) algoritmo de recomendação bidireccional de confiança única sem entidade central;

Em último lugar, os algoritmos de lista de recomendação passam a considerar a noção de confiança diversificada. O número de caminhos até ao destino é a condição de paragem dos algoritmos. Posto isto, decide-se se o destino é de confiança ou não. Desenvolveu-se um novo algoritmo para cada um dos tipos de recomendação previstos neste artigo: i) algoritmo de recomendação directa de confiança diversificada;

ii) algoritmo de recomendação inversa de confiança diversificada; iii) algoritmo de recomendação bidireccional de confiança diversificada;

5 Resultados Experimentais

Neste capítulo pretende-se apresentar os resultados provenientes da execução dos vários algoritmos desenvolvidos. Numa primeira fase demonstra-se os benefícios inerentes à utilização de soluções bidireccionais na obtenção do valor global de confiança. Seguidamente, são analisados diferentes cenários envolvendo os algoritmos de confiança diversificada.

A implementação deste trabalho foi concretizada através do recurso ao simulador PeerSim [19], tendo o Eclipse IDE como ambiente de desenvolvimento. A linguagem de programação utilizada foi o Java (versão 1.5), refém da escolha do simulador. A rede utilizada nos resultados é não estruturada e definiu-se que cada par pode apresentar um de três graus de ligação: fracamente ligado, normalmente ligado ou fortemente ligado. O conceito de grau de ligação serve de mote à ideia de densidade da rede. Esta noção traduz numericamente o número de ligações existentes na rede por tamanho total de rede (ligações/rede).

5.1 Modelo Bidireccional

5.1.1 Procedimento Experimental

Estabeleceu-se como requisito que os nós de diferentes graus de ligação deveriam coabitar numa mesma rede interligada. Nesse sentido, considerou-se uma rede de 100 nós onde se assume a existência de pelo menos um nó pertencente a cada um dos graus definidos. Com o intuito de efectuar simulações para diferentes densidades de rede, fez-se variar o número de ligações existentes de forma a alcançar densidades no intervalo de [10;100[ligações/rede. Este intervalo reflecte uma densidade mínima de 10 ligações/rede e uma densidade máxima de 99 ligações/rede (caso em que todos os nós da rede conhecem todos os outros). Considerou-se ainda um incremento de densidade correspondente a 10 ligações/rede entre cada simulação efectuada (exceptuando o incremento para a densidade máxima).

Para as várias densidades de rede, foram executados 10 ciclos de simulação sendo que em cada ciclo o valor global de confiança é obtido através das soluções algorítmicas de confiança única previstas neste trabalho. Por fim, os resultados de simulação são direccionados para um ficheiro com a finalidade de se proceder ao cálculo da sua média e à sua representação gráfica.

5.1.2 Algoritmos Centralizados e Descentralizados

Estes primeiros resultados experimentais demonstram, em termos de nós interrogados, a vantagem associada a uma solução que utilize simultaneamente recomendações bidireccionais. Numa primeira fase, foram analisados os três algoritmos centralizados de confiança única. Os resultados provenientes da sua execução são aproximados a funções no sentido de se considerar as soluções descentralizadas. Exalta-se o facto de que em ambos os casos o algoritmo Basic EigenTrust apresenta um papel primordial. Para os algoritmos centralizados, funciona como condição de paragem e para os algoritmos descentralizados permite calcular a exactidão do valor obtido pelos mesmos.

Especificamente, começa-se por registar o número de caminhos e de nós interrogados até se alcançar uma convergência com o algoritmo Basic EigenTrust. Essa convergência traduz-se pelo conceito de exactidão. Estipulou-se que cada nó apenas declara o algoritmo como finalizado, assim que obtenha um valor aproximado ao da entidade central. Para esse efeito definiram-se dois níveis que consistem numa exactidão de pelo menos: 70 % e 90 %. Tendo como ponto de partida os valores obtidos com a presença da entidade central, procede-se à sua representação em termos de funções polinomiais, através de um modelo de regressão polinomial. Assim, um nó ao saber a densidade de ligações existentes na rede, e tendo conhecimento, através da regressão polinomial, do número de caminhos a encontrar ou do número de nós que deverá interrogar, está habilitado a calcular correctamente o valor global de confiança de um qualquer nó.

Número de Caminhos O número de caminhos encontrados até se obter um valor coincidente com a entidade central depende da solução de cálculo considerada e essencialmente do *hop* no qual esse valor é obtido. Tipicamente, para um mesmo *hop*, a solução bidireccional consegue alcançar um número de caminhos significativamente superior aos que podem ser encontrados, entre a origem e o destino, em soluções puramente de recomendação inversa ou recomendação directa. No caso de uma densidade de rede elevada,

independentemente da solução algorítmica considerada, a origem tende a obter o valor global de confiança do destino com base no mesmo número de caminhos e apenas considerando o conhecimento proveniente de um único *hop*. Para densidades de rede relativamente baixas, o número de caminhos encontrados até convergência diverge. Tal como seria expectável, numa situação de mais baixa exactidão, são necessários menos caminhos para atingir um valor global de confiança coincidente com a entidade central.

Nós Interrogados Os gráficos que correspondem à aproximação por funções do número de nós interrogados até convergência com o algoritmo Basic EigenTrust estão ilustrados na figura 1.

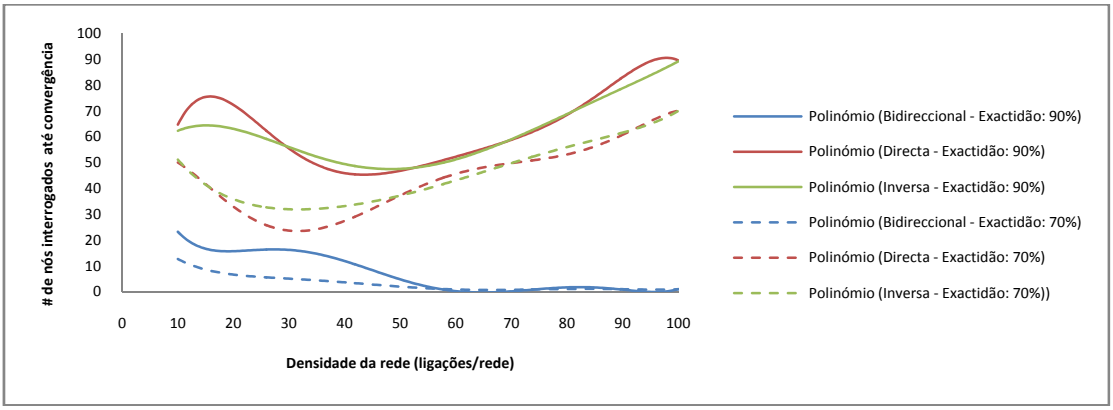

Figura 1 – Número de nós interrogados: regressão polinomial.

Os resultados obtidos para esta fase validam a solução bidireccional como a mais vantajosa para o cálculo da reputação de um determinado nó. O funcionamento dos algoritmos de recomendação directa e recomendação inversa é bastante idêntico, daí se explica a produção de resultados semelhantes. Estes algoritmos apresentam uma tendência equivalente: numa rede com poucas ligações estabelecem-se mais interrogações; numa rede de densidade elevada o número de nós interrogados tende a aumentar. No caso do algoritmo de recomendação bidireccional de confiança única, os resultados obtidos são bastante mais satisfatórios, comparativamente aos restantes algoritmos.

Para densidades de rede superiores a 50 ligações/rede, o algoritmo bidireccional apenas necessita de exercer uma única interrogação. O nó de origem, considerando a sua própria lista de recomendação directa, ao solicitar a lista de recomendação inversa a determinado destino, fica a dispor da informação necessária para proceder ao cálculo do valor global de confiança sem precisar de novas requisições. Por seu termo, os nós ao executarem os algoritmos que utilizam apenas o conhecimento em um dos sentidos, para estes níveis de densidade, apresentam um número de nós interrogados que se aproxima do seu grau de ligação.

Analogamente ao número de caminhos, independentemente da solução utilizada, quando se tolera um erro maior, o número de nós interrogados é menor. Para atingir um valor global de confiança cuja exactidão deve ser superior a 90%, o número de nós requisitados é superior.

Sintetizando, numa rede menos ligada é necessário interrogar um conjunto alargado de nós afim de se poder alcançar o valor global de confiança de determinado nó alvo, dado que apenas alguns nós conhecem o destino. Por outro lado, se o destino é conhecido por um elevado número de nós e se a origem apresenta um elevado grau de ligação, é muito provável que a generalidade dos nós interrogados pela origem consiga contribuir para a obtenção do valor de reputação do destino.

Conclusão dos Resultados do Modelo Bidireccional Os resultados evidenciam que o algoritmo bidireccional permite uma melhoria substancial em termos de interrogações efectuadas na rede. Sendo uma solução que se baseia simultaneamente em recomendações directas e inversas, permite, para densidades de rede elevadas, alcançar um valor global de confiança de elevada exactidão apenas requisitando uma única lista de recomendação. Por outro lado, considerando uma rede onde as ligações residem em menor número, a solução bidireccional consegue ainda assim obter um valor exacto, interrogando menos nós, comparativamente a modos de funcionamento simplesmente directos.

5.2 Confiança Diversificada

5.2.1 Procedimento Experimental

Começou-se por considerar uma rede de 100 nós de densidade igual a 50 ligações/rede. De seguida, tendo por base os diferentes tipos de nós definidos, fez-se variar o número de pares hábeis a fazer recomendações correctas na rede. Inicialmente, a rede exibe um conjunto de 10 nós dignos de estabelecer boas recomendações e gradualmente acresce-se esse valor em 10 unidades, até se atingir uma configuração de rede onde 90 nós recomendam bem e os restantes mal. Estabeleceu-se ainda que para cada uma dessas configurações, o nó que requisita o valor de confiança deve tomar essa decisão de acordo com um número de caminhos predefinidos. Inicialmente, um par julga o destino como confiável ou não confiável, tendo apenas por base uma única opinião recebida dos seus conhecidos. O número de opiniões que o nó deve receber até uma decisão acresce de uma unidade, em cada simulação, até um máximo de 10 opiniões. Cada uma dessas simulações compreende 20 ciclos. Em cada ciclo de simulação o nó que requisita a confiança de um destino é sempre de natureza cooperativa (nó do tipo A). Como destino são considerados 5 nós de cada tipo. Assim, findo os 20 ciclos de simulação sabe-se que 5 nós do tipo A, 5 nós do tipo B, 5 nós do tipo C e 5 nós do tipo D foram julgados em termos de confiança por parte de nós do tipo A.

Se por um lado se faz variar o número de nós hábeis a recomendar, por outro, a rede mantém constante a quantidade de nós que prestam serviços de forma honesta e desonesta (50 nós bons a prestar serviços).

Partindo da execução dos três algoritmos de confiança diversificada desenvolvidos, procedeu-se à análise de um conjunto de resultados provenientes de diferentes cenários experimentais. Os valores contidos nas listas de recomendação dos vários nós (de acordo com a tabela 2) da rede reflectem dois níveis de punição[1]: punição fraca (valor de punição corresponde a metade do valor mediano de confiança) e punição forte (valor de punição corresponde a um décimo do valor mediano de confiança)

Além dos resultados obtidos para cada uma das punições referidas, foram introduzidos os cenários: utilização de um threshold (apenas são consideradas as recomendações de nós que apresentem valores de confiança superiores ao valor mediano) e listas de recomendação inversa forjadas (os nós maus prestadores de serviço devolvem recomendações incompletas e maquinadas como tentativa de enganar o nó que solicita um serviço). O ponto fulcral destes cenários consiste em expor o nó de origem a um conjunto de situações e determinar se ainda assim este consegue decidir correctamente em termos de confiança no serviço prestado por determinado destino. Uma decisão incorrecta consiste em não confiar num bom prestador de serviços ou confiar num mau prestador de serviços. Em contrapartida, uma decisão correcta resume-se a confiar num bom prestador de serviços ou não confiar num mau prestador de serviços.

Punição Fraca Na figura seguinte apresentam-se os cenários de punição fraca e de punição fraca com utilização de um valor de limiar.

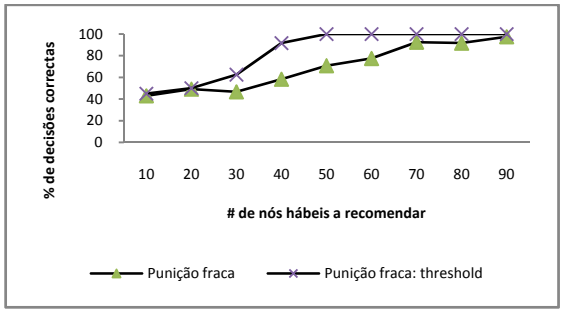

Figura 2 – Punição fraca.

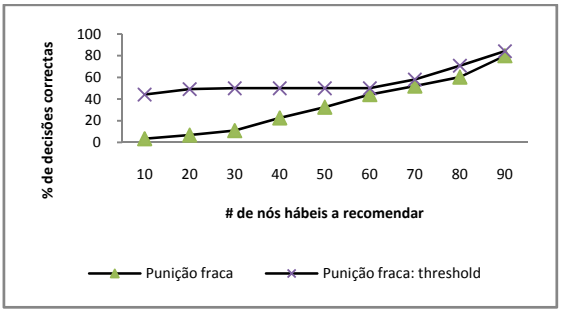

Figura 3 – Punição fraca: lista forjada.

O gráfico da figura 2 demonstra que a percentagem de decisões correctas, para ambos os cenários apresentados, é aceitável. Nomeadamente, sem a utilização de *threshold*, quando o número de nós hábeis a recomendar é superior a 40 a percentagem de decisões correctas supera os 50%.

Com a utilização de *threshold*, esta percentagem é alcançada mesmo para quando o número de nós que estabelecem recomendações correctas é igual a 20.

Tal como se referiu anteriormente, o número de entidades que prestam serviços correcta ou incorrectamente mantém-se constante. Neste sentido, caso o nó de origem, ao invés de tomar uma decisão com base em recomendações da rede, optasse por avaliar o destino de um modo completamente aleatório (decidindo à sorte),

seria expectável que a percentagem de decisões correctas fosse aproximadamente 50, independentemente do número de nós hábeis a recomendar. Por este motivo, um resultado favorável deve superar a percentagem de 50 decisões correctas.

Considerando a situação em que o destino devolve uma lista de recomendação inversa maquinada, o nó de origem tende a tomar um maior número de decisões incorrectas (figura 3).

Punição Forte No presente subcapítulo há um agravamento do valor de punição. Neste sentido, um funcionamento malicioso é penalizado de uma forma mais acentuada. A figura 4 apresenta os resultados dos cenários de punição forte, com e sem utilização de um valor de *threshold*. Os resultados apresentados sugerem uma melhoria em relação ao cenário equivalente de punição fraca. No caso em que o destino devolve uma lista de recomendação inversa forjada, para a situação de punição forte (figura 5), os resultados indicam que não há uma degradação em termos de decisões correctas. O destino ao indicar apenas os valores mais elevados está automaticamente a revelar o conjunto de nós mal intencionados que actuam com o intuito de melhorar a reputação desse destino.

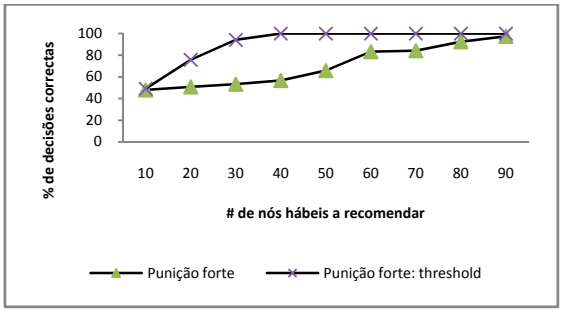

Figura 4 – Punição forte.

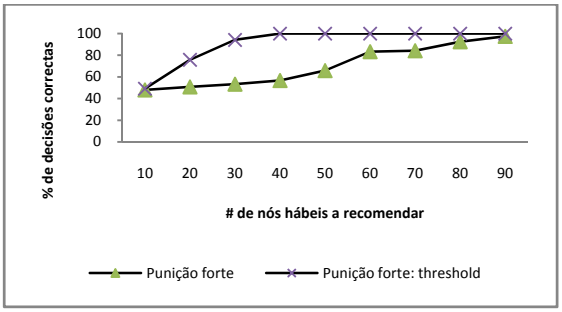

Figura 5 – Punição forte: lista forjada.

Assim, a origem ao receber a lista de recomendação inversa forjada do destino, pesa as opiniões contidas nessa lista de acordo com o respectivo valor de punição. Sendo este valor bastante penalizador, o nó de origem não será induzido em erro e o destino é incapaz de ver o seu valor de reputação aumentado.

Conclusão dos Resultados de Confiança Diversificada Independentemente do cenário considerado, os resultados de confiança diversificada sugerem uma tendência evidente: à medida que o número de nós habilitados a exercer boas recomendações aumenta, a percentagem de decisões correctas é igualmente maior.

Outra conclusão clara prende-se com a utilização de um *threshold*. Um nó que interrogue apenas os seus conhecidos acima de um limiar (ex.: valor mediano de confiança) alcançará uma percentagem de decisões correctas superior ao que conseguiria se interrogasse todos os nós sem critério.

Por último, é de destacar que a omissão de valores na lista de recomendação inversa do destino é bastante prejudicial para o funcionamento da rede. A utilização de um valor de punição elevado permite resolver o efeito danoso causado por estas listas.

6 Conclusão

O primeiro objectivo deste trabalho foi a concepção de um modelo, baseado em relações de confiança transitivas, que funcionasse num qualquer sistema distribuído, independentemente do seu tipo de arquitectura ou organização. Além destes requisitos, tal modelo deveria ser capaz de alcançar um valor de reputação sem implicar um elevado conjunto de interacções. No sentido de concretizar esta ideologia, começou-se por considerar um tipo de recomendação inovador, ao qual se designou de recomendação inversa. Partindo do desenvolvimento de algoritmos de recomendação directa e de recomendação inversa, foi implementado um modelo bidireccional que utiliza ambos os tipos de recomendação previstos neste artigo. Nos resultados obtidos, estão patentes os benefícios desta solução, concretamente no que respeita ao número de nós que se devem interrogar até à obtenção de um valor global de confiança credível.

Tendo conhecimento dos diferentes níveis de colaboração que um agente da rede pode apresentar, considerou-se a necessidade de estabelecer um mecanismo que evitasse a degradação do funcionamento das soluções desenvolvidas, nomeadamente das que fazem uso de listas de recomendação inversa. Assim, foi

estipulado um segundo objectivo no sentido de alcançar um mecanismo que funcionasse como incentivo ao funcionamento cooperativo e penalizasse os nós de índole maliciosa. Sem esta ideologia, estes nós não se sentiriam motivados a fornecer um *feedback* honesto, dado que a omissão de alguma da informação de recomendação poderia fazer elevar o seu valor de reputação. No sentido de encontrar uma protecção contra este tipo de comportamentos, idealizou-se uma noção de confiança bipartida, onde passam a ser considerados os valores de: confiança na recomendação e confiança no fornecimento de recursos ou serviços; Este mecanismo permite pesar as opiniões que vão sendo recebidas com o valor de confiança na recomendação que está associado ao nó que faculta essas mesmas opiniões. Por este motivo, um nó que seja reconhecido pelo seu mau funcionamento será associado a valores de confiança baixos.

Os resultados experimentais apresentados para esta fase demonstram que em certas situações, a utilização de valores de confiança bastante penalizadores pode conduzir a que a rede se mostre praticamente indiferente ao funcionamento malicioso de alguns dos seus constituintes.

Além de combater algumas das técnicas que visam comprometer o funcionamento das redes P2P, a confiança diversificada permite representar de um modo mais realista o comportamento dinâmico que os vários nós de uma rede apresentam.

Referências

[1] S. Kamvar, M. Schlosser and H. Garcia-Molina (2003), "The EigenTrust Algorithm for Reputation Management in P2P Networks", Budapest, Hungary, ACM Press.

[2] C.Yu (2005), "Reputation Propagation between Decentralized P2P Environments", Helsinki, Finland, Seminar on Internetworking.

[3] T. Grandison and M. Sloman (2001), "A Survey of Trust in Internet Applications", IEEE Communications Surveys and Tutorials, London, UK.

[4] S. Marti and H. Molina (2005), "Taxonomy of trust: Categorizing P2P reputation systems", Science Direct, Stanford, California, USA.

[5] K. Lai, M. Feldman, I. Stoica and J. Chuang (2003), "Incentives for Cooperation in Peer-to-Peer Networks", California, USA.

[6] Zhu, B., Jajodia, S. and Kankanhalli, M.S. (2006) 'Building trust in peer-to-peer systems: a review', Int. J. Security and Networks, Vol. 1, Nos. 1/2, pp.103–112.

[7] J.Douceur (2002), "The sybil attack", Microsoft Research.

[8] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati and F. Violante (2002), "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks.", Washington, USA. ACM Press.

[9] K. Walsh and E. Sirer (2006), "Experience with an Object Reputation System for Peer-to-Peer Filesharing", NY, USA.

[10] Gnutella Developer Forum (2003): The Annotated Gnutella Protocol Specification v0.4 website: http://rfc-gnutella.sourceforge.net/developer/stable/.

[11] L. Xiong and L. Liu (2004), "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities", Georgia, USA, IEEE Computer Society.

[12] F. Cornelli, E. Damiani, S. Vimercati, S. Paraboschi and P. Samarati (2002), "Choosing Reputable Servents in a P2P Network", Italy.

[13] M. Srivatsa, L. Xiong and L. Liu (2005), "TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks", Chiba, Japan, ACM Press.

[14] A. Cheng and E. Friedman (2005), "Sybilproof Reputation Mechanisms", NY, USA, ACM Press.

[15] H. Zhang, A. Goel, R. Govindan, K. Mason and B. Roy (2004), "Making Eigenvector-Based Reputation Systems Robust to Collusion".

[16] B. Ooi, C. Liau and K. Tau (2003), "Managing trust in peer-to-peer systems using reputation-based techniques", Singapore.

[17] J. Han and Y. Liu (2006), "Dubious Feedback: Fair or Not?", HongKong, ACM Press.

[18] P. Dewan (2004), "Peer-to-Peer Reputations", Arizona, USA, IEEE Computer Society.

[19] PeerSim. PeerSim website: http://PeerSim.sourceforge.net/.

Nonius, o nível de Segurança da Internet Portuguesa.

F. Rente$^{\phi}$, M. Rela$^{\lambda}$, H. Trovão$^{\xi}$, S. Alves$^{\mu}$
{frente, htrovao, salves}@cert.ipn.pt$^{\phi,\xi,\mu}$, mzrela@dei.uc.pt$^{\lambda}$

CERT-IPN, IPNlis, Instituto Pedro Nunes
Rua Pedro Nunes 3030-199 Coimbra, Portugal

Resumo

O projecto **Nonius** ambiciona produzir um histórico fidedigno de dados indicadores do nível de segurança da Internet Portuguesa. O sistema que o sustenta executa um rastreio, e sucessiva carga de testes, a todo o endereçamento IPv4 alocado a Portugal. Os dados recolhidos passam por um processo de anonimização sendo, posteriormente, publicados segundo um conjunto de métricas pré-definidas.

1 Introdução

No âmbito do projecto **Nonius** foi desenvolvido um sistema computacional que, através de um rastreio ao espaço de endereçamento IPv4 português, produz dados indicadores do estado e do nível de segurança da Internet Portuguesa. Projecto este que se encontra integrado nos serviços de disseminação do CERT-IPN.
O CERT-IPN é um núcleo CSIRT[1], do Laboratório de Informática e Sistemas do Instituto Pedro Nunes (IPNlis), uma instituição de utilidade pública sem fins lucrativos, que tem como missão a transferência de tecnologia entre a Universidade de Coimbra e o tecido económico Português. Os serviços de disseminação são um conjunto de serviços de natureza comunitária, em que o CERT-IPN se afirma como entidade socialmente activa, disposta a contribuir e apoiar o desenvolvimento do conhecimento na área da Segurança de Informação.[3]
Sendo o **Nonius** um sistema de medição do nível de segurança da Internet Portuguesa, um aspecto preliminar para a definição clara do âmbito do projecto é definir o que se entenderá como Internet Portuguesa no seio do **Nonius**. Neste contexto compreende-se como Internet Portuguesa o conjunto de todos os endereços IPv4 alocados a Portugal no RIPE-NCC[4].
Naturalmente esta definição não engloba na totalidade o que realmente é a Internet Portuguesa uma vez que, *p.ex.*, entidades/organizações Portuguesas podem assentar as suas infra-estruturas em endereços IP não alocados a Portugal. Contudo a abrangência da definição usada é suficientemente elevada para dar credibilidade e consistência aos dados estatísticos que o **Nonius** produz.
Os objectivos gerais do **Nonius** podem ser englobados em dois grupos distintos. Num grupo é representada a perspectiva social do projecto, e no outro a perspectiva de Engenharia:

- **Produção de dados indicadores do nível de segurança da Internet Portuguesa** - Sempre com a intenção de criar um histórico fidedigno e consistente, o **Nonius** produzirá iteração[2] após iteração uma série de dados estatísticos relativos a vulnerabilidades técnicas e à presença de *malware* [3] em toda a Internet Portuguesa.
- **Consciencialização Nacional para a problemática gerada à volta da Segurança de Informação** - Esta frase define claramente o objectivo principal do **Nonius** na sua vertente social. A necessidade de uma consciencialização deste tipo é urgente para uma realidade como a que se vive actualmente em Portugal. A título de exemplo, é generalizado o desconhecimento relativo à importância da protecção de dados, à

[1] Computer Security Incident Response Team
[2] Uma execução total do sistema, incluído varrimento e carga de testes.
[3] Sotware Malicioso

necessidade de políticas de segurança de informação, e à facilidade e proliferação dos ataques informáticos contemporâneos. Estes são apenas alguns exemplos das lacunas que o **Nonius** pretende ajudar a reduzir ou minimizar com as suas publicações.

O presente artigo serve como objecto de apresentação e descrição sumária do projecto **Nonius**. Encontra-se subdivido em sete secções, sendo a primeira, onde foi feita uma introdução ao projecto e respectivos objectivo. De seguida é apresentada a arquitectura do sistema computacional que sustenta o projecto. Posteriormente, na terceira secção, são descritas as várias fases de cada iteração. Na quarta secção, são expostos de forma sucinta os vários testes que compõem a actual carga de testes. De seguida, são apresentados os resultados obtidos na primeira iteração do sistema. Por último é explanado um pouco do que será o futuro do **Nonius** e algumas conclusões referentes ao estado actual do projecto.

2 Arquitectura do sistema

O desenho da arquitectura do sistema teve em conta a possibilidade da infra-estrutura técnica, que suporta cada um dos componentes descritos de seguida, estar em localizações geográficas distintas, bem como a possibilidade da existência de mais de um exemplar do componente referido de seguida como *Crawler*. Tal situação levaria à utilização de um ou mais canais de comunicação seguro, p.ex., estabelecido sobre a Internet.
O **Nonius** é constituído por dois componentes conceptuais distintos, o *Crawler* e o *Digester*. A figura 1 representa a visão geral da arquitectura do **Nonius**.

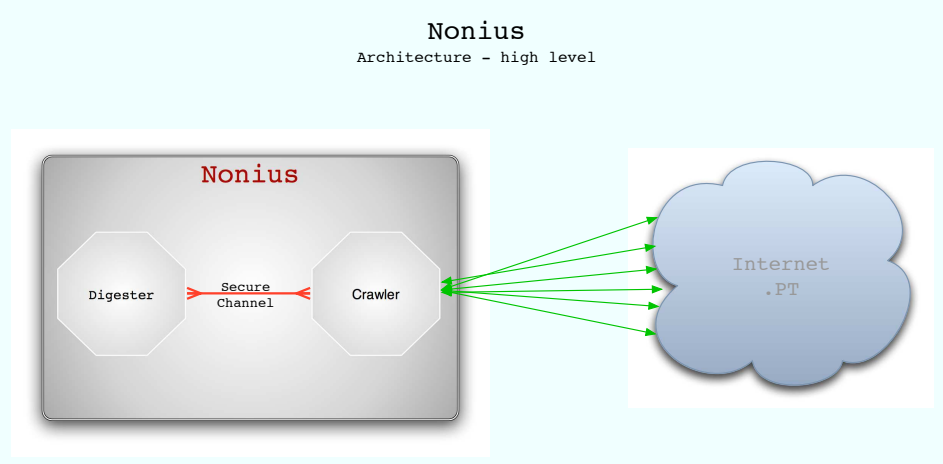

Figura 1: Arquitectura Geral do Nonius.

Atribuem-se ao *Crawler* todas as funções relacionadas com o processo de rastreio, execução e sucessiva recolha de informação da carga de testes. Por outro lado ao *Digester* são atribuídas todas as funções de tratamento e anonimização da informação recolhida, bem como a sua publicação num *web-site*.
Mais pormenorizadamente, o *Crawler* contém os seguintes sub-conponentes: *Network Engine*; *Malware Detection*; *Technical Vulnerabilities Detection*.
O *Network Engine* é responsável pela gestão de todas comunicações geradas durante o processo de rastreio, o módulo de *Malware Detection* implementa os testes de presença de *Malware*, e por último, o módulo *Technical Vulnerabilities* implementa a carga de testes referente às vulnerabilidades técnicas.

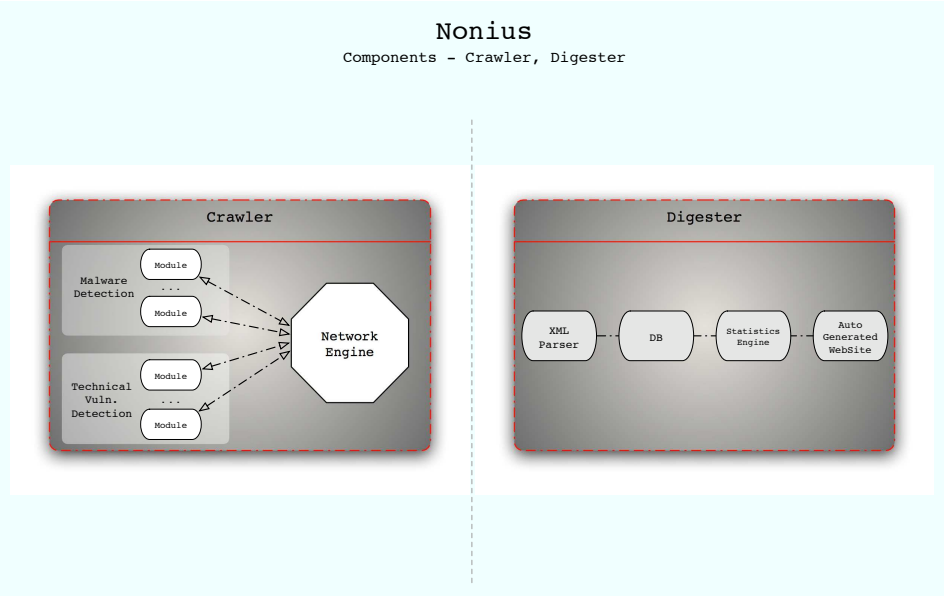

Figura 2: Componentes do Nonius.

Por sua vez, o *Digester* é subdividido nos seguintes sub-componentes: *XML Parser*; *DBMS*; *Statistics Engine*; *Auto Generated Web-Site*. Sendo o *XML Parser* responsável pelo processo de anonimização dos dados XML oriundos do *Crawler* e pela sua inserção na base de dados, o sub-componente *DBMS* (na figura, *DB*) representa o sistema de base de dados onde a informação será armazenada durante o seu processamento e, já agregada e anonimizada, no final de cada iteração do **Nonius**. O *Statistics Engine* é o sub-componente responsável pela produção dos vários dados estatísticos referentes a toda a informação contida no **Nonius**. Por último, o *Auto Generated Web-Site* é o sub-componente responsável pela criação automatizada do web-site onde serão publicados os resultados do **Nonius**. A figura 2 mostra o posicionamento de cada sub-componente no sistema.

3 Fases de cada iteração

Compreende-se por iteração do **Nonius**, o rastreio total a todo o endereçamento IPv4 e respectivos domínios *.pt* pretendidos.
Existem quatro estados distintos em cada iteração do **Nonius**, figura 3: *Host Discovery*, *Tests Payload*, *Data Anonymization*, *Published Data*.
Entende-se como *Host Discovery* o processo de rastreio que permite saber o número de endereços IPv4 vivos [4] num determinado espaço de endereçamento IP. No final deste processo alcança-se o estado ***Host Discovery***, onde existirá uma lista de endereços IPv4 considerados vivos e respectivos domínios *.pt*.
O Processo de *Host Discovery* desempenha um papel crucial na eficácia e abrangência dos resultados do **Nonius**, na medida que é este processo que permite definir a dimensão do âmbito de teste de cada iteração.
Este tipo de processos exigem uma grande optimização e especificação, na medida que têm

[4] Considera-se como *endereço IPv4 vivo* um determinado endereço IPv4 que responda a algum dos processos de sonda que lhe é feito.

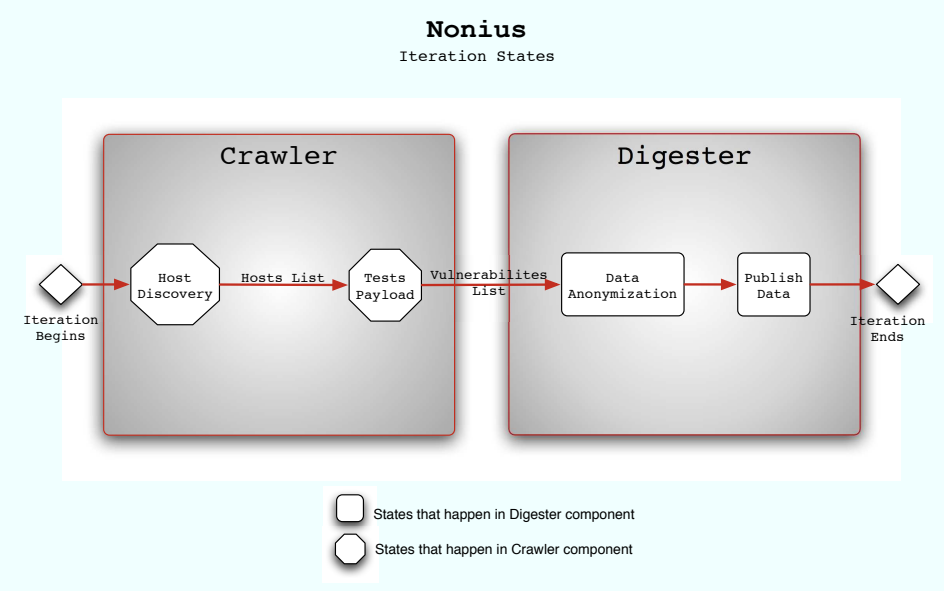

Figura 3: Vários estados de cada iteração.

que ser capazes de lidar com inúmeros (muitas vezes imprevisíveis) cenários. É esta variedade de cenários que pode distorcer os resultados obtidos durante o processo. Situações como a utilização de *traffic shapping*, *NAT*, sistemas de *QoS*, *IDS/IPS*, *firewalls* ou outros sistemas que de alguma maneira interfiram com o tráfego IP em algum ponto do caminho percorrido pelo *flow* em questão, tornam o processo de *Host Discovery* uma tarefa em que é difícil obter resultados cem por cento completos e exactos.
No sentido de tentar contornar estes obstáculos da maneira mais eficiente possível, o processo de *Host Discovery* do **Nonius** foi segmentado em três fases distintas e correlacionadas por uma ordem de preferência, relativa a exactidão e performance. Cada uma das fases distingue-se pela utilização de uma técnica distinta, sendo estas: o *reverse-DNS scan*, o *TCP SYN scan* e o *UDP Empty scan*. Cada uma destas técnicas oferece vantagens e desvantagens largamente conhecidas, de modo que não serão especificadas neste artigo.
A correlação e ordem de preferência dos resultados de cada uma das fases encontra-se descrito na seguinte tabela:

Técnica	**Inclusão dos resultados**
reverse-DNS scanning	apenas se for identificado em pelo menos mais uma técnica
TCP SYN scanning	inclusão directa
UDP Empty scanning	inclusão directa

O segmento referente aos domínios *.pt* é obtido através de, um conjunto de pesquisas a motores de busca [5], de dados oriundos do *reverse-DNS scanning* e, por último, por um conjunto de pesquisas à base de dados *WHOIS* do RIPE-NCC.
Através da lista obtida no estado *Host Discovery*, são executados os testes referentes à presença de *malware* e às vulnerabilidades técnicas. No fim desta carga de testes e após a recolha dos resultados dos mesmos, o **Nonius** encontra-se no estado *Tests Payload*. Neste estado o **Nonius** conta com uma lista das vulnerabilidades encontradas em cada endereço IPv4 testado, portanto, informação não anonimizada.

[5] Motores de busca utilizados: Google, Live Search, Sapo.

Esta lista de pares (Vulnerabilidades, endereço IPv4) passa por uma fase de anonimização, que consiste basicamente na contagem de ocorrências de cada vulnerabilidade no espaço de endereçamento testado, o que resulta num quadro de dados nos formatos: (*Vulnerabilidade X, Numero de ocorrências Y*); (*Vulnerabilidade X, Numero de ocorrências no ISP Y*). Quando este quadro estiver finalizado o **Nonius** encontra-se no estado *Data Anonymization*. Por último são gerados os dados e gráficos estatísticos baseados nas métricas internas, de maneira a ser criado/actualizado o web-site. No final deste processo o **Nonius** está no estado *Published Data*.

4 Vulnerabilidades e Presenças de Malware Testadas

A escolha das vulnerabilidades teve em conta duas preceptivas distintas, uma representativa do que se pode designar por sector organizacional, e outra representativa dos utilizadores caseiros.
Num sentido lato, os principais riscos para o sector organizacional estarão, de uma forma ou de outra, ligados com a fuga/perca de informação. Já para os os utilizadores caseiros, a fuga de informação surgirá paralela à possível utilização das seus computadores para a execução de crimes informáticos a terceiros. Estas duas premissas estiveram por de trás das decisões relativas à carga de testes que o **Nonius** executa.
De seguida é apresentada uma breve descrição das várias vulnerabilidades testadas actualmente pelo **Nonius**.

1 Acesso por SNMP com permissões apenas de leitura, e de leitura e escrita em simultâneo
A possibilidade de aceder a um sistema usando SNMP pode, no mínimo, representar a maneira de um possível atacante obter informação descritiva desse sistema. Dependendo do sistema, as informações obtidas podem variar entre simples identificações de fabricante até à totalidade das configurações usadas no sistema, ou até mesmo à identificação de software que esteja a ser executado localmente. Independentemente do tipo de informação possível de obter, o acesso por SNMP com permissões de leitura é considerada uma vulnerabilidade grave. Se a possibilidade de um atacante ter acesso a informações descritivas e/ou de configuração, de um determinado sistema é considerada uma vulnerabilidade grave, a possibilidade de alteração dessa informação é, certamente, uma vulnerabilidade crítica. Torna-se evidente que a exploração desta vulnerabilidade por parte de um atacante, pode levar facilmente a um compromisso integral da integridade e funcionamento do sistema. Juntando a isto o facto de exploração poder ser remota, torna o acesso por SNMP com permissões de escrita e leitura, umas das vulnerabilidades mais criticas das testadas pelo **Nonius** actualmente.

2 Permissão de transferências de Zona de DNS por AXFR
O AXFR é um protocolo para transferências de zonas de DNS entre servidores de DNS. A possibilidade de terceiros poderem aceder à informação contida numa zona de DNS é considerada uma vulnerabilidade grave. Vulnerabilidades deste tipo podem, entre outros, representar uma fuga de informação suficiente para um atacante identificar a estrutura e topologia da rede associada ao domínio em causa.

3 DNS Snooping
Se um determinado servidor de DNS permitir pedidos não recursivos a terceiros, possibilita a um atacante executar um ataque conhecido como *DNS Snooping*. Este tipo de ataques fornece ao atacante dados sobre a informação contida na *cache* do servidor de DNS, permitindo assim ao atacante, *p.ex.*, saber se determinado domínio foi resolvido recentemente. Numa primeira avaliação este tipo de vulnerabilidade pode ser conotado de um nível baixo

de importância, contudo, se se fizer um enquadramento com a actual dinâmica do mundo empresarial e a sua respectiva dependência das Tecnologias da Informação, essa perspectiva muda: a possibilidade de um atacante obter informação sobre fornecedores de determinados serviços (como *p.ex.* de *backup* remoto, serviço de e-mail externos, servidores de reenvio de E-mail...), ou a possibilidade de identificação de parceiros estratégicos que não sejam de conhecimento público, pode certamente representar um grave fuga de informação.
Supondo que um atacante pretendia atacar determinada entidade (X), que por sua vez requisitava um serviço de *backups* remoto a uma outra entidade (Y). Caso um dos servidores DNS da entidade X tivesse vulnerável a *DNS Snooping*, o atacante poderia identificar facilmente o(s) domínio(s) da entidade Y, e focar o seu ataque nesses domínios. Caso o atacante fosse bem sucedido no ataque à entidade Y, a informação da entidade X seria comprometida.

4 Uso de SSH versão 1.x

O uso das versões 1.33 e 1.5 do protocolo SSH permite a um atacante, que consiga interceptar o fluxo de comunicação, comprometer na totalidade a integridade da informação transitada nessa comunicação.

5 Uso de certificados SSL gerados pelo pacote do OpenSSL Debian com versões vulneráveis, em SSH[6]

No dia treze de Maio de 2008[1], o Projecto *Debian* anunciou a existência de uma vulnerabilidade no pacote do software *OpenSSL* que o projecto publicava. A vulnerabilidade em causa deveu-se a um erro do responsável pelo pacote de software que, ao comentar linhas no código do software em causa, interferiu radicalmente com a capacidade de entropia necessária para os processos de cifragem do *OpenSSL* serem considerados seguros. Mais concretamente, a vulnerabilidade debilitou o *Pseudo Random Number Generator (PRNG)* por completo. O conjunto de chaves possíveis para a cifragem viu-se assim reduzido ao minúsculo número de 32767 possibilidades, tornando assim a "adivinha"das chaves de cifragem um processo trivial, mesmo usando os tão pouco produtivos, processos de força bruta. Um dos serviços em que esta vulnerabilidade pode ter um impacto destruidor, é no *SSH (Secure Shell)*. Um servidor de *sshd* que use chaves geradas pelo pacote *Debian* do *OpenSSL* vulnerável, pode ser comprometido usando técnicas de força bruta em poucos instantes. No caso de sucesso, o atacante fica com permissões Administrador (*root* nos sistemas operativos baseado *Unix*).

6 Uso de Telnet

O uso de Telnet é considerado inseguro, umas vez que os dados em trânsito não usam qualquer tipo de cifra. O que permite a terceiros ter acesso a toda informação que circular nas comunicações geras por este serviço, incluído por exemplo credenciais de autenticação e outros dados confidenciais.

7 SMTP Relay público

Um serviço *SMTP* que permita retransmissão de mensagens para outro domínios que não o ou os que forem da sua responsabilidade, representa uma vulnerabilidade grave. Uma vez que permite a criminosos usarem o serviço em causa para proliferação de *spam*. Embora esta vulnerabilidade não ponha risco a infra-estrutura onde corre, sem contar com o uso indevido de recursos, considera-se uma vulnerabilidade bastante grave uma vez é uma das fontes de *spam*.

8 Uso de Finger

O serviço Finger tem como objectivo fornecer informações sobre os utilizadores do sistema,

[6]Uma vez que quando esta vulnerabilidade foi publicada já se encontrava a decorrer a primeira iteração do **Nonius**, na primeira iteração o teste da mesma está restringido apenas a dados recolhidos aquando dos testes relativos ao uso de *SSH1.1*.

incluído em certos casos se estes se encontram ligados ao sistema.
Este tipo de informação pode ser utilizada por um atacante para, entre outros, obter informação sobre *usernames* para, *p.ex.*, o uso posterior em ataques de força bruta.

9 Acesso público a partilhas de CIFS/SMB
O acesso público a partilhas de CIFS/SMB, vulgo partilhas de *Windows*, para além do evidente acesso directo à informação partilhada, pode ter um papel crucial em determinados tipos de ataques, nomeadamente em processos de *fingerprinting* do sistema.

10 Acesso público a partilhas de CIFS/SMB em que pelo umas delas seja C:\ou C:\windows Se o acesso público a informação que não esteja directamente relacionada com o sistema operativo em si é considerada uma vulnerabilidade grave, sê-lo-á ainda mais se a informação disponível for relacionada com o sistema operativo, onde um atacante pode facilmente obter um perfil altamente detalhado do sistema operativo da máquina em causa.

11 Uso de certificados SSL expirados
O uso de certificados SSL expirados coloca em risco a privacidade dos utilizadores do serviço em causa. Na medida em que facilita e favorece ataques, como por exemplo ataques *Man-In-The-Middle*(MITM)[7].

12 Uso de protocolos consideradas vulneráveis
O uso da versão número dois do protocolo SSL é considerada um risco. O SSL 2.0 detêm várias falhas nos seus processos criptográficos [2]. O seu uso é totalmente desaconselhado uma vez que torna possível a execução de ataques *MITM* e da decifragem, por parte de um atacante, do tráfego gerado por comunicações que façam uso de SSL 2.0.

De seguida serão descritos os teste a presença de *Malware* que o **Nonius** executou na sua primeira iteração. O testes de *Malware* executados cobrem um total de quatro espécies distintas, das quais o **Nonius** é capaz de identificar 12 estirpes diferentes.
De uma forma geral, pode-se afirmar que o grau de certeza dos teste de *Malware* é bastante mais baixo do que os testes de Vulnerabilidades técnicas. Este facto deve-se não só às características furtivas deste tipo de software, que tem como um dos principais objectivos garantir que não sejam detectados, mas também às dificultadas técnicas impostas por razões de foro legal, naturalmente adjacentes a este tipo de processos. Se é tecnicamente difícil detectar a presença de *Malware* num determinado sistema onde se detém total acesso e total permissão de uso, muito mais o será em sistemas onde esse nível de acesso e permissão não existem, como é o caso do **Nonius** e dos sistemas que testa.
Pode-se ainda referir o acréscimo de dificuldade, relativo ao facto de se tratarem de testes remotos, ao invés dos típicos testes locais de *Malware*.
Tendo isto em conta, a filosofia adoptada foi: tentar enquadrar o máximo possível de espécimens de *Malware* e apenas quando for possível fazerem-se testes com maior precisão; garantir a coerência dos resultados finais através de uma adequada distribuição dos pesos a cada um dos testes (quanto mais preciso for o teste, mais peso no resultado final terá).
A escolha dos espécimens a testar teve por base dois factores: vestígios possíveis de detectar remotamente; grau de actividade.
Na lista que se segue são apresentados as quatros espécies de *Malware* testados, acompanhadas da indicação das várias estirpes suportadas pelos teses.

[7] Ataques em que através de adulteração dos dados identificativos dos vários pontos de comunicação, o atacante, consegue passar a interceptar toda a comunicação.

NetSky

Tipo: *worm*

In the Wild:[8] Sim

***Ranking* de preexistência:** 2º lugar no ano de 2007 com 21.94% das infecções detectadas (25161383 infecções); 1º lugar no mês de Abril de 2008 com 21.17%.

Principais métodos de propagação: E-Mail

Sistema Operativo: Windows (95, 98, 98 SE, NT, ME, 2000, XP, 2003

Estirpes detectadas: AA, S, T e Z

Principais Efeitos: Geração de SPAM, modificações no *Registry* do Windows, efectua ataques DoS

Vestígios Remotos: *backdoor* nos portos 665 (estirpes AA e Z) e 6789 (estirpes S e T).

Alias: Symantec: W32.Netsky.Z@mm ; Mcafee: W32/Netsky.z@MM ; Kaspersky: Email-Worm. Win32.NetSky.aa ; TrendMicro: WORM_NETSKY.Z ;F-Secure: W32/Netsky.Z@mm

Traços Históricos:

•Ataque DoS aos domínios: www.nibis.de, www.medinfo.ufl.edu e www.educa.ch (02/05/2004 e 05/05/2004); www.cracks.am, www.emule.de, www.kazaa.com, www.freemule.net, www.keygen.us (14/04/2004 e 23/04/2004)

•Datas de descoberta: NetSky.S -04/04/2004, NetSky.T -06/04/2004, Worm/NetSky.Z -21/04/2004, Worm/NetSky.AA -22/05/2005

MyTob

Tipo: *worm*

In the Wild: Sim

***Ranking* de preexistência:** 3º lugar no ano de 2007 com 16.43% das infecções detectadas (18839787 infecções); 4º lugar no mês de Abril de 2008 com 12.37%.

Principais métodos de propagação: E-Mail, rede local através de exploração das vulnerabilidades expressas nos boletins da Microsoft, MS03-049 e MS04-011

Sistema Operativo: Windows (95, 98, 98 SE, NT, ME, 2000, XP, 2003)

Estirpes detectadas: F, CF

Principais Efeitos: Geração de SPAM, modificações no *Registry* do Windows, bloqueia o acesso a sites de fabricantes de software de Segurança (impedindo assim por exemplo a actualização de software Anti-Vírus), instala outro Malware (nomeadamente *Rootkits* [9]), rouba informação confidencial, torna o sistema num *Zombie* de uma *botnet* baseada em *IRC* (*p.ex.* no canal #.hellbot no servidor bmu.q8hell.org)

Vestígios Remotos: Servidor de *FTP* no porto 10087 e 10487 com o *banner* "220 StnyFtpd 0wns j0".

Alias: Symantec: W32.Mytob.AH@mm ; Kaspersky: Net-Worm.Win32.Mytob.t, Net-Worm.Win32.Mytob.x; TrendMicro: WORM_MYTOB.BW,WORM_MYTOB.BR; F-Secure: Net-Worm.Win32.Mytob.t

Traços Históricos: Datas de descoberta: Mytob.IJ -07-07-2005, Worm/Mytob.CF -27/04/2005

Zafi

Tipo: *worm*

In the Wild: Sim

***Ranking* de preexistência:** 7º lugar no ano de 2007 com 3.00% das infecções detectadas (3437391 infecções); 9º lugar no mês de Abril de 2008 com 3.09%.

Principais métodos de propagação: E-Mail, redes *Peer to Peer*

Sistema Operativo: Windows (95, 98, 98 SE, NT, ME, 2000, XP, 2003

[8]Expressão utilizado para indicar que um determinado espécimen de *Malware* ainda se encontra em proliferação.

[9]Tipo de software malicioso que permite o controlo total do sistema por parte do atacante, de maneira furtiva (ou seja, sem o proprietário se aperceber)

Estirpes detectadas: B, D, F
Principais Efeitos: Geração de SPAM, modificações no *Registry* do Windows, desactiva ou destrói software de protecção.
Vestígios Remotos: instala uma *backdoor* que usa o porto 2121 e 8181 para interacção com o atacante e para actualização do próprio *malware*
Alias: Symantec: W32/Zafi.d@MM, W32.Erkez.B@mm; Mcafee: W32/Zafi.d@MM; Kaspersky: Email-Worm.Win32.Zafi.d, Email-Worm.Win32.Zafi.b; F-Secure: Email-Worm.Win32.Zafi.b
Traços Históricos: Datas de descoberta: Worm/Zafi.D -14/12/2004, Worm/Zafi.B -11/06/2004

Mydoom
Tipo: *worm*
In the Wild: Sim
***Ranking* de preexistência:** 8º lugar no ano de 2007 com 2.49% das infecções detectadas (2850097 infecções); 6º lugar no mês de Abril de 2008 com 5.30%.
Principais métodos de propagação: E-Mail, redes *Peer to Peer*
Sistema Operativo: Windows (95, 98, 98 SE, NT, ME, 2000, XP, 2003
Estirpes detectadas: A, L, G, T
Principais Efeitos: Geração de SPAM, modificações no *Registry* do Windows, desactiva ou destrói software de protecção. Entre 10 e 20 minutos após a sua execução inicia um ataque de *DoS* contra o endereço www.symantec.com (a estripe Mydoom.G).
Vestígios Remotos: instala uma *backdoor* que usa os portos 1042 e 1034 5422 1080 para interacção com o atacante
Alias: Symantec: W32.Mydoom.L@mm, Backdoor.Zincite.A; Mcafee:
W32/Mydoom.n@MM; Kaspersky: Email-Worm.Win32.Mydoom.m;
TrendMicro: WORM_MYDOOM.L, WORM_MYDOOM.M
Traços Históricos: Datas de descoberta: Worm/Mydoom.L.2 em 19/07/2004, Tr/My- doom.BB.1 em 23/05/2006, I-Worm.MyDoom.gen (W32/MyDoom-Gen
ou Win32.Mydoom.S@mm) em 09-03-2004, W32/Mydoom.g@MM
(ou W32.Mydoom.G@mm) em 03-02-2004

Os dados apresentados referentes a *Malware* representam uma compilação de informação feita pelos autores através de recolhas empíricas por Análise de *Malware*, e através das seguintes fonte de informação públicas:

- *Malware prevalence Reports* - http://www.virusbtn.com/
- *Avira Virus Search* - http://www.avira.com/en/threats/
- *MacAfee Threat Center* - http://vil.nai.com/vil/
- *F-Secure Virus Description Database* - http://www.f-secure.com/v-descs/

A tabela 1, apresenta a classificação de cada Vulnerabilidade Técnica, e de cada Teste de presença de *Malware*, em termos de: Nível CVSS[5]; Vector CVSS; Nível de Precisão. O CVSS (Common Vulnerability Scoring System) é, como o próprio nome indica, um sistema métrico de classificação de vulnerabilidades. As suas classificações têm em conta três perspectivas distintas, a *base*, a *temporal*, e a *environmental*. Sendo a *base* responsável pela classificação do impacto, da complexidade e da dificuldade de exploração da vulnerabilidade, a *temporal* pela a avaliação das características que mudam com o passar do tempo, e por sua vez a *environmental*, as características especificadas de ambientes de utilização/execução.

Descrição	Nível CVSS	Precisão	Vector CVSS
SNMP, leitura	5.0	100%	(CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)
SNMP, escrita	7.5	100%	(CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)
DNS Zones, AXFR	5.0	100%	(CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)
DNS Snooping	5.0	90%	(CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)
SSH 1.1	2.9	100%	(CVSS2#AV:A/AC:M/Au:N/C:P/I:N/A:N)
SSH, Debian	10.0	100%	(CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)
Telnet	6.1	100%	(CVSS2#AV:A/AC:L/Au:N/C:C/I:N/A:N)
SMTP Open Relay	5.0	90%	(CVSS2#AV:N/AC:L/Au:N/C:N/I:N/A:P)
Finger	5.0	80%	(CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)
CIFS/SMB shares	5.0	100%	(CVSS2#AV:N/AC:L/Au:N/C:P/I:P/A:P)
CIFS/SMB shares (C:)	7.8	100%	(CVSS2#AV:N/AC:L/Au:N/C:C/I:P/A:P)
SSL Expired	5.7	100%	(CVSS2#AV:A/AC:M/Au:N/C:C/I:N/A:N)
SSLv2	5.7	100%	(CVSS2#AV:A/AC:M/Au:N/C:C/I:N/A:N)
NetSky	7.6	50%	(CVSS2#AV:N/AC:H/Au:N/C:C/I:C/A:C)
MyTob	10.0	65%	(CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)
Zafi	9.3	50%	(CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C)
MyDoom	9.3	50%	(CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C)

Tabela 1: Classificação das Vulnerabilidades Técnicas e dos Teste de presença de *Malware* (CVSS, Precisão e Vector CVSS)

5 Resultados da primeira iteração

Os resultados apresentados na tabela 2 foram obtidos numa população de **3 665 760** endereços *IPv4* e **11 304** domínios *.pt* associados a **9 320** *DNS Servers* distintos.Dessa população foi possível identificar **30 913** vulnerabilidades em **83 306** endereços *IPv4* [10], o que corresponde a 16.1% de um total de **516 213** endereços vivos.
A referida iteração demorou aproximadamente 330 horas (14 dias) a executar na sua totalidade. No referido intervalo de tempo foram executados os vários rastreiros, a respectiva carga de testes e o processamento de todos os dados recolhidos.
A tabela 2, o gráfico presente na figura 4, e o gráfico da figura 5 representam, respectivamente, a distribuição de ocorrências[11] das várias vulnerabilidades técnicas e os dados relativos às presenças de *Malware* detectadas.

O principal resultado do **Nonius** é um valor representativo do nível de Segurança da Internet Portuguesa (***NSIP***). O **NSIP**, tal como todos os resultados do **Nonius**, é um indicador com carácter estatístico, ou seja, é um dado indicador do nível de Segurança da Internet Portuguesa e não um valor exacto do que realmente é o referido nível. Até porque, o cálculo exacto do nível de Segurança de um segmento de endereçamento tão vasto e mutável como o referente a Internet Portuguesa, é difícil de alcançar, senão mesmo impossível. Este valor é obtido através de uma média ponderada dos resultados provenientes da carga de testes, onde o peso é o Valor *CVSS* e a percentagem de precisão de cada vulnerabilidade, ambos já apresentados.
O *NSIP* é calculado segundo a seguinte fórmula e corresponde a um valor entre zero e dez com uma casa decimal.

[10] Endereços onde foi possível efectuar pelo menos um teste.
[11] Uma ocorrência de uma determinada vulnerabilidade significa que o teste efectuado devolveu um resultado positivo.

NSIP, Nível de Segurança da Internet Portuguesa	
NSIP =	$\frac{\sum_{k=1}^{n}\left(\#Teste_k \;.\; CVSS \;.\; \%Prec\right)}{\#IPsTestados} = \mathbf{2.1}$
Legenda: **%Prec** - grau de precisão. **n** - número testes existentes no sistema. **CVSS** - valor CVSS de um determinado teste. #**IPs Testados** - número total de endereços IP onde foi possível executar pelo menos um teste.	

Nº Vuln.	**Descrição**	**Número de Ocorrências/Presenças**	**Precisão**
1	SNMP, leitura	697	100%
2	SNMP, escrita	690	100%
3	DNS Zones, AXFR	1256	100%
4	DNS Snooping	1962	90%
5	SSH 1.1	3442	100%
6	SSH, Debian	241	100%
7	Telnet	15782	100%
8	SMTP Open Relay	21	90%
9	Finger	431	80%
10	CIFS/SMB shares	672	100%
11	CIFS/SMB shares (C:)	988	100%
12	SSL Expired	966	100%
13	SSLv2	3765	100%
14	NetSky	94	50%
15	MyTob	329	65%
16	Zafi	239	50%
17	MyDoom	60	50%

Tabela 2: Número de ocorrências de cada Vulnerabilidade Técnica.

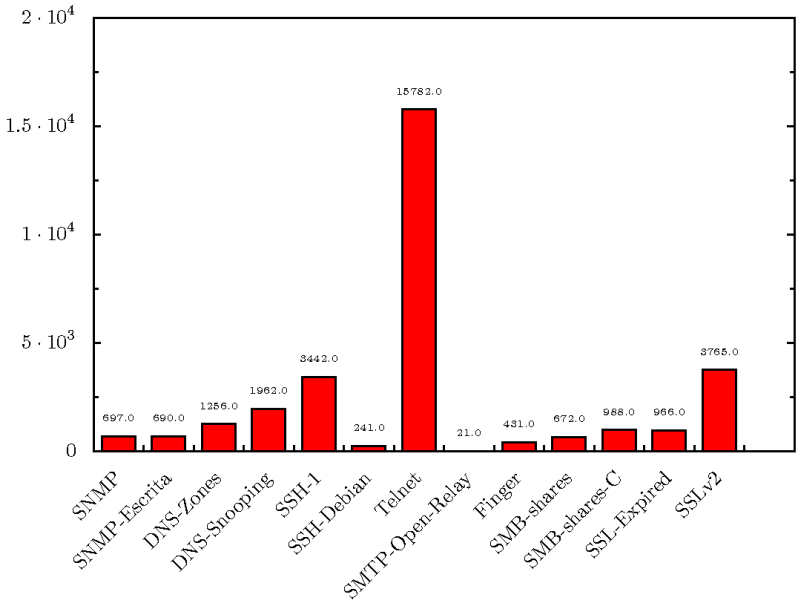

Figura 4: Ocorrências de cada Vulnerabilidade Técnica.

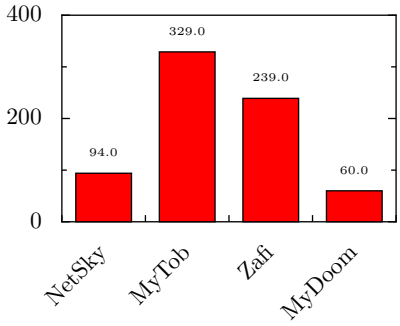

Figura 5: Presenças de *Malware* detectadas.

Resumindo, o *NSIP* corresponde ao racio entre o somatório de, o número de ocorrências de cada teste efectuado multiplicado pelo seu valor *CVSS* e pelo seu grau de precisão, e o número total de endereços *IP* onde foi possível efectuar pelo menos um teste.
Para facilitar a compreensão do significado do *NSIP* e consequentemente aumentar a abrangência da disseminação do **Nonius**, foi elaborada uma escala qualificativa que não é apresentada neste artigo, mas encontra-se totalmente descrita no *web-site* no projecto Nonius[12].
Uma das análise feitas aos resultados obtidos considerou a subdivisão entre um sector Estatal (infra-estruturas Estatais e Governamentais), e um sector Privado.
Contudo, esta subdivisão, não pode ser tida como exacta, uma vez que não é totalmente completa e pode conter falsas inclusões. O processo que suporta esta subdivisão tem um cariz não-automatizado, o que torna bastante difícil alcançar um nível de precisão completa dada a dimensão da população a ser testada, nomeadamente da lista de domínios *.pt*.

NSIP =	$\frac{\sum_{k=1}^{n}\left(\#Teste_k \ . \ CVSS \ . \ \%Prec\right)}{\#IPsTestados}$
Estatal	**NSIP = 1.6**
Privado	**NSIP = 2.2**

A tabela 3 e a figura 6, descrevem a variação dos dados globais apresentados anteriormente, pelas parcelas de endereçamento testado referentes ao Estado e ao sector privado.

A tabela 4 apresenta as percentagens de endereços com uma ou mais vulnerabilidades (endereços vulneráveis), no quadro geral, na parcela do endereçamento testado associado ao Estado, Organizações Estatais e Governamentais, e na parcela associada ao sector privado.

6 Trabalho Futuro

O futuro do **Nonius** passará obrigatoriamente por uma expansão do seu âmbito de execução, nomeadamente pelo aperfeiçoamento da detecção e diferenciação das parcelas de endereçamento referentes ao sector Privado e sector Estatal, e uma tentativa de aumento do número de domínios *.pt* a serem testados.
Para além disso será garantido que o resto do caminho a percorrer, afim de concretizar os objectivos inicialmente traçados, será concluído. Em concreto, uma maior difusão possível dos resultados obtidos e respectivos conteúdos de consciencialização.
Por último, é importante referir a ampliação da carga de testes e aperfeiçoamento da mesma. O esforço de aperfeiçoamento dos testes já existentes incidirá especialmente nos testes relativos à presença de *Malware*, uma vez que são os que apresentam menor taxa de preci-

[12]https://www.cert.ipn.pt/Nonius/tech.html

Nº Vuln.	Descrição	Nº de Ocor. (Estado)	Nº de Ocor. (Privado)	Precisão
1	SNMP, leitura	0	697	100%
2	SNMP, escrita	0	690	100%
3	DNS Zones, AXFR	415	841	100%
4	DNS Snooping	585	1377	90%
5	SSH 1.1	280	3162	100%
6	SSH, Debian	18	223	100%
7	Telnet	1453	14345	100%
8	SMTP Open Relay	9	12	90%
9	Finger	104	327	80%
10	CIFS/SMB shares	4	668	100%
11	CIFS/SMB shares (C:)	12	976	100%
12	SSL Expired	77	889	100%
13	SSLv2	418	3347	100%

Tabela 3: Ocorrências de cada Vulnerabilidade Técnica no sector Estatal e no sector Privado.

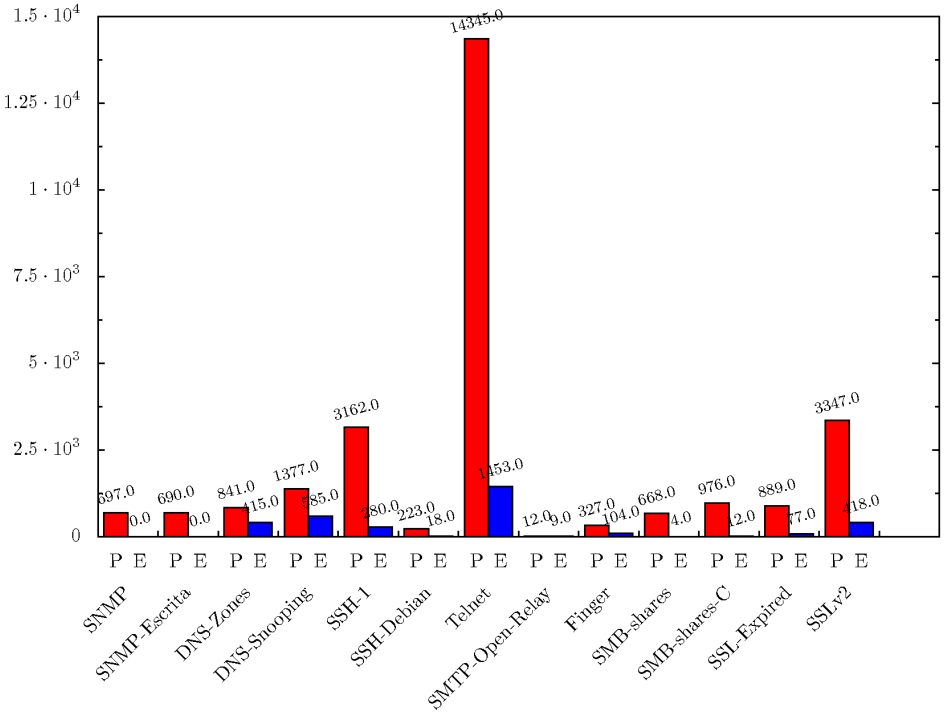

Figura 6: Variação das V.T. no sector Estatal e Privado.

são. Relativamente à implementação de novos testes, surgiram naturalmente algumas ideias ainda por concretizar. Como referência aponta-se a possibilidade de teste de *DNS Injection*, de *BGP Hijacking* ou de um conjunto de teste direccionados para as técnicas conhecidas como *Google Hacking*.

	Global	Estatal	Privado
Endereços vulneráveis	23 351	2 100	21 251
Endereços testados	83 306	11 380	71 926
Endereços vivos	516 213	16 607	499 606
Em relação aos endereços vivos	4.5%	12.6%	4.3%
Em relação aos endereços testados	28.0%	18.5%	29.5%

Tabela 4: Percentagem de endereços vulneráveis.

7 Conclusões

Evidentemente que, o trabalho aqui descrito representa apenas uma perspectiva do que poderá ser uma análise ao nível de segurança de um determinado segmento da Internet, até porque não se tratam de forma alguma de resultados exactos.
Contudo é importante reafirmar o carácter de alguma forma inovador do sistema, uma vez que não existem, pelo menos do conhecimento dos autores, sistemas que recolham o mesmo tipo de resultados de uma forma tão activa e focalizada, num âmbito de acção tão abrangente e definido. Poderão eventualmente ser comparados com o Nonius sistemas como, *Whats that site running?* e *Secure Server Survey* da Netcraft, o *Internet Threat Level*, ISS (IBM), ou *Threat Center* da Mcafee. Contudo, todos estes e muitos outros sistemas, ou são baseados em simples inquéritos, ou não abrangem um detalhe técnico tão aprofundado e um âmbito tão alargado como o Nonius, ou são baseados em recolhas passivas, ou simplesmente não revelam o processo que usam, o que demostra claramente que os resultados obtidos não poderão ser equiparados ao **Nonius**.
Paralelamente à concepção e implementação do **Nonius**, foi feito um estudo jurídico relativamente à viabilidade legal do Nonius. Estudo este que, de uma foram sucinta, diz que todos os processos e técnicas usadas pelo Nonius são completamente legais segundo a Lei vigente. Devendo-se isto, essencialmente ao facto de o Nonius se "limitar"a utilizar informação tida como pública, ou seja, não é feita nenhuma intrusão ou adulteração de sistemas de protecção e afins, para obtenção da informação utilizada.
Um outro aspecto que é interessante referir, é o facto de o sistema ter executado a primeira iteração em cerca de 14 dias, dado que teve um âmbito de acção consideravelmente alargado (cerca de 3.6 milhões de endereços IPv4 e 11 mil domínios *.pt*).
Relativamente aos resultados em si não haverá muito mais a acrescentar, são consideravelmente satisfatórios relativamente à sua expressividade e abrangência. Poderão não ser tão satisfatórios noutra perspectiva, na medida que apresentam valores de alguma forma preocupantes para a comunidade da internet Portuguesa. É importante referir, contudo, que os resultados poderiam ser bastante mais alarmantes, o que poderá vir a acontecer como consequência do alargamento do espectro da carga de testes. Todos os resultados apresentados neste artigo estão disponibilizados no *web-site* no projecto Nonius[13].

Referências

[1] *Debian Security Advisory: DSA-1571-1 openssl – predictable random number generator.* http://www.debian.org/security/2008/dsa-1571

[2] David Wagner - University of California- Berkeley, daw@cs.berkeley.edu; Bruce Schneier, Berkeley Counterpane Systems, schneier@counterpane.com *Analysis of the SSL 3.0 protocol.* http://www.schneier.com/paper-ssl.pdf .

[13] https://www.cert.ipn.pt/Nonius/

[3] *Serviços de Disseminação do CERT-IPN*. http://www.cert.ipn.pt/pt/disseminacao.html

[4] *RIPE Network Coordination Centre*. http://www.ripe.net/

[5] *Common Vulnerability Scoring System (CVSS-SIG)*. http://www.first.org/cvss/

Segurança em Redes de Acesso *Triple-Play*

T. Cruz[1], T. Leite[1], P. Baptista[1], R. Vilão[1], P. Simões[1], F. Bastos[2], E. Monteiro[1]

[1] CISUC - DEI, Universidade de Coimbra
[2] PT Inovação - Aveiro

tjcruz@dei.uc.pt

Resumo

O S3P é um projecto de investigação levado a cabo pelo grupo de Comunicações e Telemática do Centro de Informática e Sistemas da Universidade de Coimbra e pela PT Inovação. Este projecto tem por objectivo a identificação de novos riscos de segurança, introduzidos pela crescente disseminação de redes domésticas ligadas à Internet por banda larga (ADSL, cabo, fibra, 3G), e a investigação de soluções para neutralizar esses riscos. Apresenta-se aqui a arquitectura de gestão distribuída para ambientes "triple-play" que foi desenvolvida no âmbito deste projecto. Esta arquitectura, especificamente orientada para as questões da segurança nestes ambientes, caracteriza-se pelo seu carácter fortemente distribuído (melhorando assim a escalabilidade do sistema) e pela forma como integra nas soluções de segurança do operador dispositivos presentes nas redes dos clientes e na fronteira entre as redes dos clientes e a rede de acesso do operador.

1. Introdução

As redes de acesso de banda larga, na sua forma actual, representam um risco de segurança significativo para o *Internet Service Provider* (ISP), pela associação entre quatro factores: os elevados débitos disponíveis para cada um dos clientes; a massificação generalizada deste tipo de acesso, resultando num número de clientes servidos por cada ISP; o carácter tendencialmente permanente das ligações (xDSL, Cabo); e o facto de grande parte destes clientes não terem conhecimentos técnicos suficientes para garantir a segurança da sua rede doméstica. Ainda que parte dos riscos actuais existisse já anteriormente, o carácter intermitente das ligações *dial-up* clássicas e os reduzidos débitos disponíveis tornavam mais simples aos ISP a tarefa de detectar e controlar situações de risco para as suas redes, para os seus clientes ou para terceiros.

A recente convergência dos serviços de voz, dados e televisão num mesmo canal de acesso (*triple play*), aliada à profusão de serviços e aplicações (P2P, *instant messaging*, etc.) veio agravar ainda mais o problema da segurança em ambientes de banda larga, com repercussões a vários níveis. Em primeiro lugar, os clientes, por falta de sensibilidade tecnológica, têm uma crescente dificuldade para lidar com o problema de segurança ao nível das suas próprias redes domésticas, ficando estas mais vulneráveis a ataques externos que poderão posteriormente comprometer a segurança da própria rede do operador. Adicionalmente, a sucessiva adição de serviços e aplicações torna cada vez mais difícil a detecção e resolução de ataques de segurança, principalmente quando esta tarefa é confiada a sistemas centralizados na rede do operador, de limitada escalabilidade. Por último, o impacto de quebras ou limitações de serviço é cada vez maior, pois os clientes esperam que os serviços agora suportados sobre o canal de banda mantenham parâmetros de qualidade e fiabilidade não inferiores às experiências anteriores com meios convencionais de acesso a telefone e televisão.

A atitude tradicional dos ISP tem sido considerar que a segurança da rede do cliente está fora da sua esfera de influência, devendo ser administrada autonomamente pelo cliente. Em geral, os operadores consideram que a sua esfera de influência pára no seu equipamento de fronteira (e.g. DSLAMs), sendo responsabilidade do cliente a gestão dos seus equipamentos de fronteira – *home gateways* – e de tudo o que esteja para lá desses equipamentos. Esta atitude está aliás alinhada com a perspectiva cultural da maioria dos utilizadores, que veria com maus olhos a interferência do operador – implícita ou explícita – na sua rede doméstica.

As redes "*triple play*" começam a alterar parcialmente esta perspectiva, passando a ser necessárias e aceites algumas intervenções do operador no interior da rede do cliente, nomeadamente para administrar remotamente *set-top-boxes* (STB) e *gateways* de serviço telefónico. Mesmo para além desse contexto específico a profusão de novos dispositivos nas redes domésticas, a oferta de novos serviços (IPTV, VoD, telefone, televigilância, *online backup*…) e a mudança dos modelos de tráfego (com um peso cada vez maior de tráfego P2P) tornam necessário premente repensar estes pressupostos.

Mesmo sem colocar em causa a autonomia e privacidade dos utilizadores domésticos, existe actualmente uma janela de oportunidade para questionar o actual modelo de segurança, tentando articular melhor os mecanismos de segurança ao nível do ISP com os mecanismos de segurança disponíveis em cada rede doméstica. No modelo actual os operadores tentam controlar o tráfego montagem "barragens" num conjunto relativamente reduzido de pontos da sua própria rede. Essas "barragens" necessitam por conseguinte de lidar com volumes de tráfego substancialmente elevados, com as consequentes implicações ao nível de custos, escalabilidade e granularidade.

Em alternativa a esse modelo, propõe-se o aproveitamento do posicionamento específico que as *gateways* domésticas possuem no contexto das infra-estruturas de banda larga – como mecanismos que fazem a mediação entre as fronteiras da rede do operador e dos clientes – para implementar um IDS/IPS (*Intrusion Detection System/Intrusion Protection System*) largamente distribuído. Caso o operador possa usar essas *gateways* domésticas como primeiro ponto de defesa da sua própria rede, poderá implementar mecanismos de segurança mais sofisticados, mais escaláveis e mais granulares. Em paralelo, os próprios utilizadores sem conhecimentos técnicos beneficiarão com esta gestão partilha das suas *gateways*, passando a ter redes domésticas mais protegidas.

O Projecto S3P – um trabalho de investigação levado a cabo pelo grupo de Comunicações e Telemática do Centro de Informática e Sistemas da Universidade de Coimbra e pela PT Inovação – tem por principais objectivos a definição, implementação e avaliação de uma arquitectura de segurança baseada nesse pressuposto de melhor articulação entre redes domésticas e rede do operador, passando a encarar a *gateway* doméstica como um dispositivo útil ao ISP e ao cliente. Nesta comunicação são apresentados os principais aspectos da arquitectura do S3P, de acordo com a seguinte organização: a Secção 2 discute em maior detalhe o contexto do projecto (ambientes *Triple Play* e tendências da indústria), a Secção 3 apresenta os traços gerais da solução proposta, e as Secções 4 e 5 apresentam a arquitectura da plataforma S3P (na perspectiva da *gateway* doméstica e do operador, respectivamente). A Secção 6 apresenta as conclusões e discute trabalho futuro.

2. Motivação

Tal como foi já mencionado na Secção anterior, nos ambientes de banda larga a segurança da rede doméstica do cliente é, tradicionalmente, da sua inteira responsabilidade. Ainda que isto seja aceitável para clientes tecnicamente qualificados, coloca riscos consideráveis no caso da esmagadora maioria dos clientes domésticos, cuja capacidade para instalar e gerir mecanismos de segurança é nula ou bastante reduzida. Esta situação afecta em primeiro lugar o próprio cliente mas acarreta também consequências para o operador. Por um lado tem um cliente potencialmente menos satisfeito (degradação de serviços prestados a esse cliente, incidentes graves de segurança na esfera do cliente). Por outro lado, caso a rede do cliente seja comprometida poderá ser usada para atacar outros clientes do ISP, o próprio operador ou terceiros. Nesse cenário, os elevados débitos oferecidos pelas actuais redes de acesso, a profusão de aplicações P2P e a convergência para cenários totalmente suportados sobre IP aumenta substancialmente os riscos, tanto para o cliente (numa perspectiva de intrusão na sua rede e acesso a dados confidenciais) como para o operador, que passa a estar muito mais vulnerável a ataques concertados de DoS e a situações de uso abusivo da sua infra-estrutura de rede.

A visão tradicional dos serviços de acesso de banda larga – nos quais o fornecedor apenas oferece serviços de conectividade, dando completa autonomia ao utilizador na forma de organizar a sua rede doméstica – perde algum sentido com *Triple Play* e outros serviços de valor acrescentado que dependem directamente de equipamento a colocar em casa do cliente. Na maior parte dos casos estes serviços exigem a instalação de equipamentos fornecidos especificamente pelo operador (*set-top-boxes*, telefones IP, centrais de alarme…), quer por questões de compatibilidade técnica quer por estratégias comerciais (por exemplo capacidade de garantir níveis de DRM apropriados para conteúdos multimédia). A crescente aceitação desses dispositivos abre caminho para uma redefinição da fronteira entre cliente e operador que permita uma melhor articulação entre a rede de acesso e a rede doméstica, sem com isso deixar de garantir a autonomia, liberdade de escolha e privacidade do cliente.

Esta tendência tem-se reflectido na indústria, com a entrada na rede doméstica de equipamentos do operador (em especial set-topboxes) e com a presente tendência de normalização e convergência. Seja por iniciativas como a HGI [1] e o Broadband Forum (ex-DSL Forum) [2] seja por produtos como o *Windows Home Server* [3], passará a ser possível contar na rede de cada cliente com um conjunto homogéneo de serviços de segurança e administração remota, capazes de monitorizar a rede interna do cliente (se este assim o desejar) e a ligação entre a rede doméstica e a rede do operador. Diversas propostas técnicas produzidos pelo *Broadband Forum* e pela HGI apontam neste sentido, com a proposta de *interfaces* normalizados para configurações de serviços de segurança e operações de manutenção remota [4-6] em dispositivos localizados na rede do cliente (CPE, *Customer Premise Equipment*).

Em conjunto, estas tendências abrem caminho para repensar a segurança das redes domésticas e a segurança das redes de acesso. Por um lado, é necessário identificar e caracterizar as novas ameaças de segurança associadas a este cenário. Por outro lado, é necessário investigar e avaliar novas abordagens à forma de lidar com a rede doméstica do cliente.

3. Abordagem Proposta

3.1 Modelo de gestão e integração na infra-estrutura do operador

Como resposta à crescente dificuldade em escalar soluções de segurança clássicas do lado do operador, o Projecto S3P propõe um modelo de segurança distribuído, aproveitando as capacidades de processamento e gestão remota das *home gateways*. Transferem-se assim parte das funções de segurança para o equipamento do cliente. As *home gateways* são actualmente dispositivos com capacidade computacional bastante razoável (processadores entre 200 e 400 MHz, memória RAM adequada, versões reduzidas de Linux), já estão disponíveis sem custos adicionais e estão num ponto privilegiado da rede (mediação da rede de um único cliente com a rede de acesso do operador). Podem assim ser usadas para filtrar o tráfego de rede (em ambos os sentidos), enviar informação relevante para o operador e/ou para o cliente (alarmes de segurança, padrões de utilização, etc.) e implementar medidas de protecção (por exemplo bloqueio selectivo de tráfego em resposta a eventuais ataques). Adicionalmente, nos casos em que sejam usadas exclusivamente para monitorizar o tráfego entre o ISP e o cliente, não reduzem a privacidade do cliente: o ISP poderia sempre proceder a uma monitorização semelhante dentro da sua rede, ainda que com custos substancialmente mais elevados.

O Projecto S3P propõe assim a criação de uma estrutura descentralizada em que as *gateways* domésticas actuam na linha da frente da protecção das redes internas, de modo a conter os efeitos de um eventual ataque a uma rede doméstica ou à rede do operador, ou mesmo evitá-lo de todo. A Figura 1 ilustra esta abordagem.

Essas *gateways* (designadas por CPE no âmbito do S3P, ainda que habitualmente o termo CPE tenha um âmbito mais alargado) passam a funcionar de forma coordenada. Para além de realizarem funções de monitorização e prevenção de ataques através de meios próprios (com base em configurações previamente definidas pelo operador), podem também notificar o IDS do operador de determinados eventos e exercer acções de controlo de tráfego com base em instruções do IDS central.

Continuarão obviamente a existir clientes cujas *gateways* não colaborem com o IDS do ISP e que existe o risco de ter *gateways* comprometidas dentro da estrutura, pelo que o grau de confiança depositado pelo ISP em cada *gateway* doméstica nunca pode ser absoluto. A plataforma do operador terá pois de ter flexibilidade suficiente para suportar simultaneamente clientes com *gateways* cooperantes, clientes com *gateways* comprometidas e clientes sem *gateways* integradas. Apesar disso, do ponto de vista global, os potenciais ganhos de granularidade e de escala são consideráveis.

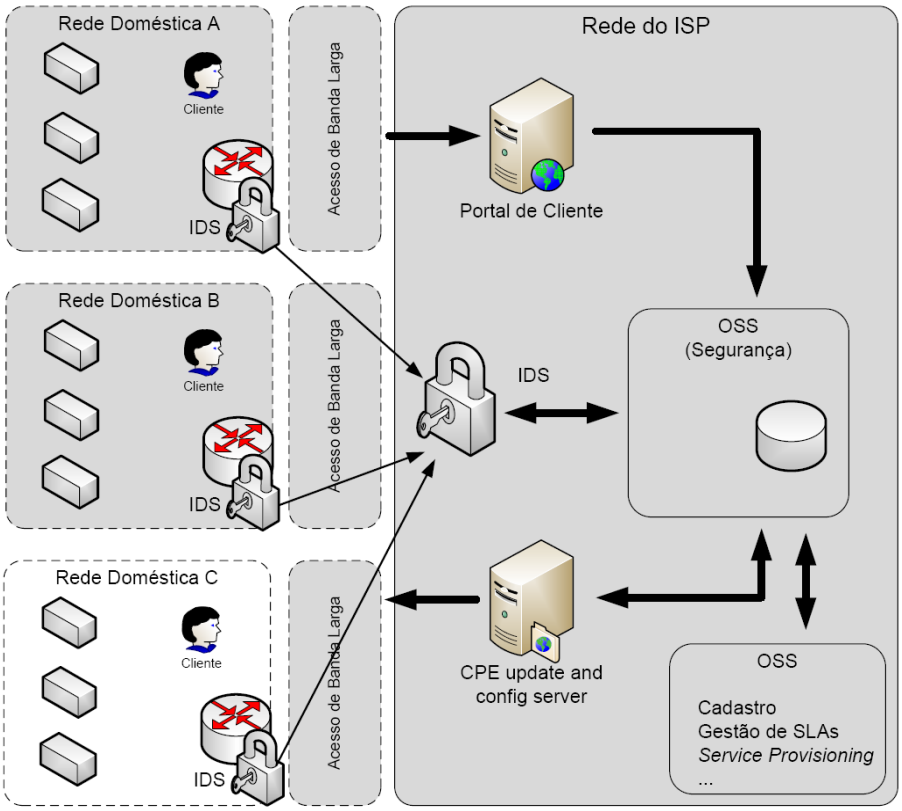

Figura 1. Modelo Genérico da Solução Proposta.

A arquitectura proposta não corresponde apenas ao "aproveitamento" da *gateway* doméstica pelo IDS do operador, tentando também aumentar efectivamente a articulação entre a rede do cliente e o ISP. Para o efeito as políticas de segurança adoptadas pelo IDS distribuído tomam em consideração o perfil do utilizador (cadastro, serviços contratados, etc.) e também permitem ao utilizador algum grau de personalização, por meio de um portal de cliente onde este pode por exemplo solicitar suporte explícito para algumas aplicações ou especificar perfis de uso mais detalhados que possam ser repercutidos no funcionamento do sistema (por exemplo controlo parental de conteúdos Web, bloqueio de acesso a servidores SMTP não previamente discriminados, etc.). Este cruzamento de informação é útil para o ISP e para o próprio cliente.

No seu essencial, a solução proposta pelo projecto S3P vai de encontro à noção de IDS distribuído. Este modelo conceptual é suportado por várias estações colectoras de dados e uma ou várias estações centrais que realizam a correlação dos dados obtidos. Independentemente de aspectos como as topologias adoptadas [7] [8] ou mesmo a disposição das estações colectoras [9] a ideia base tem-se vindo a manter relativamente inalterada desde a sua concepção. O projecto S3P procura integrar esta noção num contexto mais específico, com recurso às tecnologias existentes e especificamente desenvolvidas para os ambientes de banda larga.

A arquitectura prevê o uso de entidades (agentes) presentes ao nível do ISP e CPE (Figura 2). Estas entidades actuarão sobre a análise do tráfego de dados transmitido, além de outros dados que possam ser obtidos a partir das bases de dados do ISP. Para que haja coordenação entre os CPE dos diversos clientes, será adicionalmente

necessário o suporte por aplicações centrais ao nível do operador. Estas aplicações fornecerão actualizações e outras rotinas de verificação, monitorização e configuração.

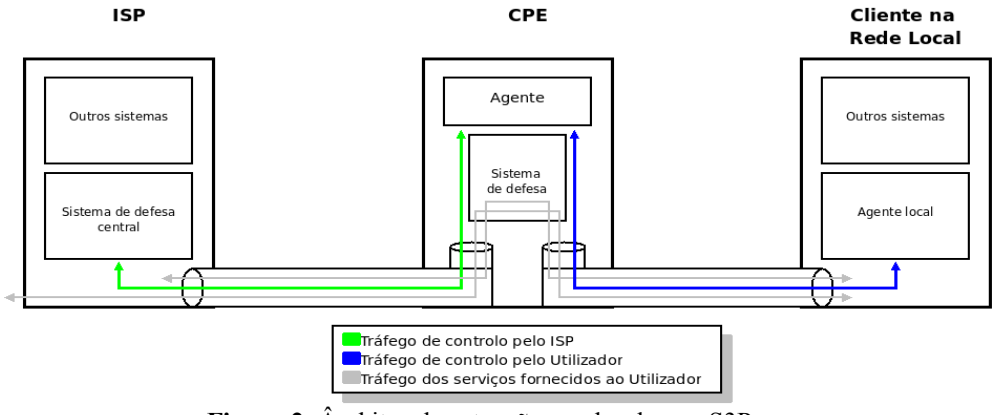

Figura 2. Âmbitos de actuação na abordagem S3P.

Conforme transparece da Figura 1, o modelo proposto não é totalmente descentralizado, visto continuar a existir uma infra-estrutura de gestão do lado do operador que coordena os vários intervenientes neste processo, orquestrando a sua operação com base na correlação das informações recolhidas na rede do próprio operador e nos diversos CPEs.

Para a comunicação entre as aplicações do operador e os CPEs optou-se pela norma TR-069 ou CWMP (*CPE WAN Management Protocol* [5]), desenvolvida pelo *Broadband Forum* para a gestão remota de equipamentos da rede doméstica. O TR-069 pertence ao âmbito das *Broadband Suites* do referido *forum*, fazendo assim parte de uma família mais de normas e protocolos extensíveis e orientados para a gestão em ambientes de banda larga. Esta norma tem vindo a conhecer crescente aceitação, sendo de esperar que seja gradualmente integrada por todas as aplicações de administração, do lado dos operadores, e por todos os equipamentos, pelo lado dos fabricantes de CPEs (em especial routers/modems ADSL e *set top boxes*).

A adopção do TR-069, complementado pelo já referido modelo de gestão de perfis de utilizadores, permite que seja relativamente simples ao operador disseminar novas regras ou configurações para grupos alargados de utilizadores, em função dos seus perfis específicos e dos equipamentos instalados.

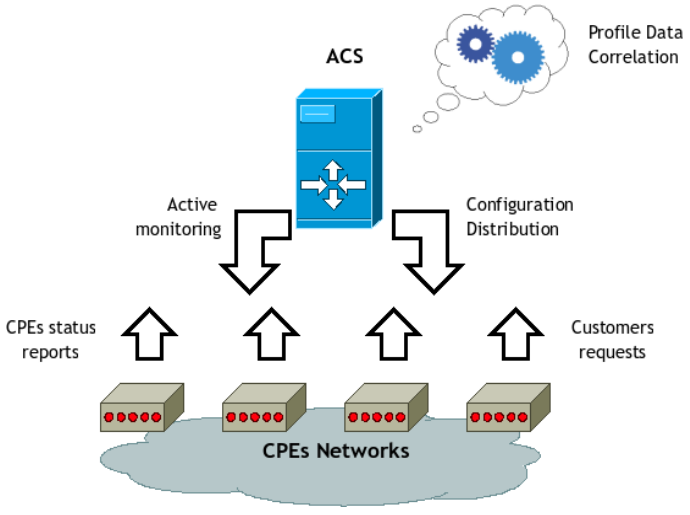

Figura 3. Operação com recurso ao protocolo TR-069.

A gestão de configurações é realizada recorrendo ao servidor de auto-configuração (ACS, ou *Auto-configuration Server*, segundo a terminologia TR-069), que realiza a distribuição de actualizações de software

dos CPEs, a adição de novos serviços e a gestão dos perfis do ambiente. A maior parte das transacções realizadas entre o ACS e os CPEs são já normalizadas pelo *Broadband Forum*, tendo as restantes transacções sido implementadas como extensões da norma TR-069, de acordo com o modelo de extensões *vendor-specific* previstas na norma. A Figura 3 ilustra o relacionamento entre os CPE e o ACS.

3.2 Mecanismos de segurança e tratamento de eventos na arquitectura S3P

A ideia de proporcionar um papel mais activo às *gateways* domésticas não é propriamente novidade. Ferramentas como *firewalls* (inicialmente do tipo *stateless* e, mais recentemente, *stateful*) e mecanismos de gestão de *QoS* fazem hoje parte da maior parte desses equipamentos, podendo ser configurados pelos utilizadores por meio de interfaces *Web*. No entanto, esta abordagem à questão é limitada por um conjunto de factores:

- A responsabilidade de configurar estas ferramentas é do utilizador, que frequentemente não tem a preparação técnica adequada para o efeito.
- A *gateway* funciona de forma isolada, não havendo por exemplo correlação de ataques com outros utilizadores do mesmo ISP ou com serviços específicos do operador ou da rede local. Um ataque realizado de modo distribuído, como é característico das *botnets,* não é detectável através da análise de uma rede isolada.
- A capacidade destas ferramentas é relativamente limitada, podendo não ser suficiente para os ataques cada vez sofisticados a que hoje se assiste. Uma ferramenta, por mais flexível e poderosa que seja, não é efectiva se não for acompanhada por um conjunto de regras e mecanismos adaptáveis à altura nas necessidades.

Estes três factores são minimizados, no projecto S3P, de vários modos:

- Pela maior sofisticação dos mecanismos locais de segurança, incluindo filtros de pacotes, *proxies*, detecção e prevenção de intrusos, detecção de *portscans* e vários outros mecanismos habitualmente reservados para redes de maior dimensão.
- Pela capacidade que o operador tem de correlacionar incidentes ocorridos em clientes distintos e activar mecanismos de resposta coordenados ao nível da sua rede e de todos os seus clientes.
- E pelo facto de as configurações de segurança de cada CPE serem geridas pelo operador, ainda que tendo em conta as preferências do cliente.

A detecção e o correcto tratamento de incidentes de segurança (que designaremos genericamente por *eventos*) é uma peça fundamental da arquitectura proposta. A geração e o tratamento de eventos ocorrem a dois níveis distintos:

- Ao nível local (CPE). Por razões de eficiência e escalabilidade, e de modo a permitir uma elevada granularidade no processo de monitorização, o CPE está dotado de um motor local de correlação de eventos (eventos esses captados pelas ferramentas de análise do tráfego, nomeadamente o sistema de detecção de intrusão e *portscans* e registos da actuação de outras ferramentas, tais como os bloqueios realizados pelo *firewall*, *proxy* e sistema de prevenção de intrusões). Todos os eventos são processados pelo motor local de correlação, podendo despoletar contra-medidas de natureza local, baseadas na aplicação de regras e procedimentos ao nível dos mecanismos de segurança do próprio CPE e/ou notificações de eventos para o ISP. Um potencial ataque que seja detectado através dos registos do IDS será enviado ao correlacionador de eventos que por sua vez deverá acionar a(s) medida(s) adequada(s) quando determinar que exista ameaça ao ambiente. Os eventos poderão ser enviados para o ISP se a gravidade da ocorrência assim o exigir e/ou ser utilizados localmente para geração de regras de modo automático.
- Ao nível do ISP. Os eventos recebidos dos diversos CPEs são correlacionados pelo motor de eventos do ISP, permitindo assim detectar, por exemplo ataques concertados a/de vários clientes do ISP. O

ISP pode reagir a esses eventos tomando medidas preventivas na sua própria rede ou alterando as configurações dos CPEs (Figura 4). Um exemplo deste modelo seria por exemplo a detecção de um ataque concertado por meio de uma *botnet*, com diversos clientes infectados, e a distribuição pelos CPEs de regras de bloqueio das portas usadas por essa *botnet*.

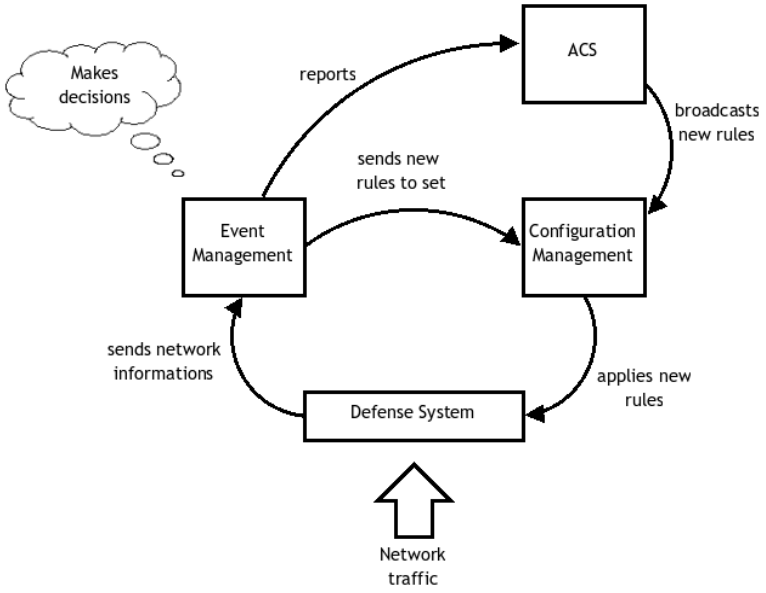

Figura 4. Modelo operacional de tomada de decisão na arquitectura S3P.

Em termos gerais a plataforma proposta funciona como um sistema de detecção e prevenção de intrusões distribuído, abrangendo a rede do operador e os pontos de entrada/saída da rede dos clientes. A título de exemplo, consideremos um conjunto de redes cliente que acabam de ser atacadas por um *trojan* e estão a cooperar num ataque DDoS (*Distributed Denial of Service*) sincronizado. Os CPEs poderão, ao detectar actividade anómala, alertar o operador através de um evento. Do lado do operador o mecanismo de correlação, ao detectar um padrão global, efectua a distribuição de novas regras de segurança de modo a prevenir este ataque nos restantes clientes. O mecanismo de correlação actua de acordo com os perfis associados ao ambiente em questão. Padrões de tráfego que não estejam de acordo com as premissas do perfil poderão ser considerados como eventos passíveis de activar contramedidas por parte deste mecanismo. É nesta óptica que se destaca a importância da gestão de eventos distribuída no projecto S3P.

Em termo de tecnologias, optou-se por uma solução capaz de suportar em simultâneo os dois níveis de funcionamento (CPE, ou microscópico, e operador, ou macroscópico) e com bons mecanismos de comunicação entre os dois níveis. Essa solução assenta na ferramenta *Prelude IDS* [10], ao nível de ambos os motores de correlação, e em IDMEF (*Intrusion Detection Message Exchange Format* [11]), ao nível da comunicação de eventos entre os CPEs e o operador. O IDMEF é um protocolo recentemente proposto pelo IETF para troca de informação relacionada com eventos de segurança, esperando-se que seja gradualmente aceite pela indústria como norma aberta para troca de informações de segurança. Esta solução será discutida em maior pormenor nas Secções 4 e 5.

4. Arquitectura da Plataforma S3P (CPE)

A Figura 5 apresenta a arquitectura da plataforma S3P na perspectiva do CPE. Os três módulos nucleares do CPE correspondem ao sistema de defesa (*Defense System*), ao motor de gestão de eventos (*Event Management*) e à gestão de configuração (*Configuration Management*). Entre os módulos de suporte inclui-se

o gestor de falhas (*Failure Management*), destinado a gerir o funcionamento do próprio CPE (avarias de hardware, perdas de configurações, etc.) e o monitor da rede do cliente (*Customer Network Monitoring*), um módulo que poderá mais tarde ser usado para monitorizar a rede doméstica do utilizador[1].

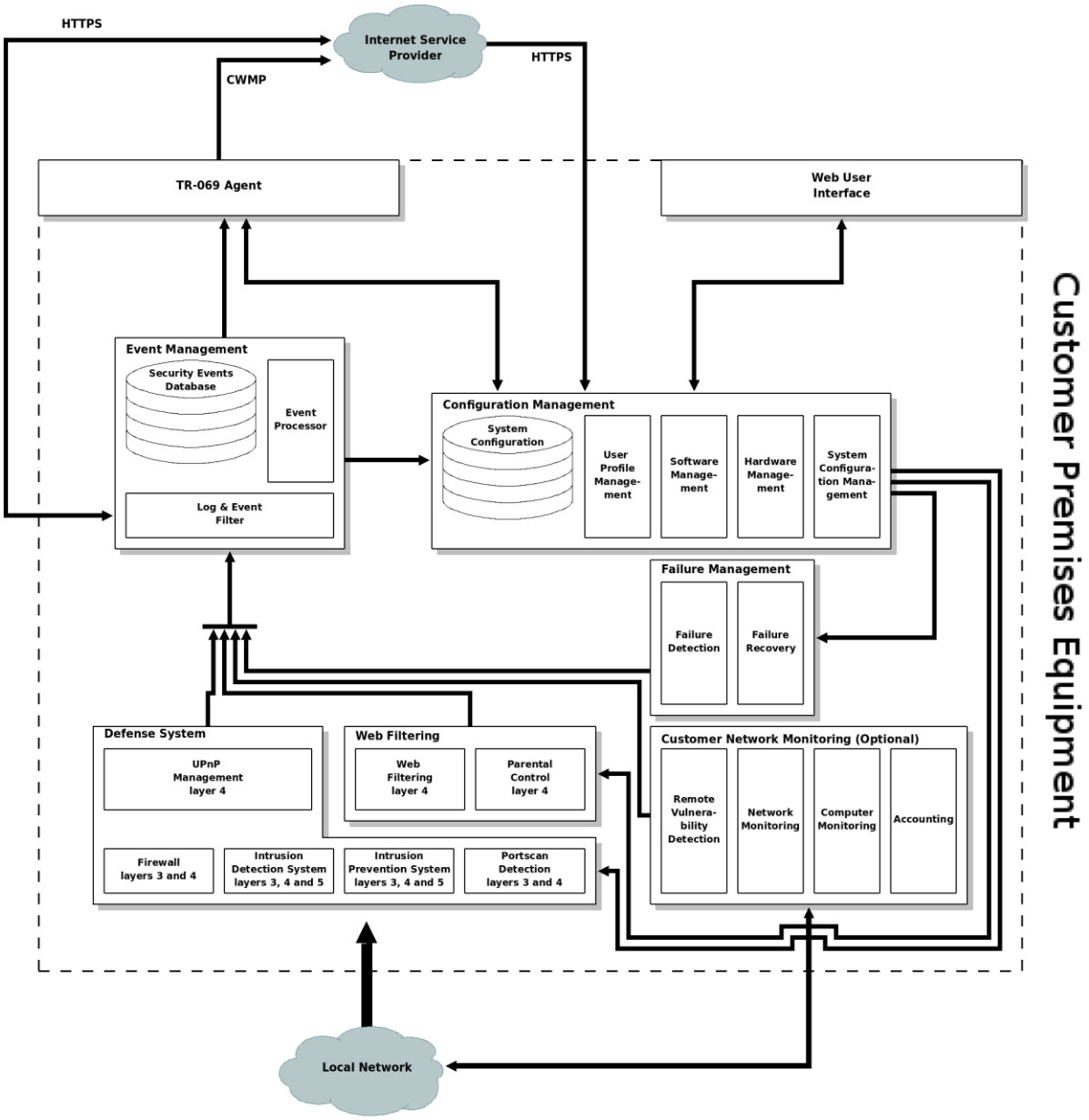

Figura 5. Arquitectura da Plataforma S3P (CPE)

[1] Este módulo não foi até agora objecto de trabalho no projecto S3P e poderá suscitar algumas questões éticas, já que permitirá ao operador monitorizar proactivamente a própria rede local do seu cliente, detectando e/ou colmatando falhas de segurança (por exemplo PCs locais com fragilidades de segurança). No entanto, do ponto de vista arquitectural é importante inclui-lo pelos serviços adicionais de segurança que oferece e que poderão interessar a determinados clientes.

Do ponto de vista de hardware, optou-se por uma plataforma de referência ligeiramente acima da capacidade dos *routers* de banda larga correntes. Em vez dessa capacidade corrente (tipicamente com CPUs entre 200 e 400 MHz e cerca de 256 Mbyte de RAM) optou-se por tomar como referência uma configuração que será previsivelmente atingida dentro de 2 a 3 anos: CPUs com velocidades na ordem dos 900 MHz, 512 Mbyte a 1 Gbyte de RAM, 2 a 4 Gbyte de capacidade de armazenamento não volátil. Espera-se que esta configuração se torne vulgar dentro de pouco tempo – em especial por via da disseminação da plataforma Atom, da Intel, que reduzirá significativamente os custos deste tipo de dispositivos – e com ela torna-se mais simples integrar um grande número de ferramentas *opensource* disponíveis para Linux.

4.1 *Defense System*

O *Defense System* agrega os mecanismos activos de protecção do ambiente e análise passiva do tráfego que circula pelo CPE. A sua configuração (regras, listas de acesso, etc.) é controlada pelo sistema de gestão de configurações que pode actuar em consonância com regras criadas pelo correlacionador de eventos ou através de comandos enviados pelo operador.. Para que isto seja possível, os componentes de software serão geridos através da distribuição de pacotes personalizados. O *Defense System* inclui os seguintes componentes: *firewall*, filtragem *Web* (*proxy* e controlo parental), sistema de detecção e prevenção de intrusão (IDS/IPS), detector de *portscans* e um gestor de dispositivos *UPnP* (*Universal Plug and Play* [12]).

Alguns destes componentes terão funções passivas (detectores de *portscans*, IDS e e gestor uPnP), limitando-se a gerar eventos para tratamento pelo motor local. Outros terão também funções activas, podendo a sua configuração ser alterada dinamicamente, por decisão local ou do operador, em reacção a incidentes de segurança.

Do ponto de vista de implementação do protótipo, todos estes componentes foram integrados a partir de ferramentas *opensource* já disponíveis (*GD UPnP*, *squid*, com *plug-in SquidGuard*, *scanlogd*, *Iptables*, *Snort* e *Snort inLine*), o que facilita a futura actualização da plataforma. A justificação para o uso das referidas ferramentas deve-se á sua popularidade, facilidade de manutenção, suporte e disseminação.

4.2 Gestão de Eventos

Tal como foi já mencionado, o gestor de eventos do CPE assenta no *Prelude IDS*. Nesta ferramenta os eventos são provenientes de sensores (agentes simples que analisam fontes de informação e a partir daí constroem mensagens IDMEF que posteriormente são enviadas para o módulo de gestão de eventos através de uma ligação segura) e mantidas numa base de dados local. Para além de existirem já sensores parametrizáveis para as ferramentas de segurança mais comuns (incluindo parte das ferramentas usadas pelo *CPE Defense System*) e para os serviços de rede mais habituais, é simples construir novos sensores, expandindo assim o sistema.

O processador de eventos inclui mecanismos sofisticados de correlação e de tratamento de eventos, usando para o efeito a linguagem LUA [13]. Esta é uma inovação que desataca o *Prelude IDS* dos outros HIDS existentes no mercado. Segue-se um exemplo para detecção de um ataque por força bruta a um sistema de autenticação interactivo, descrito na linguagem LUA:

```
function brute_force(INPUT)

local is_failed_auth = INPUT:match("alert.classification.text",
 "[Ll]ogin|[Aa]uthentication","alert.assessment.impact.completion",
 "failed")

local userid = INPUT:get("alert.target(*).user.user_id(*).name");

if is_failed_auth and userid then
 for i, user in ipairs(userid) do
    local ctx = Context.update("BRUTE_U_" .. user, { expire =
     120,threshold = 2 })
```

```
    ctx:set("alert.source(>>)", INPUT:getraw("alert.source"))
    ctx:set("alert.target(>>)", INPUT:getraw("alert.target"))
    ctx:set("alert.correlation_alert.alertident(>>).alertident",
    INPUT:getraw("alert.messageid"))
    ctx:set("alert.correlation_alert.alertident(-1).analyzerid",
    INPUT:getAnalyzerid())

    if ctx:CheckAndDecThreshold() then
     ctx:set("alert.classification.text", "Brute force attack")
     ctx:set("alert.correlation_alert.name", "Multiple failed login")
     ctx:alert()
     ctx:del()
    end
 end
end

end -- function brute_force(INPUT)
```

O motor de correlação é um mecanismo de natureza não-reactiva, dotado de memória. Os eventos, ao serem enviados para o motor de correlação, irão despoletar o processamento de todos os *scripts* registados no respectivo módulo. Assim, cabe ao programador fazer as verificações necessárias de modo a que sejam capturados os eventos correctos para correlação, sendo esta a primeira acção a ser efectuada - no exemplo acima mencionado existe uma verificação para confirmar se o texto capturado é proveniente de uma autenticação que não tenha sido bem-sucedida.

De seguida é extraído o utilizador ao qual o login foi negado. A partir daqui é iterado um ciclo para cada utilizador envolvido no evento (note-se que é possível ter vários utilizadores associados a um login falhado, pois poderão ser provenientes de uma anterior correlação), onde é verificado se este é reincidente recorrendo para o efeito à memória do correlador de eventos (Context.update). Caso não exista um contexto associado áquela chave, então irá ser criado uma nova instância com uma expiração de 120 minutos e um limite de duas ocorrências, i.e., para o um novo evento ser despoletado (ctx:alert()), terão de ocorrer duas falhas no espaço de dois minutos.

Estes mecanismos são bastante flexíveis, permitindo definir os vários tipos de comportamento previstos para a plataforma S3P: descarte de eventos; registo local de eventos, para histórico (por exemplo correlação com eventos futuros); tomada de decisões locais, com execução de medidas autónomas de resposta a incidentes de segurança; envio para o ISP de eventos locais (sejam eles os eventos originais ou eventos agregados), usando IDMEF, para que possam ser correlacionados com eventos de outros clientes e dar origem a respostas concertadas ao nível do operador.

Neste contexto, a correlação de eventos é importante na medida em que permite reduzir o número de alertas recebidos ao nível do operador, visto existirem eventos cujo processamento e tratamento será feito localmente a nível do CPE. Além disso, é necessário ter em conta que nem todos os eventos deverão gerar alertas, pois nem todos apresentam a mesma severidade. A título de exemplo: caso exista uma tentativa de autenticação no CPE com um conjunto de credenciais inválidas, será gerado um evento mas à partida este evento *per se* não deverá ser considerado um ataque. No entanto, se a mesma situação se repetir várias vezes no espaço de um minuto, podemos considerar que se trata de um ataque por força bruta. Caberá então ao mecanismo de correlação fazer a distinção entre as duas situações e, caso se verifique um ataque, gerar um alarme para o sistema tomar alguma medida de segurança que poderá passar, por exemplo, pelo bloqueio do endereço IP de onde provém o ataque na *firewall* local (Figura 6).

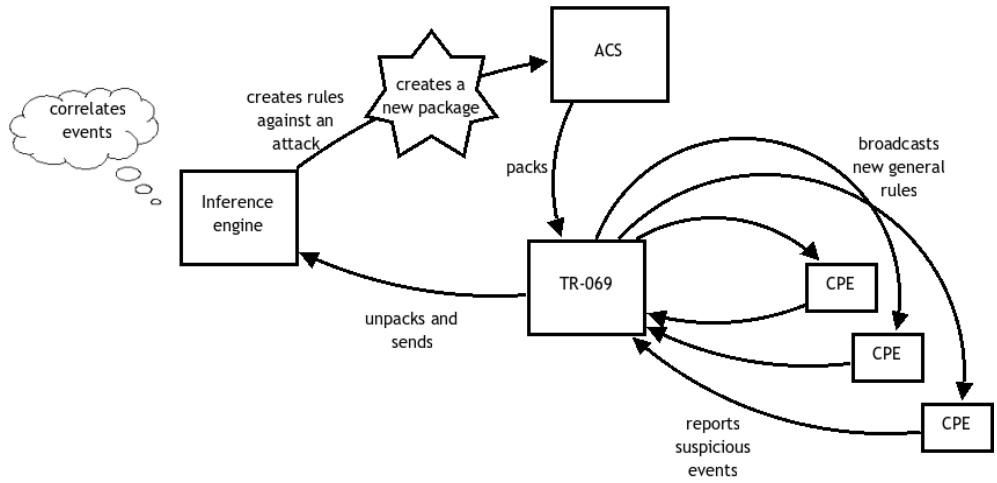

Figura 6. O mecanismo de correlação de eventos na perspectiva macroscópica

4.3 Gestão de Configurações

O Gestor de Configurações assegura a gestão das configurações do CPE (serviços instalados, configurações activas). As actualizações podem ser despoletadas remotamente pelo ISP (envio de um novo serviço ou de versão actualizada de um serviço já existente; envio de novas configurações para *firewall*, etc.) ou pelo próprio CPE (recuperação de configurações em caso de falha do *file system*, alteração de configuração decidida localmente para reacção a incidente de segurança…). Através do gestor de configurações é garantido que todas as alterações, sejam elas ao nível dos serviços ou regras de segurança, sejam aplicadas nos CPEs com o uso de pacotes que podem ser enviados pelo ISP, como actualização, ou gerados localmente.

5. Arquitectura da Plataforma S3P (na óptica do Operador)

A Figura 7 apresenta a arquitectura da plataforma, do lado da infra-estrutura do operador. Os principais componentes correspondem ao gestor de perfis (*Profile Management*), ao gestor de eventos de segurança (*Security Event Management*) e ao gestor dos CPEs (*CPE Management*). Entre os módulos complementares, incluem-se o Gestor de Falhas (*Failure Management*) e o monitor da rede do cliente (*Customer Network Monitoring*), assim como os componentes de integração com os sistemas OSS (*Operations and Support Systems*) e AAA (*Authentication, Authorization, and Accounting*) do operador.

5.1 Gestão de Eventos de Segurança

O gestor de eventos de segurança corresponde a um concentrador de eventos de segurança – alimentado pelos gestores de eventos de cada CPE e também por eventos detectados por sensores instalados na própria rede do operador – que permite correlacionar acontecimentos ocorridos em pontos distintos da rede e accionar respostas orquestradas a esses acontecimentos – por exemplo bloqueio de tráfego ao nível da rede do operador e/ou reconfigurações de *firewalls* dos CPEs.

Ao nível de tratamento de eventos é mais uma vez usado o *Prelude IDS*, num formato semelhante ao já descrito para os CPE. Relativamente às ferramentas de IDS e IPS que actuam na rede do operador, a plataforma S3P é neutra, podendo à partida ser integrada com as ferramentas que os operadores tenham já em exploração.

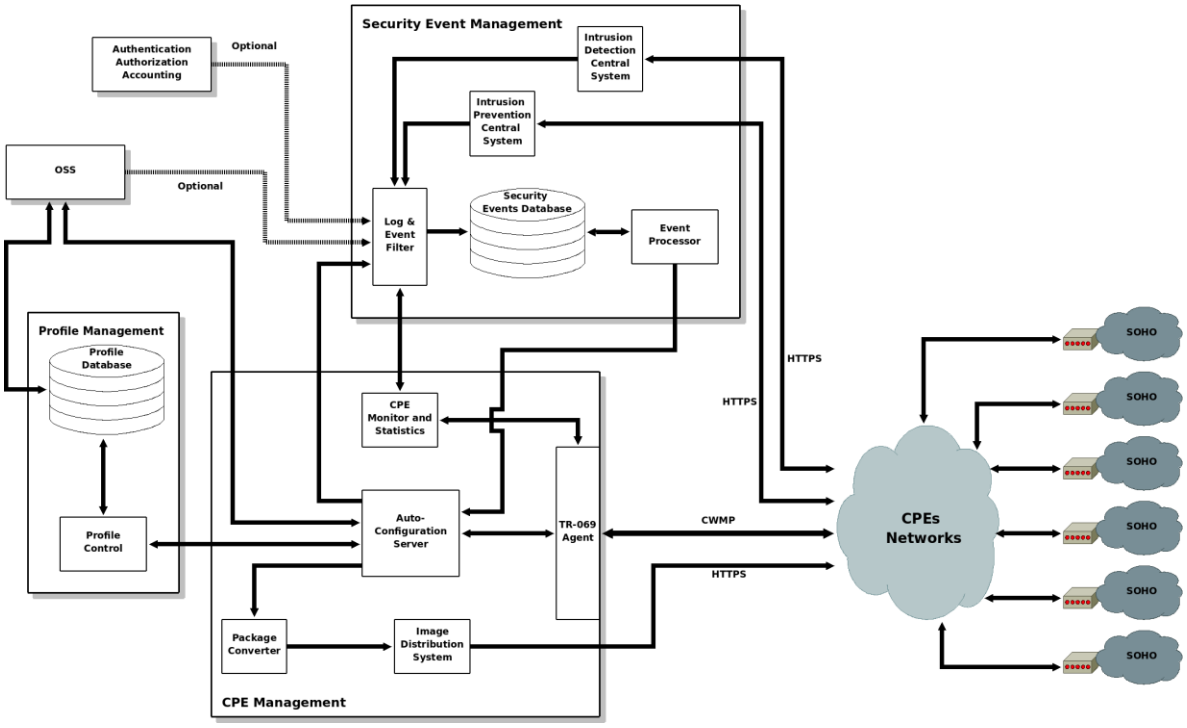

Figura 7. Arquitectura S3P (do lado da infra-estrutura do operador).

5.2 Gestão de CPE

A gestão dos CPE consiste essencialmente na configuração remota dos CPE (distribuição das aplicações e das configurações que devem aplicadas por cada CPE) e na monitorização do funcionamento do CPE (detectando e reagindo a falhas de funcionamento). Estas ferramentas servem para gestão de configuração, numa perspectiva genérica (inventário, *updates*, etc.) e são também o mecanismo de actuação remota que o ISP usa para alterar dinamicamente a forma de funcionamento dos CPE em resposta a incidentes de segurança.

A monitorização e troca de informação entre o ISP e os CPE usa canais seguros e a já referida norma TR-069, e a distribuição de aplicações e configurações segue um modelo de distribuição de imagens (pacotes).

5.3 Gestão de Perfis

A gestão de perfis assegura a manutenção de uma base de dados com perfis de equipamentos (fabricantes das CPE, modelos e versões instalados em cada cliente) e de utilizadores (serviços contratados ao ISP, preferências definidas no portal de cliente, etc.).

Do ponto de vista da plataforma S3P estes perfis são essenciais para definir que configurações devem ser enviadas para cada CPE. Estes perfis também determinam os padrões de tráfego aceitáveis para cada ambiente, de modo a que eventuais desvios possam ser detectados e eventualmente interpretados como passíveis de despoletar reacções por parte do IDS distribuído.

6. Conclusão

Nesta comunicação foram apresentados os aspectos mais relevantes da arquitectura proposta pelo Projecto S3P. Esta arquitectura distingue-se por aproveitar activamente a gateway doméstica – enquanto dispositivo de fronteira entre a rede de acesso e a rede doméstica – para criar uma plataforma distribuída de segurança, com ganhos para o operador (maior escalabilidade e granularidade, menores custos com sistemas centralizados na sua própria rede) e para o cliente. Ainda que esta abordagem pareça ir contra a visão tradicional do serviço internet – com a fronteira na rede de acesso do ISP – ela ajusta-se bem aos recentes desenvolvimentos com a introdução de redes *Triple Play* e com a crescente adopção pelos fabricantes de normas como o TR-069 para gestão remota de CPEs.

O protótipo desenvolvido mostrou que é possível implementar esta plataforma distribuída com base em ferramentas *open source* – resultando em menores custos de desenvolvimento e manutenção – e normas já adoptadas pela indústria. Esse protótipo usa para os CPE uma plataforma de *hardware* com capacidades superiores às das *gateways domésticas* actualmente comercializadas pelos ISP, mas espera-se que num prazo de 2 a 3 anos as gateways domésticas atinjam essas capacidades sem acréscimos de custo, tornando possível a massificação de plataformas como o S3P.

O próximo passo será a validação deste protótipo com utilizadores reais, numa rede piloto, de modo a que se possa depois avançar com um trabalho mais extenso de validação da escalabilidade da plataforma, por meio de medições experimentais e de métodos analíticos.

7. Agradecimentos

O trabalho de investigação subjacente a esta comunicação foi parcialmente financiado pela Fundação para a Ciência e Tecnologia, por meio do Projecto DOMUS (referência POSC/EIA/61076/2004) e pela PT Inovação, por meio do Projecto S3P.

Referências

[1] *Home Gateway Initiative*, http://www.homegatewayinitiative.org/

[2] *Broadband Forum* (ex-*DSL Forum*), http://www.dslforum.org

[3] Windows Home Server, www.microsoft.com/windows/products/winfamily/windowshomeserver

[4] HGI, Home Gateway Technical Requirements: Release 1, Version 1.0, July 2006.

[5] DSL Forum TR-069, Amendment 1, CPE WAN Management Protocol, November 2006.

[6] DSL Forum TR-124, Functional Requirements for Broadband Residential Gateway Devices, December 2006.

[7] R. Puttini, J.-M. Percher, L. Me, R. de Sousa, A fully distributed IDS for MANET, Proceedings of the Ninth International Symposium on Computers and Communications 2004 Volume 2 (ISCC"04) - Volume 02, pp. 331-338, 2004

[8] Luo Guangchun, Lu Xianliang, Li Jiong, Zhang Jun, MADIDS: a novel distributed IDS based on mobile agent, ACM SIGOPS Operating Systems Review Volume 37, Issue 1, pp. 46-53, January 2003

[9] Jing Wang, Naoya Nitta, Hiroyuki Seki, An Efficient Method for Optimal Probe Deployment of Distributed IDS, IEICE - Transactions on Information and Systems Volume E88-D Issue 8, pp. 1948-1957, August 2005

[10] PreludeIDS Technologies, http://www.prelude-ids.com

[11] H. Debar, et al, The Intrusion Detection Message Exchange Format (IDMEF), RFC 4765, March 2007

[12] Universal Plug and Play Forum, http://www.upnp.org/

[13] PUC-Rio, http://www.lua.org

Towards Intrusion-Tolerant Process Control Software

Hugo Ortiz Paulo Sousa Paulo Veríssimo
LaSIGE, University of Lisbon, Portugal
ortiz@lasige.di.fc.ul.pt, {pjsousa,pjv}@di.fc.ul.pt

Abstract

The security of critical infrastructures like water, gas or power grid control systems has been discussed more thoroughly in recent years due to recent events that have questioned their security. Terrorist groups are betting on cyber attack methods due to obvious advantages: it is cheaper than traditional methods, it is very difficult to be tracked, terrorists can hide their personalities and location, do the attack remotely from anywhere in the world, affect a large number of people, and finally, there are no physical barriers or checkpoints to cross. One has to understand that, despite some systems being considered secure, attackers will continue to discover new vulnerabilities, to try new attacks and some of those attempts will succeed. One approach to address this problem that is gaining momentum recently is intrusion tolerance. Based on this paradigm, there already are intrusion-tolerant network architectures that enhance the protection of critical infrastructures. However, even using such enhanced protection mechanisms, control systems remain with a certain level of vulnerability, which can be decreased if the process control software (PCS) itself is prepared to tolerate intrusions. This paper justifies the importance of developing intrusion-tolerant process control software and presents some insights on how to do it.

1 Introduction

Industrial control systems (ICS) are becoming one of the most relevant areas in the research of embedded-control applications. In the beginning, these systems were isolated systems running proprietary control protocols using specialized hardware and software. However, ICS are now starting to be similar to IT systems. Widely available, low cost Internet Protocol (IP) devices are now replacing proprietary solutions, which increases the possibility of cyber security vulnerabilities and incidents [26]. ICS are being designed and implemented using industry standard computers, operating systems (OS) and network protocols. Although this is essential to promote corporate connectivity and remote access capabilities, it provides notably less isolation for ICS from the outside world than predecessor systems, creating a greater need to secure these systems [14, 8, 12, 4]. It is a matter of time until hackers understand how to attack control systems underlying critical infrastructures.

It is extremely important to understand the impact differences of compromised systems. If an attacker compromises a home banking service, the system can be quickly recovered, for example, through backup databases. However, if an ICS is compromised, attackers get access to crucial resources, which may be part of critical infrastructures like water, gas or power grid control systems, and their actions will certainly have severe consequences on the

equipment being (mis-) controlled, on the services provided and on the services' clients.

Intrusion tolerance is a new approach to address accidental and malicious faults, such as attacks and intrusions, in complex and distributed systems [30]. The idea is to assume that: systems remain to a certain extent vulnerable; attacks on components or sub-systems can happen and some will be successful; and one has to ensure that the overall system remains correct and operational. Some works have taken this approach when designing group communication (e.g., [25, 6]) or protocols and services for replicated systems (e.g., quorum systems [17, 32], state machine replication [5, 2]], or a hybrid of quorums and state machine replication [1, 7]).

This leads to the idea of handling - reacting, counteracting, recovering, masking - a wide set of intentional and malicious faults (we may collectively call them intrusions), which may lead to the failure of the system security properties if nothing is done to counter their effect on the system state [28]. In short, instead of trying to prevent every single intrusion, these are allowed, but tolerated: the system has the means to trigger mechanisms that prevent the intrusion from generating a system failure.

Based on this approach, the goal of our work is to investigate ways to develop intrusion tolerant software for process control. In this way, a higher system security level can be obtained, since the system will perform its operation even in the presence of attacks and intrusions. The remainder of the paper is organized as follows: Section 2 mainly points out the relevance and the need of ICS security; Section 3 summarizes previous and ongoing work on this subject; Section 4 describes what is missing in order to better protect critical infrastructures; and finally, Section 5 presents some conclusions and directions for future work.

2 A Real Threat

Terrorism is changing. Nowadays, terrorist groups are betting on cyber attack methods due to the set of advantages it offers. First of all, cyber attacks are cheaper than traditional methods. Secondly, since an attack can be done remotely from anywhere in the world, terrorists hide their personalities and location, becoming hard to be tracked. Finally, without any physical barrier or checkpoint to cross, they can attack several targets affecting a large number of people.

Nowadays, a cyber attack can affect a whole country if some critical infrastructure (e.g., power grid) is networked through computers, which is common in most developed countries. In other words, the more technologically developed a country is, the more vulnerable it becomes to cyberattacks against its critical infrastructures.

2.1 ICS Threats and Vulnerabilities

Threats to control systems can come from numerous sources, including adversarial sources such as hostile governments, terrorist groups, industrial spies, disgruntled employees, malicious intruders, and natural sources such as from system complexity, human errors and accidents, equipment failures and natural disasters. The following is a list of possible threats to ICS [21]:

- **Attackers** - they usually break into networks for the thrill of the challenge or for bragging rights in the attacker community.

- **Bot-network operators** - they take over multiple systems to coordinate attacks and to distribute phishing schemes, spam, and malware attacks.
- **Criminal groups** - they seek to attack systems for monetary gain.
- **Insiders** - the disgruntled insider is a principal source of computer crime. Insiders may not need a great deal of knowledge about computer intrusions because their knowledge of a target system often allows them to gain unrestricted access to cause damage to the system or to steal system data.
- **Phishers** - individuals or small groups that execute phishing schemes in an attempt to steal identities or information for monetary gain.
- **Spammers** - individuals or organizations that distribute unsolicited e-mail with hidden or false information to sell products, conduct phishing schemes, distribute spyware/malware, or attack organizations (*e.g.,* DoS).
- **Spyware/malware authors** - Individuals or organizations with malicious intent carry out attacks against users by producing and distributing spyware and malware. Several destructive computer viruses and worms have harmed files and hard drives, including the Melissa Macro Virus, the Explore.Zip worm, the CIH (Chernobyl) Virus, Nimda, Code Red, Slammer, and Blaster [15].
- **Terrorists** - they seek to destroy, incapacitate, or exploit critical infrastructures to threaten national security, cause mass casualties, weaken the country's economy, and damage public morale and confidence.
- **Industrial Spies** - industrial espionage seeks to acquire intellectual property and know-how by clandestine methods.

The following lists vulnerabilities that may be found in typical ICS. Any given ICS will usually exhibit a subset of these vulnerabilities, but may also contain additional vulnerabilities unique to the particular ICS implementation that do not appear in this listing.

- **Policy and Procedure:** this kind of vulnerabilities are often introduced into ICS because of incomplete, inappropriate, or nonexistent security documentation, including policy and implementation guides (procedures). Security documentation, along with management support, is the cornerstone of any security program.
  - Inadequate security policy for the ICS;
  - No formal security training and awareness program;
  - Inadequate security architecture and design;
  - No specific or documented security procedures were developed from the security policy for the ICS;
  - Absent or deficient ICS equipment implementation guidelines;
  - Lack of administrative mechanisms for security enforcement;
  - Few or no security audits on the ICS;
  - No ICS specific continuity of operations or disaster recovery plan (DRP);
  - Lack of ICS specific configuration change management.
- **Platform Configuration:** this kind of vulnerabilities can occur due to flaws, misconfigurations, or poor maintenance of their platforms, including hardware, operating systems, and ICS applications.
  - Platform configuration:
    - OS and vendor software patches may not be developed until significantly after security vulnerabilities are found;
    - OS and application security patches are not maintained;

    - OS and application security patches are implemented without exhaustive testing;
    - Default configurations are used;
    - Critical configurations are not stored or backed up;
    - Data unprotected on portable device;
    - Lack of adequate password policy;
    - No password used;
    - Password disclosure;
    - Password guessing;
    - Inadequate access controls applied.
  - Platform hardware:
    - Inadequate testing of security changes;
    - Inadequate physical protection for critical systems;
    - Unauthorized personnel have physical access to equipment;
    - Insecure remote access on ICS components;
    - Machines with dual network interface cards (NIC) connected to different networks;
    - Undocumented assets;
    - Vulnerable to radio frequency and electro-magnetic pulse (EMP);
    - Lack of backup power to critical assets;
    - Loss of environmental control;
    - Lack of redundancy for critical components.
  - Platform software:
    - Vulnerable to buffer overflows;
    - Installed security capabilities not enabled by default;
    - Vulnerable to DoS attacks;
    - Vulnerable to packets that are malformed or contain illegal or otherwise unexpected field values;
    - Use of insecure industry-wide ICS protocols;
    - Use of clear text;
    - Unneeded services running;
    - Inadequate authentication and access control for configuration and programming software;
    - Intrusion detection/prevention software not installed;
    - Logs not maintained.
  - Platform Malware Protection:
    - Malware protection software not installed;
    - Malware protection software or definitions not current;
    - Malware protection software implemented without exhaustive testing.
- **Network:** these kind of vulnerabilities may occur from flaws, misconfigurations, or poor administration of ICS networks and their connections with other networks [23].
  - Network configuration:
    - Weak network security architecture;
    - Data flow controls (such as access control lists) not employed;
    - Poorly configured security equipment;
    - Network device configurations not stored or backed up;

    - Passwords are not encrypted in transit;
    - Passwords are not changed regularly;
    - Inadequate access controls applied.
- Network hardware:
    - Inadequate physical protection of network equipment;
    - Unsecured physical ports;
    - Loss of environmental control;
    - Non-critical personnel have access to equipment and network connections;
    - Lack of redundancy for critical networks.
- Network Perimeter:
    - No security perimeter defined;
    - Firewalls nonexistent or improperly configured;
    - Control networks used for non-control traffic;
    - Control network services not within the control network.
- Network Monitoring and Logging:
    - Inadequate firewall and router logs;
    - No security monitoring on the ICS network.
- Communication:
    - Critical monitoring and control paths are not identified;
    - Standard, well-documented communication protocols are used in plain text;
    - Authentication of users, data or devices is substandard or nonexistent;
    - Lack of integrity checking for communications.
- Wireless Connection:
    - Inadequate authentication and data protection between clients and access points.

2.2 Common Attacks

Understanding attack vectors is essential to build effective security mitigation strategies. The level of knowledge in the control system community regarding these vectors should increase in order to mitigate these risks. Effective security depends on how well the community of control system operators and vendors understand the ways that architectures can be compromised. The following is a discussion of some attacks that are usually used against ICS [18, 22, 19].

2.2.1 Backdoor Attacks via Network Perimeter

Industrial control system networks as common networking environments possess innumerable vulnerabilities and holes that can provide an attacker a 'backdoor' to gain unauthorized access. A backdoor is an undocumented way to gain access to a program, online service or an entire computer system. They are often simple shortcomings in the architecture perimeter, or embedded capabilities that are forgotten, unnoticed, or simply disregarded. Process control systems, often have inherent capabilities that are deployed without sufficient security analysis and so can provide access to attackers once they are discovered. Usually, backdoors in the network perimeter are the greatest concern (firewalls, public-facing services, and wireless access).

2.2.2 Attack into Control Systems via Field Devices

This kind of system architectures usually support the capability to remotely access terminal end points and telemetry devices through telephonic and dedicated means. Some of these devices are equipped with embedded file servers and web servers to facilitate robust communication of operational and maintenance data. However, since these devices are part of an internal and trusted domain, an attacker will try to compromise them to obtain an unauthorized vector into the control system architecture. Thus, field devices such as remote terminal units (RTUs) are viable targets to be investigated by attackers, during their reconnaissance and scanning phase of the attack. Since the connections between the devices and the control system are not monitored for malicious or suspect traffic, attackers can use the communication protocols to scan back into the internal control network. They can also alter the data that is sent to the control master or change the behavior of the device itself.

2.2.3 Database and SQL Data Injection Attacks

Database applications have become core application components of control systems and traditional security models attempt to secure systems by isolating them and concentrating security efforts against threats specific to those computers or software components. These networks usually comprise independent systems that rely on one another for proper functionality, creating an expanded threat surface. As field devices, the compromise of database applications creates additional resources an attacker can use for both reconnaissance and code execution since they usually interact with other core components of control systems. The information contained in databases makes them high-value targets for any attacker, and the cascading effect of corrupted database content can impact other core components of the system, such as data acquisition servers, historians and even the operator HMI (Human-Machine Interface)console.

2.2.4 Man-in-the-Middle Attacks

Control system environments have traditionally been (or been intended to be) protected from non-authorized persons by air gapping. In these networks, data that flows between servers, resources, and devices is often less secured. Three of the key security issues that arise from assumed trust are the ability of an attacker to (1) re-route data that is in transit on a network, (2) capture and analyze open critical traffic that is in plaintext format, and (3) reverse engineer any unique protocols to gain command over control communications. By combining all of these, an attacker can assume exceptionally high control over the data flowing in a network, and ultimately direct both real and 'spoofed' traffic [1] to network resources in support of the desired outcome. To do this, a 'man-in-the-middle', or MITM, attack is executed. Because the attack is on the control domain, this plaintext traffic can be harvested (sniffed) and taken offline for analysis and review. This allows the attacker to review and re-engineer packet and payload content, modify the instruction set to accommodate the goal of the attack, and reinject the new (and perhaps malicious) packet into the network.

[1] A spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.

2.3 Reported Incidents

Monitoring and controlling this kind of systems is an enormous undertaking, requiring constant supervision. Any single point of failure can disrupt the entire process flow and can potentially cause a domino effect that shuts down the entire system. Control systems that are not properly monitored can greatly affect the economy. Regarding power grid control systems, the following is a recap of some blackouts, which have thus far been attributed to equipment failure and lack of operator training, causing production loss, physical loss, physical destruction, and being responsible for several human fatalities.

On August 25, 2003, more than 100 electric plants were shut down, including 22 nuclear power plants, affecting 50 million people in the U.S. and Canada. This was the biggest blackout in North American history, forcing the closure of 10 major airports, causing the cancellation of 700 flights, and leaving 350,000 people stranded on the New York City subway. A broken alarm at First Energy, a northern Ohio utility company, may have allowed too much to go wrong before technicians noticed the problem. The reversal of power happened so fast that operators did not have time to react, and within about 10 seconds, vast sections of the grid were overwhelmed. The failed lines in Ohio started a cascade that crashed several systems, despite a structure built for this type of defense.[9]

On August 29, 2003, a failure of England's National Electric Grid caused a blackout in Central and Southern London affecting more than 250,000 people, 270 sets of traffic lights, and 1,800 trains. According to the latest findings, there was a fault in the volt system that apparently had not been properly maintained. [20]

On September 28, 2003, a power failure left most of Italy without power for several hours interrupting rail and air traffic and jamming emergency phone lines. Thousands were forced to take refuge in Rome's subways. As investigations revealed more information, it was found that the Italian response was either lacking, or too slow, and that Italian operators had made a wrong decision when coping with the interruption from Switzerland and France. Consequently, a cascade of power line outages resulted within Italy, and along its border. [24]

Although some concerns exist for possible sabotage, the breakdowns are reported not to be the result of terrorist or sabotage attacks. In any case, they demonstrate how much damage can be caused if every system is not properly safeguarded, monitored, and maintained. Those with the appropriate skills, knowledge, and access could generate major catastrophes and greatly hurt the country's economic stance. It is imperative that the critical infrastructure be secured in order to protect the resources of the nation and sustain its economic health.

3 State of the Art

In order to protect these systems, one has to ensure that they operate correctly despite the occurrence of accidental and malicious faults (including security attacks and intrusions). However, it is not just threats from the outside that are posing problems, but those from the inside (e.g., careless or disgruntled employees) as well.

Multiple private and public sector entities, university researchers and other professionals are working to help secure control systems. Their efforts include developing standards, providing guidance to members, reducing systems vulnerabilities, improving service responsiveness, among others.

The I3P [11] (Institute for Information Infrastructure Protection) is a consortium made up of 27 entities managed by Dartmouth College, including academic research centers, government laboratories and non-profit organizations, that was established to address security issues facing the U.S. information infrastructure. In 2005, the institute launched the Process Control Systems Security Research Project (funded by the Department of Homeland Security and the National Institute of Standards and Technology) which focuses on cyber security research on improving the robustness of the information infrastructure in the oil and gas sector. Initiatives completed include a source code checking tool, an intrusion detection and event correlation tool for process control systems, and a tool for building a business case for investing in security.

The LOGIIC [16] (Linking the Oil and Gas Industry to Improve Cyber Security) consortium brought together 14 organizations to identify ways to reduce cyber vulnerabilities in process control and SCADA (Supervisory Control And Data Acquisition) systems. The project goals were to identify new types of security sensors for process control networks, to develop better ways to protect the critical infrastructures and finally to transfer that technology and know-how to actual field operations. The result was a monitoring system based on the very latest commercial enterprise detection and correlation technologies adapted to monitor control networks, which was tested in five vulnerability scenarios based on cyber compromises commonly used in the hacker community.

IRRIIS [13] (Integrated Risk Reduction of Information-based Infrastructure Systems) is an ongoing project that aims at increasing dependability, survivability [2] and resilience of large complex critical infrastructures. The project goal is the development of a novel simulation environment called SimCIP for modeling, simulation and analysis of this kind of infrastructures. Moreover, the development of a Middleware Improved Technology (MIT) to facilitate information exchanges between different critical infrastructures is also on the scope of the project, which can help to prevent or mitigate cascading failures.

MAFTIA was the world's first project to investigate a comprehensive approach for tolerating both accidental faults and malicious attacks in large-scale distributed systems, thereby enabling them to remain operational during attack, without requiring time-consuming and potentially error-prone human intervention. This consortium brought together significant expertise from the fault tolerance, distributed computing, cryptography, formal verification, computer security and intrusion detection communities. Bringing together research groups from different disciplines resulted in novel work that bridged the gaps between those fields in many ways, including the integration of intrusion detection and fault tolerance concepts in the conceptual model, the recursive use of fault prevention and fault tolerance techniques to create trustworthy components, the use of distributed cryptography techniques for secure replication, group communication, authorization and secure trusted services, techniques for building intrusion-tolerant, intrusion detection systems, and techniques for combining cryptographic and formal methods approaches to analyze security protocols [29].

CRUTIAL (CRitical UTility InfrastructurAL Resilience) is a European project within the research area of critical information infrastructure protection, with a specific focus on the infrastructures operated by power utilities, widely recognized as fundamental to national and international economy, security and quality of life. CRUTIAL's innovative approach resides in modeling interdependent infrastructures taking into account the multiple dimensions of interdependencies, and attempting at casting them into new architectural

[2] Survivability is the ability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.

patterns, resilient to both accidental faults and malicious attacks. The project's results will help in designing and assessing new electric power systems and information infrastructures. Thus, they will enable to reduce the current (unfortunately repetitive) blackouts, in terms of frequency, duration and extent, and provide insights to electric power companies and standardization bodies for exploiting resilience in critical utilities infrastructures. One of the concrete results of CRUTIAL is an intrusion-tolerant distributed firewall, which was developed by some of the authors [31, 3, 27]. This firewall is capable of maintaining correct operation even in the presence of accidents, attacks and intrusions.

International conferences and workshops in this area, like CRITIS (International Workshop on Critical Information Infrastructures Security) or ITCIP (Conference on Information Technology for Critical Infrastructure Protection) are recently beginning to emerge. Such conferences are extremely important, since they bring together researchers and professionals from universities and private companies and public administrations interested or involved in all security-related heterogeneous aspects of Critical Information Infrastructures.

4 What is Missing?

Even if one uses all project results described in Section 3 and all security guidelines presented in Section 2, the maximum we can get is an industrial control system where is hard to penetrate and compromise the controllers. However, an attacker who can get access to a controller (*i.e.*, to the process control software), has the power to destabilize the physical controlled process (e.g., the production of electrical energy, the supply of water or gas). Therefore, our goal is to build intrusion-tolerant PCS (process control software).

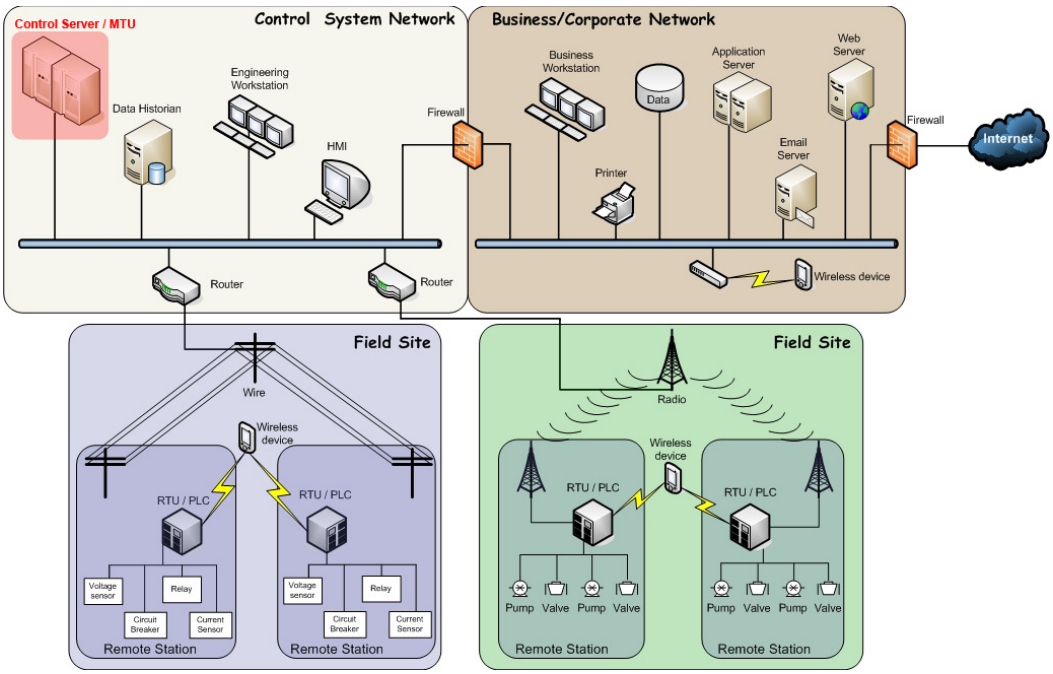

Figure 1: Typical Industrial Control System

Figure 1 illustrates a common implementation of an industrial control system. As we

can see, the control system network houses a control server (also known as MTU - Master Terminal Unit), the communication routers, the HMI (Human-Machine Interface), engineering workstations, and the data historian, which are all connected by a local area network (LAN). This part of the system is known as control center and is responsible for collecting and logging information gathered by the field sites, displaying information to the HMI, as well as centralized alarming, trend analysing and reporting. Furthermore, it may generate actions based upon detected events. The field site performs local control of actuators and monitor sensors. As mentioned in Section 2.2.2, field sites are often equipped with a remote access capability to allow field operators to perform remote diagnostics and repairs, usually over a separate dial up or WAN connection.

What was been presented in the state of the art (Section 3) mostly tries to improve the network security of the system (e.g., intrusion detection, event correlation tools, monitoring systems and advanced firewalls). As depicted in Figure 1, our goal is to make the control server (marked in red) intrusion-tolerant, that is, the PCS running in the control server should tolerate Byzantine faults, namely intrusions. In this way, even if an attacker penetrates the control system network, the control server is able to resist the attack campaign that the attacker will certainly deploy, and continue to perform its operation, albeit perhaps in a degraded mode.

Currently, there are two major techniques for building real time software with fault tolerance capabilities: *recovery blocks* and *N-version programming*. Both are based on traditional hardware fault tolerance and mainly use redundancy and diversity. The basic *recovery blocks* (RB) scheme consists of an executive, an acceptance test (AT), and primary and alternate try blocks (variants). Many implementations of RB, especially for real-time applications, include a watchdog timer (WDT) [10]. The executive orchestrates the operation of the RB technique, that is, it first attempts to ensure the acceptance test by using the primary alternate (or try block). If the primary algorithm's result does not pass the acceptance test, then n alternates will be attempted until an alternate's result passes the AT. If no alternates are successful, an error occurs. *N-version programming* technique (NVP) consists of an executive, n variants (versions), and a decision mechanism (DM). The executive orchestrates the NVP technique operation where n versions execute concurrently. The results of these executions are provided to the decision mechanism, which operates upon them to determine if a correct result can be adjudicated. If one can, then it is returned, otherwise, an error occurs.

However, these techniques cannot be directly applied in the development of intrusion-tolerant PCS because they do not address intrusions. Even if one develops different versions of a certain PCS and builds a system that runs these versions in parallel, an attacker may intrude not only the different versions/variants, but also the RB executive or the NVP decision mechanism. An attacker is an intelligent adversary that will not restrict his actions to the PCS itself, affecting also the behavior of any other components in order to fulfill his goals.

5 Conclusions and Future Work

This paper described the current status of industrial control systems (ICS) security and has shown why is it important to develop intrusion-tolerant process control software (PCS). PCS is the brain of any ICS, and therefore its correctness is vital, namely if it is being used in the context of a critical infrastructure. Current research on ICS security focus on developing security guidelines and/or protection mechanisms that decrease the probability of PCS

being affected by an attacker. However, if an attacker is able to circumvent these protection mechanisms, he will have a clear path to PCS and to the physical process controlled by it.

We have described two classic techniques that can be used to build real time software capable of tolerating (accidental) software design faults. However, these techniques do not allow to tolerate (intentional and malicious) intrusions. We are currently working on how to extend these techniques such that intrusions can be tolerated.

References

[1] M. Abd-El-Malek, G. Ganger, G. Goodson, M. Reiter, and J. Wylie, *Fault-scalable Byzantine fault-tolerant services*, Proc. of the 20th ACM Symposium on Operating Systems Principles, 2005, pp. 59–74.

[2] Y. Amir, C. Danilov, D. Dolev, J. Kirsch, J. Lane, C. Nita-Rotaru, J. Olsen, and D. Zage, *Scaling Byzantine fault-tolerant replication to wide area networks*, in Proc. Int. Conf. on Dependable Systems and Networks, 2006, pp. 105–114.

[3] A. Bessani, P. Sousa, M. Correia, N. F. Neves, and P. Veríssimo, *Intrusion-tolerant protection for critical infrastructures*, Technical Report DI/FCUL TR-07-8, Department of Computer Science, University of Lisboa (2007).

[4] E. Byres, D. Hoffman, , and N. Kube, *The special needs of SCADA/PCN firewalls: Architectures and test results*, In Proc. of the 10th IEEE Int. Conf. on Emerging Technologies and Factory Automation (2005).

[5] M. Castro and B. Liskov, *Practical Byzantine fault-tolerance and proactive recovery*, ACM TOCS **20** (2002), no. 4, 398–461.

[6] M. Correia, N. F. Neves, L. C. Lung, and P. Verissimo, *Worm-IT – a wormhole-based intrusion-tolerant group communication system*, Journal of Systems and Software **80** (2007), no. 2, 178–197.

[7] J. Cowling, D. Myers, B. Liskov, R. Rodrigues, and L. Shrira, *HQ-Replication: A hybrid quorum protocol for Byzantine fault tolerance*, Proc. of 7th Symposium on Operating Systems Design and Implementations, 2006, pp. 177–190.

[8] J.D. Fernandez and A.E. Fernandez, *Scada systems: Vulnerabilities and remediation*, Journal of Computing Sciences in Colleges **20** (2005), no. 4, 160–168.

[9] N. Gibbs, *Lights out*, Time Magazine (2003), 30–39.

[10] H. Hecht, *Fault tolerant software for real-time applications*, ACM Computing Surveys **8** (1976), no. 4, 391–407.

[11] I3P, *Institute for information infrastructure protection*, http://www.thei3p.org.

[12] V.M. Igure, S.A. Laughter, and R.D. Williams, *Security issues in SCADA networks*, Computers & Security **25** (2006), no. 7, 1–9.

[13] IRRIIS, *Integrated risk reduction of information-based infrastructure systems*, http://www.irriis.org.

[14] T. Kropp, *System threats and vulnerabilities [power system protection]*, Power and Energy Magazine **4** (2006), no. 2, 46–50.

[15] J. Leyden, *Why power plants need anti-virus*, The Register (2005).

[16] LOGIIC, *Linking the oil and gas industry to improve cyber security*, http://www.cyber.st.dhs.gov/logiic.html.

[17] D. Malkhi and M. Reiter, *Byzantine quorum systems*, Distributed Computing **11** (1998), no. 4, 203–213.

[18] T. Nash, *Backdoors and holes in network perimeters*, http://www.us-cert.gov/control_systems/pdf/backdoor0503.pdf (2005).

[19] T. Nelson, *Control systems security center common control system vulnerability*, http://www.us-cert.gov/control_systems/pdf/csvul1105.pdf (2005).

[20] BBC News, *Power cut causes chaos*, http://news.bbc.co.uk/1/hi/england/london/3189755.stm (visited January 2008).

[21] NIST, *Guide to industrial control systems (ICS) security*, Second Public Draft (2007).

[22] U.S. Department of Homeland Security National Cyber Security Division, *Control systems cyber security: Defense in depth strategies*, Control Systems Security Program (2006).

[23] McAfee White Paper, *Mitigating the top 10 network security risks in scada and process control systems*, Mcafee (2007).

[24] E. Povoledo, *Most of italy is blacked out for several hours*, New York Times (2003), Section A6.

[25] M. K. Reiter, *The Rampart toolkit for building high-integrity services*, Theory and Practice in Distributed Systems **938** (1995), 99–110.

[26] C. Smith, *Connection to public communications increases danger of cyber-attacks*, Pipeline and Gas Journal **230** (2003), no. 2, 20–24.

[27] P. Sousa, A. Bessani, M. Correia, N. Neves, and P. Veríssimo, *Resilient intrusion tolerance through proactive and reactive recovery*, Proc. of the 13th IEEE Pacific Rim Dependable Computing conference, 2007.

[28] P. Veríssimo, *Intrusion tolerance: Concepts and design principles. a tutorial*, Technical Report 02-06 **4** (2002), no. 2, 46–50.

[29] P. Veríssimo, N. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud, and I. Welch, *Intrusion-tolerant middleware: The road to automatic security*, IEEE Security & Privacy **4** (2006), no. 4, 54–62.

[30] P. Veríssimo, N. Neves, and M. Correia, *Intrusion tolerant architectures: Concepts and design*, Architecting Dependable Systems **2677** (2003), 3–36.

[31] P. Veríssimo, N. Ferreira Neves, and M. Correia, *CRUTIAL: The blueprint of a reference critical information infrastructure architecture*, Proc. of the 1st International Workshop on Critical Information Infrastructures, 2006.

[32] L. Zhou, F. Schneider, and R. Van Rennesse, *COCA: A secure distributed online certification authority*, ACM TOCS **20** (2002), no. 4, 329–368.

"Democratizando" a Filtragem e Bloqueio de Conteúdos Web

Filipe Pires[1], Alexandre Fonte[1], Vasco Soares[1]

[1] Escola Superior de Tecnologia de Castelo Branco
Instituto Politécnico de Castelo Branco
Av. do Empresário, 6000-767 Castelo Branco, Portugal.

{fpires,adf,vasco_g_soares}@est.ipcb.pt

Resumo

A filtragem e bloqueio de conteúdos Web é um assunto polémico e controverso. Para isto contribui o facto de que a maioria dos sistemas que efectuam esta actividade se encontram maioritariamente implementados em países com regimes políticos opressivos, sob a forma de mecanismos legais e tecnológicos de censura. Tal mediação vai contra os princípios gerais da Internet, uma rede global de partilha de informação pública, revogando os direitos dos utilizadores face à utilização de tais sistemas. Estes sistemas encontram-se actualmente numa fase de proliferação e existem certas áreas onde a sua aplicação se poderá tornar benéfica. Um exemplo destas áreas é a filtragem e bloqueio de conteúdo pedófilo. Neste artigo apresenta-se a arquitectura de um sistema de filtragem e bloqueio de conteúdos Web, denominado Sisbloque, projectado para ser implementado sobretudo em ISPs (Internet Service Providers), grandes instituições ou companhias, que para além de possuir um conjunto melhorado de mecanismos de filtragem de conteúdos, introduz um conceito inovador ao nível de transparência suportado pelo seu mecanismo de manipulação de erros.

1 Introdução

A Internet é cada vez mais um ambiente inseguro quer para partilha de informação quer em qualquer outra actividade, tal deve-se ao facto da criminalidade se começar a integrar de forma consistente nas novas tecnologias. Apesar das controvérsias que a filtragem e o bloqueio de acesso a conteúdos Web suscita, este método apresenta-se como uma solução eficaz frente a problemas como a publicação on-line de conteúdos pedófilos. Actualmente a maioria dos sistemas de filtragem e bloqueio de conteúdo Web, ou são soluções proprietárias ou produtos comerciais, sendo a maioria dos detalhes de implementação destes sistemas desconhecidos pela comunidade científica.

Esta lacuna motivou o desenvolvimento e concepção de um sistema aberto de filtragem e bloqueio de conteúdos Web, designado por Sisbloque. Este sistema está concebido para potencial uso em ISPs, grandes companhias ou instituições que necessitem deste tipo de serviço e propõe um mecanismo de filtragem de conteúdos mais conciso e fiável, proveniente do melhoramento de métodos existentes como a filtragem baseada na origem, filtragem de conteúdos e imagens.

O restante conteúdo deste artigo encontra-se organizado da seguinte forma. A secção 2 descreve a arquitectura do sistema Sisbloque. A secção 3 apresenta os resultados referentes a alguns testes de desempenho efectuados ao protótipo do sistema Sisbloque. Finalmente, a secção 4 conclui este artigo.

2 Visão Geral da Arquitectura Sisbloque

O sistema Sisbloque é composto por três módulos distintos que interagem entre si: o módulo de filtragem de conteúdos Web, o módulo de serviços e o módulo de manipulação de erros (ver figura 1) [1].

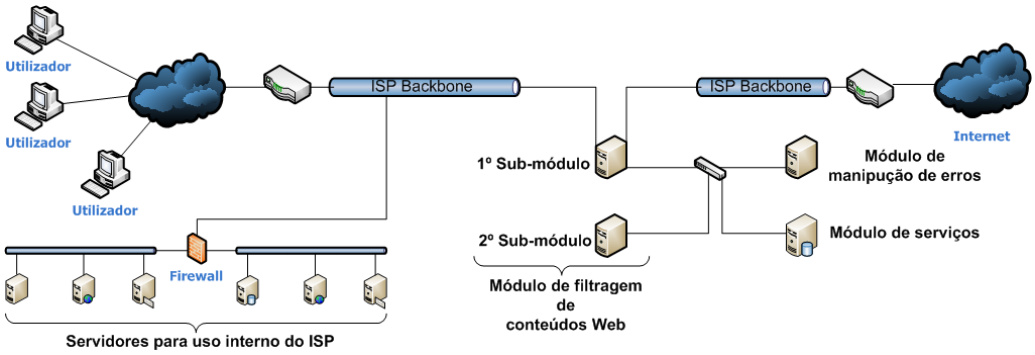

Figura 1: Visão geral da Arquitectura Sisbloque.

O módulo de filtragem de conteúdos Web é composto por dois sub-módulos: o primeiro consiste num método de filtragem baseado na origem do tráfego; o segundo é composto por um filtro de conteúdos mais específico. Esta abordagem permite tirar maior partido das vantagens de cada método, sem comprometer o sistema com as desvantagens de cada um.

O módulo de serviços é composto por todos os servidores necessários ao funcionamento do sistema, como por exemplo o suporte ás bases de dados que contêm as diversas listas de endereços, sejam estas a lista de inclusão, a lista de exclusão e lista de imunidade.

O módulo de manipulação de erros é responsável por elaborar uma mensagem de erro real, para que esta seja retornada a um utilizador sempre que este tenta aceder a um web site malicioso.

Finalmente, é importante notar que o desenho desta arquitectura procurou considerar três requisitos fundamentais, os quais devem ser observados pelo sistema Sisbloque; a saber um baixo custo de implementação e manutenção, uma elevada fiabilidade e precisão na avaliação de conteúdos Web, bem como um elevado grau de transparência.

Nas subsecções 2.1.1 e 2.1.2 são apresentados em detalhe o mecanismo de filtragem de conteúdos e o mecanismo de manipulação de erros. Na secção 2.2 é discutido uma característica chave do sistema Sisbloque, que é a sua potencial modularidade.

2.1 Tratamento de Conteúdos

2.1.1 Mecanismo de Filtragem de Conteúdos

O filtro baseado na origem do Sisbloque é o primeiro filtro do mecanismo de filtragem de conteúdos [2]. No sistema Sisbloque este filtro é constituído por duas técnicas de filtragem, uma orientada aos endereços URL contidos no protocolo HTTP e outra baseada nos endereços IP dos respectivos pacotes de dados. Estes endereços são posteriormente avaliados pelo filtro de inclusão e exclusão.

O filtro de inclusão funciona através de uma lista de acesso composta por endereços URL/IP autorizados [2-3]. Os endereços URL/IP contidos nesta lista de acesso contêm apenas informação segura, relativa a instituições de ensino, bancos, serviços governamentais, entre outros, sendo criada e mantida pelo administrador do sistema.

O filtro de exclusão funciona através de uma lista de bloqueio composta por endereços URL/IP banidos [2]. A lista de bloqueio do sistema é actualizada sempre que o filtro de conteúdos detecta um novo servidor Web malicioso, ou quando um servidor Web de conteúdo malicioso muda de endereço IP. Noutros sistemas, as listas de bloqueio são actualizadas por entidades externas, como IWF (Internet Watch Foundation) ou ECPAT (End Child Prostitution, Child Pornography and Trafficking of Children for Sexual Purpose) [4-5]. Contudo este procedimento pode induzir erros nas respectivas listas, como a introdução de servidores

Web não maliciosos, devido a falsas denúncias. De modo a garantir uma menor susceptibilidade da lista de bloqueio a ataques DNS [2], deve configurar-se no sistema Sisbloque um conjunto fiável de servidores DNS, para que este possa comparar os dados provenientes de vários servidores DNS em simultâneo.

Existe ainda o filtro de imunidade, o qual possui uma lista de endereços URL/IP referentes a web sites que não deverão ser incluídos nem na lista de acesso nem na lista de bloqueio. Tal deve-se ao facto de determinados web sites que apesar de poderem retornar conteúdo malicioso, não são directamente responsáveis por ele. Um exemplo deste caso são os motores de busca e também web sites de alojamento, entre outros. Este método garante que todo o conteúdo proveniente destes web sites será sempre convenientemente filtrado.

O filtro de conteúdos Web arquitectado para o sistema Sisbloque, é constituído por uma técnica de filtragem mais específica do que a habitual filtragem por palavras [1]. Sempre que um web site não se encontre incluído na lista de acesso, na lista de bloqueio e na lista de imunidade, o seu conteúdo é comparado as duas listas distintas, uma composta por palavras caracteristicas de web sites maliciosos e uma composta por palavras não maliciosas, onde a cada palavra maliciosa é atribuído um valor positivo e a cada palavra não maliciosa um valor negativo. Posteriormente é comparado a cotação geral do web site com o valor limite predefinido pelo administrador de sistema. Caso a cotação seja mais elevada que o limite definido então o web site em causa é considerado malicioso, sendo de imediato adicionado à lista de bloqueio do sistema.

2.1.2 Mecanismo de Manipulação de Erros

Em geral de entre os sistemas de filtragem e bloqueio de conteúdos Web que preponderam nesta área, quando é bloqueado o acesso de um utilizador a determinado conteúdo, é apresentado ao utilizador um aviso ou em alguns casos este é simplesmente deixado sem resposta. Este tipo de respostas torna perceptível, inclusivamente a utilizadores menos experientes, que algo está a bloquear o seu acesso aos conteúdos.

O grau de transparência neste tipo de sistemas é um factor de extrema importância, pois caso a metodologia de como o sistema reage quando permite ou bloqueia o acesso a conteúdos Web seja perceptível, então o sistema em causa encontrar-se-á vulnerável a possíveis ataques Oracle [6]. Segundo Lowe [6], um ataque Oracle consiste em fazer com que um sistema responda rigorosamente a qualquer número de perguntas que lhe são efectuadas, sem ter a noção das possíveis consequências. Se um sistema deste âmbito for vulnerável a este tipo de ataque, é então possível efectuar um scan de um intervalo de endereços IP, e de acordo com o tipo de resposta obtido é então construída uma lista semelhante, se não exactamente igual, à lista de bloqueio presente no respectivo sistema de filtragem. Este tipo de ataque foi efectuado e documentado sobre o sistema Cleanfeed [7].

No sistema Sisbloque sempre que um utilizador efectua um pedido de acesso a conteúdos maliciosos, este é redireccionado para o mecanismo de manipulação de erros. O mecanismo de manipulação de erros do sistema Sisbloque gera aleatoriamente erros da gama 5xx do protocolo HTTP. Posteriormente cada vez que um utilizador tenta aceder a conteúdo malicioso, é apresentado um erro gerado pelo mecanismo de manipulação de erros, iludindo o utilizador perante uma falha referente ao servidor de conteúdos a que este tentou aceder, garantindo portanto um nível mais elevado de transparência ao sistema. Através deste mecanismo torna-se imperceptível ao utilizador se o erro em causa é proveniente do servidor de conteúdos ou do sistema Sisbloque, tal garante uma maior robustez por parte do sistema a ataques deste tipo.

2.2 Modularidade do Sistema

No que diz respeito à capacidade de modularidade do sistema Sisbloque, este segue uma abordagem de implementação de um sistema distribuído. Como já foi acima referido, o

sistema Sisbloque é composto por três módulos distintos sendo estes: módulo de filtragem de conteúdos Web, módulo de serviços e módulo de manipulação de erros (ver figura 1). Destes três módulos é denotar ainda que o módulo de filtragem de conteúdos Web se divide em dois sub-módulos. Assim o sistema pode ser distribuído em vários suportes físicos, dedicando o poder de processamento e os recursos disponíveis de cada suporte físico exclusivamente a um único módulo ou sub-módulo. Através desta implementação obtém-se maior desempenho por parte de cada módulo, o que aumenta significativamente a capacidade de resposta geral do sistema em si. Dependendo dos recursos disponíveis por parte da empresa ou instituição, fica ao critério do administrador do sistema gerir a distribuição dos diferentes módulos, podendo em último recurso integrar todos os módulos em apenas um suporte físico se assim for necessário.

3 Protótipo do Sistema Sisbloque

O protótipo do sistema Sisbloque é suportado por um conjunto de componentes de software abertos, amplamente disponíveis e de uso livre, como tal a sua implementação baseia-se na integração destes componentes e de um conjunto de extensões e melhoramentos introduzidos à medida. Este conjunto tem como ambiente base de execução o sistema operativo Linux, mais especificamente a distribuição Fedora Core 8, e a respectiva kernel 2.6.25.9 a qual fornece suporte a uma framework de filtragem de pacotes designada de Netfilter [8]. A gestão de acessos a endereços URL/IP é implementada com base na integração do Web proxy Squid, juntamente com o plugin SquidGuard [9-10]. Um servidor HTTP de tecnologia Apache compõe o sistema com o propósito do Sisbloque poder usar e jogar com determinados parâmetros do protocolo HTTP durante a sua actividade de bloqueio do tráfego HTTP [11]. Para suporte de informação é usado o sistema de gestão de bases de dados relacionais, MySQL [12].

Na secção 3.1 são discutidos os detalhes do actual protótipo do sistema. Na secção 3.2 são apresentados os resultados de uma avaliação de desempenho ao protótipo.

3.1 Detalhes de Implementação

O módulo de filtragem de conteúdos Web, sendo constituído por dois sub-módulos, é o modulo mais complexo do sistema Sisbloque. O primeiro sub-módulo é introduzido num ponto específico da topologia de rede, onde passa obrigatoriamente o fluxo de dados a ser filtrado, sendo a prévia ligação existente restabelecida através de uma bridge de rede controlada pela framework Netfilter. O controlo de acessos a conteúdos Web, é também efectuado neste sub-módulo, através do Web proxy Squid, onde os pedidos a endereços IP/URL contidos na lista de bloqueios são redireccionados para o módulo de manipulação de erros. Durante este processo, é ainda enviado ao segundo sub-módulo qualquer endereço que não se encontre contido na lista de bloqueio, de acesso ou de imunidade. Cabe ao segundo sub-módulo avaliar os endereços recebidos, através do filtro de conteúdos Web. Caso seja determinado que um endereço possui conteúdo considerado malicioso, este é adicionado à lista de bloqueio do sistema que se encontra no módulo de serviços, sendo posteriormente enviado um pedido de actualização ao primeiro sub-módulo para que este actualize a sua lista de bloqueio.

No módulo de manipulação de erros reside o mecanismo de manipulação de erros. Este módulo tem como objectivo base a geração de códigos de erro aleatórios da gama 5xx do protocolo HTTP. Para tal é utilizado neste módulo o servidor HTTP Apache, o qual é periodicamente forçado a causar de forma aleatória os erros pretendidos. Sempre que um erro é gerado existe um tempo de duração que lhe é associado, isto permite que vários utilizadores não se deparem com erros diferentes no mesmo intervalo de tempo quando solicitarem conteúdo considerado malicioso.

O módulo de serviços serve de suporte ao sistema de gestão de bases de dados relacionais MySQL. É neste módulo que as diferentes listas do sistema são armazenadas, de forma a manter a sua coerência de dados e também a facilitar a sua actualização, sempre que um módulo de filtragem de conteúdos Web efectua uma actualização ou inicializa o seu processo.

De notar que de entre estes módulos, apenas o módulo de filtragem de conteúdos Web interage directamente com a rede. Os restantes módulos encontram-se isolados da rede principal, sendo apenas acessíveis pelo mesmo módulo de filtragem de conteúdos Web.

3.2 Avaliação de Desempenho

O protótipo do sistema Sisbloque encontra-se em fase de desenvolvimento, durante a qual têm sido efectuados testes ao sistema. Nesta secção discute-se os resultados obtidos relativamente à avaliação de desempenho ao qual o sistema foi submetido.

A avaliação de desempenho foi efectuada ao primeiro sub-módulo do módulo de filtragem de conteúdos Web. O seu objectivo foi consignar os recursos usados pelo respectivo módulo (ver figura 2) bem como as latências induzidas nos utilizadores (ver figura 3), quando submetido o sistema a uma sobrecarga de informação a filtrar. Este sub-módulo é composto por um processador Pentium 4 3.0GHz, por 1Gb de memória DDR 400MHz TwinMOS, por uma motherboard Gigabyte GA-8I915G-MF sendo o seu chipset Intel 915G Express.

A avaliação de recursos focou-se na percentagem de processamento usado pela unidade central de processamento, dividindo-se esta na percentagem de utilização de CPU por parte dos processos e na percentagem de utilização de CPU por parte do sistema operativo. A avaliação de latência determinou a latência provocada nos utilizadores da rede, à medida que o número de estações, que originaram a sobrecarga, foi incrementado.

No que diz respeito à sobrecarga, foi elaborado um ficheiro contendo aleatoriamente cinco mil endereços de servidores de conteúdos Web considerados maliciosos e cinco mil endereços de servidores de conteúdos Web considerados não maliciosos, denotar que nenhum destes endereços foi repetido, posteriormente este ficheiro foi dividido em vinte ficheiros os quais foram distribuídos por vinte estações. A sobrecarga teve um período de sessenta minutos no qual as vinte estações acederam ciclicamente, de um em um segundo, e em simultâneo ao conteúdo Web referente aos endereços incluídos nos respectivos ficheiros, originando desta forma tráfego HTTP a ser filtrado. Face a esta sobrecarga o sub-módulo em causa teve um aumento na sua percentagem de processamento total usado, permanecendo noventa e quatro porcento da sua capacidade de processamento disponível, foi também verificado um aumento relativamente à percentagem de memória usada contudo não foi de modo algum significativo. Durante este período foi ainda verificado que o tempo de acesso por parte dos utilizadores desta rede ao seu respectivo conteúdo sofreu alguma latência, a qual permaneceu pela média de sessenta e cinco milissegundos.

O tempo e os recursos utilizados pelo sub-módulo quando se efectua actualizações na lista de regras do Netfilter, varia consoante a localização onde o sistema é implementado na topologia da respectiva rede. A tabela de regras do Netfilter é composta principalmente pelas próprias regras que salvaguardam o sistema bem como pelas regras que reencaminham o tráfego relativo ao protocolo HTTP para o porto do Squid, adicionalmente e dependendo da localização do sistema na topologia de rede são adicionadas excepções correspondentes aos servidores internos da rede. Este número de excepções pode ser bastante reduzido ou praticamente inexistente caso o sistema seja integrado entre um servidor proxy e o respectivo gateway da rede em causa. Contudo, se esta lista possuir grande dimensão, então a sua actualização irá demorar algum tempo o que levará à indução de latências nos utilizadores, um método de evitar esta situação será a redireccionar o tráfego a ser filtrado para um segundo sub-módulo de filtragem isto enquanto o primeiro actualiza as novas definições, tornando a redireccionar o tráfego para o primeiro sub-módulo após a conclusão da actualização deste.

No respectivo protótipo do sistema a actualização quer da lista de regras do Netfilter

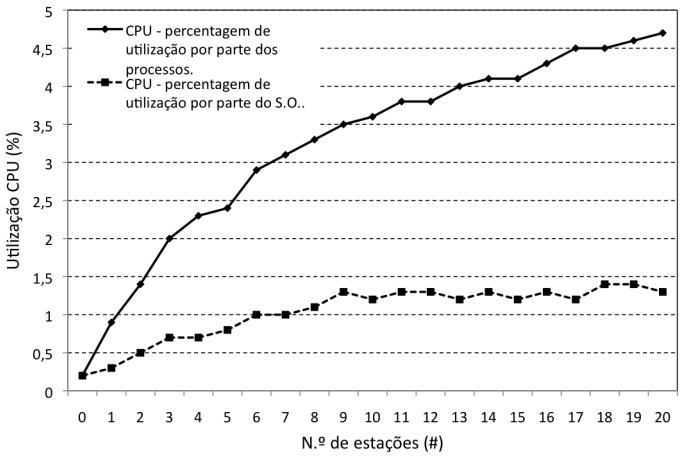

Figura 2: Percentagem de utilização de CPU.

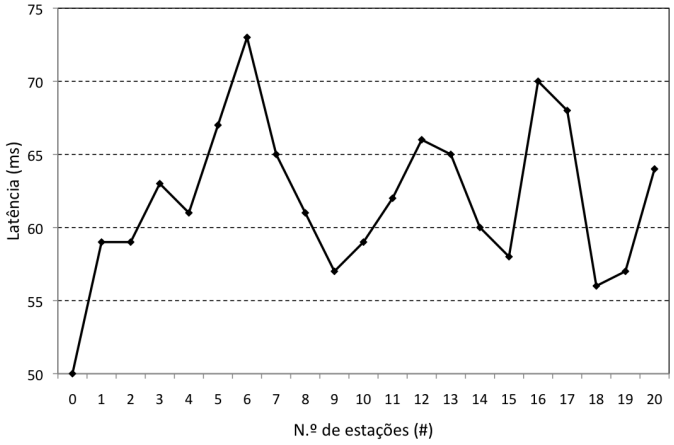

Figura 3: Latências induzidas nos utilizadores.

quer da lista de endereços do SquidGuard, embora requeira algum tempo de processamento tal não induz latência no acesso dos utilizadores.

4 Conclusão

Presentemente a criminalidade tira cada vez mais partido das novas tecnologias. Como tal, apesar da actual controvérsia em torno de sistemas de filtragem e bloqueio de conteúdos Web, estes revelam-se eficazes quando aplicados a determinados conteúdos como é o caso de conteúdos pedófilos ou qualquer conteúdo implícito a abuso de menores.

Neste artigo apresentámos a arquitectura integral de um sistema de filtragem e bloqueio desenvolvido para ser implementado sobretudo em ISPs. O Sisbloque tem como principais objectivos garantir um baixo custo de implementação e manutenção, associado a uma fiabilidade e a um sistema de filtragem e bloqueio de conteúdos Web altamente preciso. A sua arquitectura juntamente com os seus mecanismos de filtragem melhorados e o seu inovador

mecanismo de manipulação de erros, introduzem um novo conceito de transparência, o que apresenta melhorias significativas quando comparado a outros sistemas concorrentes.

Face à avaliação do protótipo do sistema concluímos que apesar do ambiente desta avaliação ser relativamente reduzido, o protótipo do sistema demonstra já uma excelente capacidade de resposta, contudo no melhor do nosso conhecimento não existe informação científica disponível relativa a testes de desempenho efectuados a sistemas concorrentes, o que impossibilita a sua comparação com o protótipo do sistema Sisbloque. O desenvolvimento deste projecto irá oferecer à comunidade científica informação única naquilo que é filtragem e bloqueio de conteúdos Web.

Referências

1. Pires, F., Fonte, A., Soares, V., "A Filtragem e Bloqueio de Conteúdos Web Segundo o Projecto Sisbloque". 3ª Conferência Ibérica de Sistemas e Tecnologias de Informação. pp. 1-6, 2008.

2. Greenfield, P., Rickwood, P., Cuong Tran, H., "Effectiveness of Internet Filtering software products", CSIRO Mathematical and Information Sciences, pp. 6-12, 2001.

3. Carlzon, M., Hagsand, O., Widell, F., Danielsson, B., "Blocking Web Contents using BGP", Royal Institute of Technology. Stockholm, Sweden, pp. 2-5, 2005.

4. Internet watch foundation, "The Internet Watch foundation", accessed at 5 October 2008, <http://www.iwf.org.uk/>.

5. Ecpat, "Ecpat sverige", accessed at 5 October 2008, <http://www.ecpat.se/>.

6. Lowe, G., " An Attack on the Needham-Schroeder Public-Key Authentication Protocol", Information Processing Letters, 56(3), p.131-133, 1995.

7. Clayton, R., Failures in a Hybrid Content Blocking System. Workshop on Privacy Enhancing Technologies. Dubrovnik, Croatia, p. 12, 2005.

8. Netfilter, "Netfilter/Iptables Project Homepage ", accessed at 5 October 2008, <http://www.netfilter.org/>.

9. Squid, "Squid-cache.org - Optimizing Web Delivering", accessed at 5 October 2008, <http://www.squid-cache.org/>.

10. Squidguard, "Squidguard", accessed at 5 October 2008, <http://www.squidguard.org/>.

11. Apache, "The Apache Software Foundation", accessed at 5 October 2008, <http://www.apache.org/>.

12. MySQL, "MySQL - The world's most popular open source database", accessed at 5 October 2008, <http://www.mysql.com/>.

A Evolução do Parâmetro de Hurst e a Destruição da Auto-Semelhança Durante um Ataque de Rede Intenso

Pedro R. M. Inácio[1,2,a], Mário M. Freire[1,b],
Manuela Pereira[1,c], Paulo P. Monteiro[2,3,d]

[1]IT-Networks and Multimedia Group
Departamento de Informática
Universidade da Beira Interior
Rua Marquês de Ávila e Bolama
6201-001 Covilhã, Portugal
{[b]mario,[c]mpereira}@di.ubi.pt

[2]Nokia Siemens Networks Portugal S.A.,
Rua Irmãos Siemens, no. 1,
2720-093 Amadora, Portugal
[a]pedro.inacio@nsn.com, [d]paulo.1.monteiro@nsn.com

[3]Instituto das Telecomunicações - Pólo de Aveiro,
Universidade de Aveiro,
3810-193 Aveiro, Portugal

Resumo

A propriedade matemática conhecida como *auto-semelhança* tem sido o assunto de inúmeras contribuições científicas na área das redes de computadores. Por definir parcialmente a natureza do tráfego em nós onde este é agregado, a referida característica pode tornar-se num potencial factor de diferenciação na presença de algumas anomalias. Este artigo resume um estudo ao comportamento do parâmetro que mede o grau de auto-semelhança (o parâmetro de Hurst) face a um ataque com expressão significativa ao nível do tráfego de rede, e avalia a resiliência da propriedade em função da intensidade daquele. A análise é conduzida recorrendo à simulação de tráfego auto-semelhante e a versões modificadas de dois estimadores do parâmetro Hurst, que permitem processamento do sinal de entrada de modo computacionalmente eficiente e ponto-a-ponto, para uma janela de valores fixa. A perca da auto-semelhança é avaliada através de dois testes estatísticos. Os resultados obtidos provam que a presença de um ataque não resulta necessariamente na destruição da auto-semelhança e que, independentemente disso, os valores devolvidos pelos dois estimadores aumentam assim que o tráfego relativo ao ataque entra na janela de observação.

1 Introdução

A propriedade matemática conhecida como *auto-semelhança* tem sido o assunto de inúmeras contribuições científicas na área das redes de computadores. A sua popularidade deve-se sobretudo ao facto de esta explicar a razão do tráfego chegar por *rajadas* a pontos de agregação, produzindo um efeito directo sobre os parâmetros de *Qualidade de Serviço* das ligações. Por definir parcialmente a natureza do tráfego de rede [8; 11; 16], alguns investigadores encontraram na auto-semelhança uma oportunidade para modelar tráfego *bem comportado*, propondo a sua análise como um meio para detectar intrusões ou anomalias [2; 5; 9; 14].

O presente documento visa descrever o impacto de um ataque no grau de auto-semelhança do tráfego de rede. Ao longo da descrição ficará mais claro que o tipo de ataques referidos recaí naqueles cuja execução adquire alguma expressão estatística ao nível da largura de banda ocupada num ponto de agregação da rede. Esses ataques são aqui designados de *ataques de rede intensos*. O estudo aqui reportado fez uso de dois estimadores do parâmetro de Hurst (que é considerada a medida do grau de auto-semelhança), que foram alterados de maneira a devolver estimativas ponto-a-ponto, e para uma janela de valores com tamanho fixo e ambulante. A evolução da propriedade pode assim ser investigada para um contexto local e de modo contínuo. O funcionamento dos dois métodos modificados é inovador, assim como a perspectiva por eles produzida. A análise é desprovida de quaisquer pressupostos iniciais acerca da preservação ou perda daquela propriedade estatística, sendo esse um dos aspectos apurados ao longo da descrição que se segue.

A parte restante deste documento está organizada da seguinte forma. A secção 2 apresenta matematicamente os principais conceitos relativos à auto-semelhança, e discute brevemente a sua expressão no tráfego de rede. A secção 3 contém uma análise crítica às referências que se debruçaram de modo mais vincado sobre o tema deste artigo. A secção 4 apresenta os métodos usados durante o trabalho de investigação na estimação do grau de auto-semelhança, e descreve brevemente o procedimento de geração de tráfego de rede. A mesma secção relata a forma de simular e injectar ataques nos registos de tráfego *legítimos*. Na secção 5 incluem-se e interpretam-se os resultados obtidos por simulação. A secção 6 apresenta as principais conclusões deste documento.

2 Auto-Semelhança e sua Expressão no Tráfego de Rede

Esta secção formaliza a propriedade da auto-semelhança e discute em que aspecto do tráfego de rede esta se manifesta.

2.1 Auto-Semelhança e Parâmetro de Hurst

A auto-semelhança é definida em termos de condições estatísticas. Um determinado processo estocástico $\{X(t)\}_{t\geq 0}$, definido para $t \geq 0$ é dito *auto-semelhante*, com *parâmetro de Hurst* H, se a equação (1) for verdadeira para qualquer valor positivo de $a \epsilon \mathbb{R}$. Note-se que o símbolo $\stackrel{d}{=}$, na referida equação, denota *igualdade em distribuição*.

$$X(t) \stackrel{d}{=} a^{-H} X(at). \tag{1}$$

Considere a particularização da definição anterior para processos com domínio temporal discreto. Nesses termos, a definição que melhor serve o propósito da explicação da auto-semelhança na área da análise de tráfego é baseada no chamado *processo das diferenças de primeira ordem* $\{Y(t)\}_{t\geq 0}$, dado por $Y(t) = X(t+1) - X(t)$. O processo $\{X(t)\}_{t \epsilon \mathbb{N}}$ é dito auto-semelhante se o seu processo das diferenças de primeira ordem respeitar a condição (2), para qualquer m positivo e inteiro. Por vezes, também se diz que $\{Y(t)\}_{t \epsilon \mathbb{N}}$ é auto-semelhante no sentido dado pela condição (2), e m é normalmente designado por *escala de agregação* [8].

$$Y(t) \stackrel{d}{=} m^{1-H} Y^{(m)}(i), \text{ onde } Y^{(m)}(i) = m^{-1} \left(X(i.m) + ... + X((i+1).m) \right). \tag{2}$$

Se a correlação de $\{Y(t)\}_{t \epsilon \mathbb{N}}$ e de $\{Y^{(m)}(i)\}_{i \epsilon \mathbb{N}}$ for a mesma para todo o $m \epsilon \mathbb{N}$, então $\{Y(t)\}_{t \epsilon \mathbb{N}}$ é dito ser *exactamente auto-semelhante de segunda ordem*. Se, por outro lado, a correlação de $\{Y(t)\}_{t \epsilon \mathbb{N}}$ e a de $\{Y^{(m)}(i)\}_{i \epsilon \mathbb{N}}$ só coincidirem para $m \to \infty$, o processo é dito ser *assimptoticamente auto-semelhante de segunda ordem*. Quando o parâmetro de Hurst é superior a 0.5 e inferior a 1, $\{X(t)\}_{t \epsilon \mathbb{N}}$ (ou $\{Y(t)\}_{t \epsilon \mathbb{N}}$ no sentido dado por (2)) exibe a propriedade da *persistência* ou da *dependência de longo-alcance*.

2.2 Expressão da Auto-Semelhança no Tráfego de Rede

O estudo que chamou a atenção da comunidade cietífica para a relação entre auto-semelhança e o tráfego de rede foi levado a cabo por Leland et al. e reportado exaustivamente em [8], publicado em 1994. O artigo entitulado *On the Self-Similar Nature of Ethernet Traffic* apresenta os resultados do estudo conduzido para registos de tráfego de uma rede de área local, implementada sobre *Ethernet*. Segundo a referida análise, a *dependência de longo-alcance* é fruto da agregação de processos com memória curta, mas cuja função de distribuição de probabilidades é uma curva com cauda alargada. Em [8; 16], cada fonte de tráfego (cada nó terminal) é modelada como uma variável independente que toma o valor 1 (=ON), sempre que o terminal está a transmitir, e o valor 0 (=OFF) quando está em silêncio. O número de bits por unidade de tempo que chegam a um determinado ponto de rede meeiro é então o resultado da agregação (soma) de diversos processos concorrentes, independentes e identicamente distribuídos, e é precisamente nessa métrica do tráfego que a auto-semelhança se revela. O *processo do número de bits por unidade de tempo* é assimptoticamente auto-semelhante de segunda ordem e a sua fisionomia é parecida com a do *ruído Gaussiano fraccionário* (rGf), um processo exactamente auto-semelhante no sentido dado por (2), com distribuição Gaussiana.

3 Trabalhos Relacionados

Dado a auto-semelhança definir parcialmente a natureza do tráfego em pontos de agregação, alguns estudos [2; 5; 9; 14] encontraram nessa propriedade uma oportunidade de categorizar o tráfego, e de detectar anomalias relacionadas com intrusões. A maior parte desses estudos [2; 5; 14] partem do pressuposto de que a auto-semelhança é perdida durante um ataque de rede intenso.

No início do artigo de Ming [9], a análise parece direccionada ao entendimento do comportamento do parâmetro de Hurst durante um ataque, mas acaba por concretizar um trabalho confuso, cujas conclusões colidem com as dos outros, e mesmo com as deste trabalho. A conclusão de que o parâmetro de Hurst decresce durante uma intrusão é fruto de uma análise ao *tamanho das unidades de dados*, ao invés de ter sido conduzida para o *processo da quantidade de informação por unidade de tempo*.

O trabalho relatado em [5] é mais vocacionado para a descoberta do melhor tamanho da janela de observação, que para a análise da auto-semelhança. O artigo é construído à volta do que os seus autores chamaram de *método de optimização*, cujo racional se resume à análise sucessiva do mesmo processo para tamanhos amostrais cada vez maiores, e à identificação do volume de pontos da melhor taxa de detecção. Após escassa discussão, e sem apresentarem quaisquer razões teóricas para o facto, o tamanho amostral de 1400 segundos (s) é indicado como aquele que onde a taxa de detecção de intrusões com duração superior a 500s é melhor.

O trabalho descrito em [14] faz referência ao estudo de [9], parecendo apoiar as suas conclusões mas, contrariamente ao esperado, no seu conteúdo é demonstrado que os valores do parâmetro de Hurst aumentam durante as anomalias investigadas. No artigo são usadas janelas de observação de 30 minutos, e os registos de tráfego são agregados para unidades de tempo que variam entre os 10ms e os 1000ms. A perca de auto-semelhança é sinalizada por desvios médios superiores a 10^{-3} entre a função de auto-correlação empírica e teórica, mas nada é dito acerca da intensidade ou duração das anomalias que podem provocar esses desvios.

O tipo de ataques que Allen et al. se propõem detectar em [2] corresponde ao tipo de ataques abrangidos pelo presente estudo. O tamanho das janelas de observação varia entre os 10 e os 30 minutos, dependendo do tamanho dos registos disponíveis e da carga de tráfego, e o valor do parâmetro de Hurst é calculado de 5 em 5 minutos. Um *ataque de exploração de tráfego* (designação usada na referência) é sinalizado quando o valor do parâmetro de Hurst

é superior a 1.0, ou inferior a 0.5. O artigo não contém um estudo à evolução do parâmetro de Hurst, nem refere a possibilidade de existirem *ataques de exploração de tráfego* que não resultem na perca da auto-semelhança.

Em nenhum dos artigos mencionados é mostrada uma evolução contínua dos valores do parâmetro de Hurst, sendo esse um dos principais factores de diferenciação do estudo aqui descrito. As simulações levadas a cabo durante este trabalho de investigação permitiram verificar uma panóplia mais abrangente de cenários de anomalia, e tirar conclusões daí. De igual modo, a implementação dos estimadores aqui proposta permitiu estudar o comportamento do grau de auto-semelhança para janelas temporais muito mais pequenas (na ordem dos 8s) que em qualquer outra contribuição científica. A interpretação teórica dos resultados não só explica fielmente os valores observados, como permite generalizar as conclusões para qualquer cenário de rede que se coadune com a simples condição do tráfego ser auto-semelhante.

4 Estimação do Parâmetro de Hurst e Simulação de Tráfego Auto-Semelhante

Esta secção apresenta os meios usados na estimação do parâmetro de Hurst, e relata brevemente o modo como o *tráfego legítimo* foi simulado computacionalmente. Note-se que no âmbito deste trabalho, a noção de *tráfego legítimo* é a mesma de *tráfego auto-semelhante*, de acordo com o que antes foi dito.

4.1 Estimação Móvel do Parâmetro de Hurst

Existem vários métodos de estimação do parâmetro de Hurst [3; 7]. A maior parte desses métodos é normalmente utilizado de modo retrospectivo, para análises conduzidas para registos de tráfego previamente capturados. Por muitas vezes se fundamentarem em estatísticas do processo auto-semelhante e das suas respectivas agregações, os estimadores dependem do processamento recorrente da mesma série de dados, pelo que apresentam uma complexidade computacional nunca inferior a $O(n \times \log(n))$ e requerem elevadas quantidades de memória.

Neste trabalho foram utilizados o método *Variância Tempo (VT)* (descrito, por exemplo, em [3]) e o método do *Processo Ramificado Embutido (PRE)* [7], por serem os que melhor se deixam moldar aos objectivos do estudo. Os dois métodos foram modificados e implementados de maneira a devolverem estimativas para uma janela de valores fixa, denominada aqui por *janela de observação*. Por motivos de falta de espaço, a formalização matemática das alterações sofridas pelos dois métodos não é aqui contida. Contudo, o conjunto de equações que formalizam as modificações para o VT pode ser encontrado em [6]. Os métodos modificados são aqui designados por VT (ou PRE) *móvel* ou *incremental*, assim se trate da versão que implementa o conceito da janela de observação, ou aquela que devolve o valor histórico (ponto-a-ponto) do parâmetro de Hurst.

Do ponto de vista conceptual, a filosofia dos estimadores móveis é simples. À instância que executa um desses métodos é pedido que devolva uma estimativa do parâmetro de Hurst cada vez que um ponto do processo em análise se torna disponível. No caso do VT, as variâncias das várias agregações do processo são actualizadas à medida que os valores do processo da *quantidade de bits por unidade de tempo* se tornam disponíveis, através da adição do efeito do valor de chegada, e da elimição do efeito do valor mais antigo na janela de observação. O valor cujo efeito foi eliminado dos cálculos é então substituido na memória pela mais recente ocorrência do processo. Contudo, deixa-se a ressalva de que o conceito da janela de observação não é directamente aplicável a todos os estimadores do parâmetro de Hurst, e que mesmo a implementação do VT ou do PRE na supramencionada filosofia, implica um efémero desvio ao seu racional inicial. Este desvio pode traduzir-se em ligeiras

instabilidades nas estimativas, que foram resolvidas pelos autores, mas cuja discussão está fora do âmbito deste artigo.

As modificações impostas ao VT e ao PRE permitiram não só construir autênticos histogramas dos valores do parâmetro de Hurst (ver secção 5), como também a abstracção ao problema levantado pela *lei dos grandes números*, que limitava fortemente a sensibilidade dos estimadores a alterações provocadas por mudanças nas propriedades do tráfego. Os estimadores retrospectivos, ainda que aplicados de maneira a devolver valores ponto-a-ponto, rapidamente perdem a capacidade de notar pequenas alterações, penalizados pelo peso estatístico da história da análise.

4.2 Simulação de Tráfego Auto-Semelhante

Os resultados contidos na secção 5 foram obtidos através de simulação computacional. Dado a especificidade do problema não requerer mais do que emulação do tráfego ao nível inferior da camada de ligação de dados, todos os registos de tráfego foram modelados como sequências de *tamanhos de pacotes* e *intervalos entre chegadas*. Convencionou-se que os tamanhos de pacotes eram incidências de uma variável aleatória com distribuição empírica conhecida (por exemplo de [10]) e que, por conseguinte, o seu valor esperado $E(P_t)$ era sabido à priori. A simulação de uma *Carga* de rede efectiva C (dada em relação a uma Largura de Banda total de LB) é conseguida através da geração de n_p *tamanhos de pacotes* e n_p *intervalos entre chegadas* por unidade de tempo, em que n_p é dado por:

$$n_p = \frac{LB \times C}{E(P_t)}. \tag{3}$$

Nestas circunstâncias, a média dos tempos entre chegadas $E(IC_t)$ é necessariamente descrita por (4) e, dado $\{IC_t\}_{t\epsilon\mathbb{N}}$ ser inferiormente limitada por um valor mínimo positivo IC_{min}, decidiu-se que o seu intervalo de variação se devia confinar a $[IC_{min}, 2 \times (E(IC_t) - IC_{min})]$.

$$E(IC_t) = \frac{LB \times (1 - C)}{n_p}. \tag{4}$$

A impressão da estrutura fractal (auto-semelhança) no *processo da quantidade de informação por unidade de tempo* foi conseguida através da modelização dos tempos entre chegadas como sendo incidências de rGf, de acordo com a expressão (5), onde rGf_t^H denota a ocorrência t de um rGf com parâmetro de Hurst H. Note-se que o desvio padrão do processo ($\sigma = (E(IC_t) - IC_{min})/3$) foi escolhido de modo a que, 99.7% das vezes, o valor produzido pela implementação de (5) esteja contido no intervalo de variação definido. O gerador de tráfego usado neste trabalho de investigação trunca automaticamente todos os valores que estejam fora do referido intervalo, para evitar inconsistências. A simulação de rGf é feita através de um método baseado em *Onduletas*, descrito em [3].

$$IC_t = rGf_t^H \times \frac{E(IC_t) - IC_{min}}{3} + E(IC). \tag{5}$$

4.3 Simulação de Ataques de Rede Intensos

Antes de prosseguir com a descrição do modo de como os ataques foram simulados, é pertinente o comentário ao tipo de anomalias que um método baseado em auto-semelhança tem possibilidades de detectar. Dado a princípal estatística em análise estar dependente de uma quantidade amostral necessariamente grande (o parâmetro de Hurst reflecte *dependências de longo-alcance*), qualquer ataque constítuido por um número de pacotes expressivamente pequeno tem poucas hipóteses de ser detectado por este tipo de análise. O interesse incide, portanto, naqueles ataques que, a determinada altura da sua investida, ocupam uma

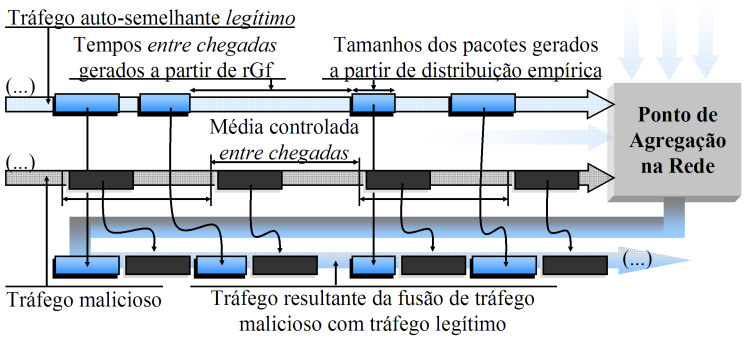

Figura 1: Representação gráfica do procedimento usado para *injectar* as unidades de dados de um ataque simulado no registo de tráfego auto-semelhante.

quantidade de largura de banda não negligenciável. São estes os ataques que aqui são denominados de *ataques de rede intensos*. Várias instâncias de ataques de negação de serviço (ataques a protocolos ou de inundação, distribuídos ou não) [13; 15] recaem precisamente na categoria indicada. Note-se que a definição destes ataques vai de encontro ao que foi dito sobretudo em [2; 9]. Em [9] é aliás enfatizada e utilizada a designação de *ataques de largura de banda*, sugerida pela *Computer Emergency Response Team* (CERT) [4].

A simulação dos ataques de rede intensos foi feita recorrendo à especificação do parâmetro de Intensidade (I), que determina a quantidade de pacotes que chega ao nó fictício, por unidade de tempo e em função da largura de banda disponível. Depois de se escolher o tamanho do pacote malicioso (por exemplo, o tamanho de um pacote SYN do *Transmission Control Protocol* (TCP)), calcula-se a média do ritmo de geração dos pacotes maliciosos. Devido ao facto do gerador de tráfego legítimo apresentado permitir a sua representação conceptual em termos de sequências de unidades de dados, a injecção do tráfego relativo ao ataque pode ser conseguida pela implementação directa do procedimento ilustrado pela figura 1. As unidades de dados do tráfego malicioso são simplesmente inseridas no tráfego legítimo por ordem de chegada, podendo isso incorrer no atraso de ambos os tipos de tráfego. A análise da auto-semelhança é feita para o fluxo resultante da fusão do tráfego malicioso com o legítimo. Na figura 1, esse fluxo está representado em baixo, ilustrado como o único que *sai* do equipamento de agregação.

5 Análise dos Resultados

Esta secção está dividida em quatro partes. As duas primeiras partes são dedicadas à análise da evolução do parâmetro de Hurst durante ataques de rede intensos. Os dois cenários estudados exploram a possibilidade da duração do ataque ser ou não superior à janela de observação dos estimadores móveis utilizados. A terceira parte é dedicada aos testes estatísticos de resiliência da auto-semelhança. A última subsecção elabora numa possível interpretação dos resultados.

5.1 Duração do Ataque Menor que o Tamanho da Janela de Observação

Um dos primeiros cenários observados com as ferramentas antes mencionadas foi aquele em que a duração do ataque é menor do que o tamanho da janela de observação. A representação gráfica do lado esquerdo da figura 2 mostra um dos cerca de 150 histogramas construídos e analisados empiricamente durante esta parte do trabalho de investigação. O cenário simulado é o de um ponto de agregação capaz de operar a 1Gbps, mas cuja carga

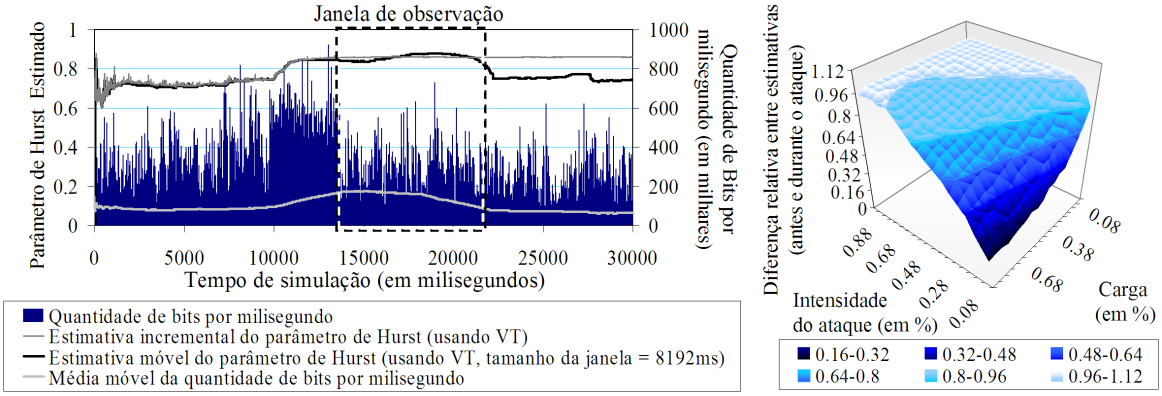

Figura 2: Representação gráfica de alguns dos resultados relativos às simulações com ataques de duração inferior ao tamanho da janela de observação (o tamanho da janela de observação era de 8,192s, a duração do ataque era de 4s, a carga era de 10% e a intensidade do ataque de 10%). Do lado esquerdo pode ver-se o histograma da evolução do parâmetro de Hurst (calculado através do estimador baseado no VT); enquanto que no lado direito é incluído o gráfico da diferença máxima entre estimativas do parâmetro de Hurst obtidas antes e durante o ataque, para diferentes combinações de carga e intensidade do ataque.

útil é de 10%. O parâmetro de Hurst do gerador de tráfego legítimo foi inicializado a 0.75, e um ataque com intensidade de 10% e duração de 4s foi injectado aos 10s de simulação. Para além da *quantidade de bits por milisegundo* e da respectiva média móvel (calculada para uma janela de observação de 8192ms), são também apresentadas no gráfico as curvas de evolução do parâmetro de Hurst, calculado usando o VT móvel e incremental. O tamanho da janela de observação está também representado (à escala) na figura.

Como se pode ver, depois de um período inicial de instabilidade, os valores do parâmetro de Hurst tendem para o valor esperado de 0.75. Assim que o ataque começa, os mesmos valores aumentam para 0.86 (aproximadamente) e variam em torno desse valor durante $4000 + 8192 \approx 12000$ms. Logo que o registo de tráfego contendo o ataque *abandona* a janela de observação, os valores do estimador móvel decrescem novamente para próximo de 0.75. O facto das estimativas do parâmetro de Hurst se manterem elevadas durante um período de tempo que supera o da duração do ataque está relacionado com o tamanho do mesmo. Como será explicado com mais detalhe em baixo, a entrada do ataque no estimador móvel corresponde a uma translação do processo analisado, que é inicialmente entendida como uma mudança no grau de auto-semelhança. Este *novo* estado em que o VT móvel se encontra é o resultado da presença de dois tipos de tráfego dentro da janela (legítimo, e legítimo+malicioso), que só é anulado depois do tráfego relativo ao ataque abandonar *por completo* a janela de observação. Note-se ainda que as estimativas devolvidas pelo VT incremental permanecem elevadas, mesmo depois do ataque terminar, já que o seu efeito demora a desaparecer da memória do estimador retrospectivo.

De maneira a obter uma ideia mais clara do comportamento do parâmetro de Hurst face a diferentes cenários de rede, foi desenhado um procedimento para testar (repetidas vezes) cerca de 324 combinações do par (C, I). De entre várias estatísticas, o procedimento devolvia a diferença máxima entre valores do parâmetro de Hurst local, antes e durante o ataque, o momento apontado como sendo o início do ataque e a duração do mesmo (o início do ataque era sinalizado por estimativas locais do parâmetro de Hurst superiores ao valor esperado em cerca de 0.01, por períodos de tempo superiores a 100ms; o fim do ataque correspondia ao *regresso* do parâmetro de Hurst ao valor esperado). Para facilitar a sua análise crítica, as diferenças máximas entre valores do parâmetro de Hurst foram normalizadas (divididas pelo supremo de todos os valores calculados) e representadas, como

uma função da carga de rede e da intensidade do ataque, no gráfico do lado esquerdo da figura 2. As referidas diferenças atingem máxima expressão para aproximadamente metade das combinações simuladas, tornando-se mais notáveis à medida que a intensidade do ataque aumenta.

5.2 Duração do Ataque Maior que o Tamanho da Janela de Observação

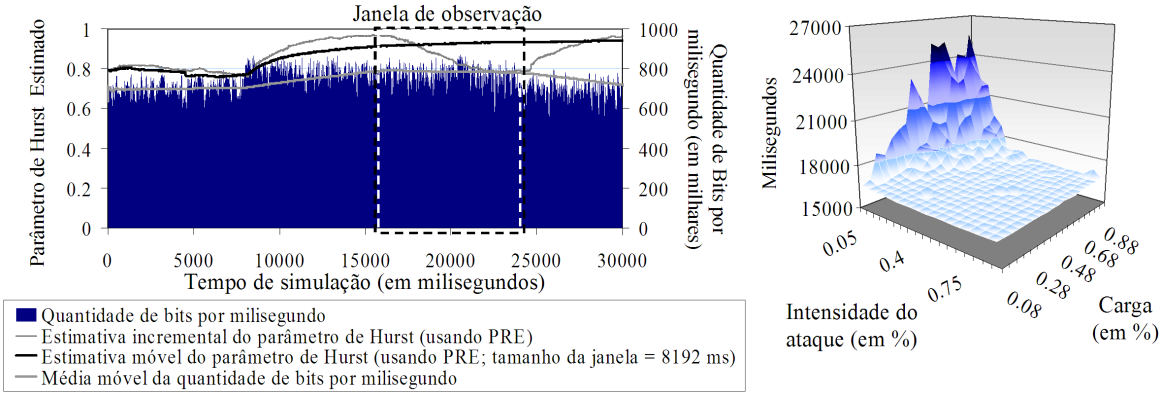

Figura 3: Representação gráfica de alguns dos resultados relativos às simulações com ataques de duração superior ao tamanho da janela de observação (o tamanho da janela de observação era de 8,192s, a duração do ataque era de 16s, a carga era de 70% e a intensidade do ataque de 10%). Do lado esquerdo pode ver-se o histograma da evolução do parâmetro de Hurst (calculado através do estimador baseado no PRE); enquanto que no lado direito figura a média dos momentos apontados como o início das anomalias.

O segundo tipo de cenário encenado diz respeito à situação onde a duração do ataque supera o tamanho da janela de observação. Uma possível ilustração deste cenário é incluída no lado esquerdo da figura 3. Neste caso, o gerador de tráfego foi instruido a simular uma carga de rede de 70% e um ataque com 10% de intensidade aos 8s. O parâmetro de Hurst do tráfego legítimo foi ajustado para 0.80 e os respectivos métodos de estimação eram baseados no PRE. O tamanho da janela de observação valia metade da duração da anomalia.

Como se pode verificar, e contrariamente ao que acontecia anteriormente, a estimativa móvel do parâmetro de Hurst regressa ao valor esperado ainda durante a análise do ataque. Isto acontece porque, assim que o registo de tráfego contendo o ataque passa a dominar o contexto do PRE móvel (note-se que isto também se aplica ao VT móvel), o método é incapaz de distinguir o processo em análise de uma translação do processo inicial (e legítimo). Logo que o ataque termina e começa a sair da janela de observação, o PRE detecta novamente a translação e as suas estimativas aumentam novamente até que o efeito daquele desaparece da janela de observação.

No lado direito da figura 3 foi incluida a representação da média dos momentos apontados como o início dos anomalias, em função da sua intensidade e da carga de rede. Qualquer ataque com expressão superior a 3% (em termos de largura de banda total) produz efeito suficiente para o detector antes descrito apontar com exactidão o momento em que processo muda. Para a maior parte dos casos, o início do ataque é apontado estar entre os 16 e os 17s.

5.3 A Destruição da Propriedade da Auto-Semelhança

De modo a investigar se a auto-semelhança é ou não perdida durante um ataque de rede intenso, foram implementados dois testes estatísticos diferentes. O teste de Kolmogorov Smirnov (*teste K-S*) foi usado para apurar se a distribuição de um registo de tráfego contendo um ataque é ou não *semelhante* à distribuição de várias agregações do processo em análise. O segundo teste avalia a qualidade da estimativa devolvida pelo VT (para mais detalhes, considere a leitura das referências [3; 6]), através do estudo da estatística conhecida como o *coeficiente de determinação*, normalmente simbolizada por R^2 [1].

A introdução de tráfego de um ataque de rede intenso corresponde a uma translação do processo auto-semelhante, e a uma consequente mudança das suas propriedades. A média móvel do processo aumenta, tal como a sua variância. Antes do início do ataque, e até durante o mesmo, a auto-semelhança é mantida a nível local, mas o mesmo pode não acontecer durante o período em que a janela de observação transita de tráfego legítimo para o registo contendo unidades de dados do ataque. Foi precisamente neste período de tempo que a análise foi efectuada.

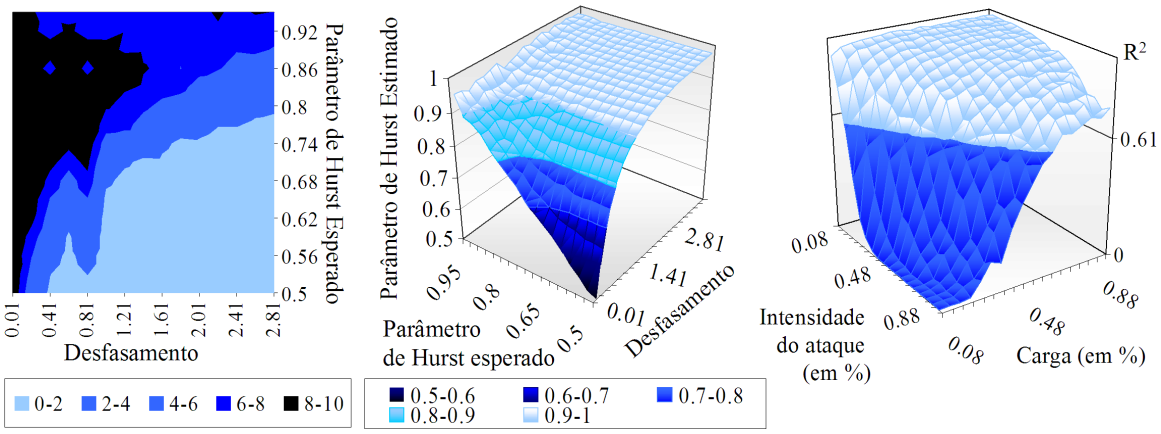

Figura 4: Compilação dos resultados dos testes à resiliência da auto-semelhança. Do lado esquerdo, representa-se o número médio de testes Kolmogorov Smirnov bem sucedidos (de um total de 10), em função do grau de auto-semelhança e do desfazamento aplicado ao sinal; ao centro mostra-se a superfície que define a diferença entre o parâmetro de Hurst do sinal auto-semelhante e o do transladado; e do lado direito representa-se a *qualidade* da regressão linear do VT, de acordo com a estatística R^2.

Foram simulados vários registos do período temporal em que o ínicio do ataque está algures no meio da janela de observação. Depois, o *processo da quantidade de bits por unidade de tempo* foi normalizado e agregado k vezes, vindo posteriormente a produzir-se a distribuição de probabilidade para todos esses processos. De modo a aplicar o *teste K-S*, foram tiradas as k maiores distâncias D_k, entre todas as distribuições dos processos agregados e a distribuição do processo inicial. Os vários D_k foram então comparados com valores críticos (para o grau de significância de 0.01) tabelados [12], sendo considerados como *bem sucedidos* no caso em que D_k era menor que o valor crítico. Simultaneamente, era pedido à classe que implementava o VT móvel que devolve-se o valor R^2 da pior regressão obtida ao longo da *sua* análise.

Os gráficos contidos na figura 4 resumem parte da investigação conduzida para um cenário em que a carga de rede era de 50% e a janela de observação era de 8192ms (os testes foram aplicados para $k = 10$, para escalas de agregação de $2^i, i = 1, ..., 10$). Os resultados representados à esquerda da figura mostram que a resiliência ao desfazamento provocado à sequência de valores em análise depende do grau de auto-semelhança da mesma. Apesar da dependência não aparentar ter uma fórmula explícita que a explique, facilmente

se depreende que à medida que o parâmetro de Hurst aumenta, maior é o desfazamento suportado pelo processo, antes da destruição da auto-semelhança. Repare-se que, neste caso, os autores consideraram que a perca da referida propriedade implicava o falhanço em pelo menos 2 dos 10 testes aplicados. Note também que para valores do parâmetro de Hurst entre 0.75 e 0.85, o processo analisado parece ser especialmente resistente às transformações a que foi sujeito, sendo capaz de suportar intensidades de aproximadamente 30% (igual ao desvio padrão do processo), antes de perder a auto-semelhança.

A última fase da maior parte dos estimadores do parâmetro de Hurst compreende a aproximação de determinado número de coordenadas via regressão linear. A adequação do modelo resultante dessa regressão, dada pelo valor de R^2, pode ser entendida como uma medida do *quão bem* a auto-semelhança se reflecte no processo em análise. R^2 varia entre 0 e 1, argumentando em favor da qualidade do método para valores próximos do limite superior. O gráfico colocado à direita da figura mostra que a introdução de ataques no tráfego auto-semelhante afecta a lei exponêncial em que o método VT se baseia. Contudo, se a perca da propriedade estudada fosse definida em função de R^2 (ou do chamado *teste F* [1] que se pode aplicar à transformação dada por $F = (k-2) \times R^2/(1-R^2)$), só após uma intensidade considerável é que se podia concluir acerca do falhanço da aproximação linear aos vários logaritmos das variâncias. A verdade é que o valor de R^2 (e consequentemente de F) se mantém elevado mesmo na presença de ataques com intensidade média, pelo que o teste mencionado não é capaz de descartar a possibilidade de existir uma relação exponêncial (fractal) entre o processo e suas agregações. Note que a linha de decisão da perca da auto-semelhança é a que divide a superfície nas duas secções com cores diferentes.

O gráfico do meio da figura 4 foi incluído com o objectivo de mostrar o comportamento das estimativas do parâmetro de Hurst, perante os cenários anteriormente descritos e testados. Como se pode observar, o sucesso ou falhanço dos *testes K-S*, ou a qualidade da regressão do VT, não parecem influenciar directamente os valores devolvidos pelos estimadores, que se aproximam imperetrivelmente de 1 (sem nunca o ultrapassar) à medida que o desfazamento aumenta.

5.4 Interpretação dos Resultados

Dos resultados incluídos anteriormente conclui-se que (i), a auto-semelhança não é necessariamente destruída pela presença de um ataque de rede intenso e que (ii), o parâmetro de Hurst aumenta *sempre* que um fluxo constante de tráfego é injectado na rede. Nesta secção propõe-se uma possível interpretação para estes factos, com base na teoria subjacente à auto-semelhança.

O processo das diferenças de primeira ordem de um processo auto-semelhante (definido em (2)) é parcialmente dominado por componentes constantes, cuja duração, amplitude e sinal, determinam as suas propriedades fractais. O parâmetro de Hurst aumenta de 0.5 para 1 à medida que a extensão e magnitude da parte constante aumenta. Durante os períodos de normal funcionamento da rede, as referidas componentes são o produto de vários fluxos de informação, gerados por nós remotos e direccionados até ao ponto de agregação, onde alimentam continuamente o *processo da quantidade de informação por unidade de tempo*, conferindo-lhe propriedades mais ou menos *persistentes*. Durante um ataque de rede intenso, os dois factores acima mencionados (duração e amplitude) são ambos afectados positivamente, e a *persistência* (a parte constante) do sinal é reforçada. Neste caso, e apesar de não serem conceitos equivalentes, a *constância* pode fortalecer a *auto-semelhança* ou até destruí-la, mas nunca diminuí-la.

É óbvio que a inserção de tráfego malicioso resulta sempre numa perca de estacionariedade, mas essa perca pode não resultar na destruição da auto-semelhança (a figura 3 ilustra um cenário que não resulta na perca da referida propriedade). É sabido, por exemplo de [3], que a estacionariedade dos *processos com dependências de longo alcance* é difícil de avaliar, já que a própria natureza dos processos fractais dita o deslocamento das pro-

priedades estatísticas a nível local. Estes deslocamentos podem, numa primeira análise, ser confundidos com falta de estacionariedade mas, na verdade, são apenas viés definidos pelas (auto) correlações. A introdução de modestos (em relação à carga de rede) fluxos de tráfego malicioso pode apenas resultar num deslocamento local e pequeno da largura de banda ocupada, que aumenta a auto-semelhança. Alheios à qualidade exacta do sinal de entrada, tudo o que os estimadores utilizados são capazes de observar é, basicamente, a transformação de um processo variável em um mais estável, para o qual a estimativa do parâmetro de Hurst *só* pode ser superior.

6 Conclusão

Este artigo resume a análise detalhada à evolução do grau de auto-semelhança durante intrusões com expressão significativa ao nível da largura de banda. Todos os resultados aqui contidos foram obtidos por simulação, mas corroborados por dois estimadores do parâmetro de Hurst diferentes. A possível perca da auto-semelhança, durante os referidos ataques, é testada recorrendo a duas abordagens estatísticas completamente diferentes, e todos os resultados são analisados do ponto de vista teórico, embora de modo breve.

A especificidade da análise delimita perfeitamente os dois cenários tomados em consideração nas simulações. Para o caso em que a duração do ataque é inferior ao tamanho da janela de observação dos estimadores, as estimativas do parâmetro de Hurst mantêm-se acima da média (ou daquilo que era esperado) enquanto o tráfego relativo ao ataque (ou parte desse tráfego) se encontra *dentro* da janela de observação. No caso oposto, as estimativas começam por subir assim que o tráfego malicioso começa a *chegar* no estimador, descendo até ao valor esperado assim que a janela de observação é completamente obsorta pela mistura de tráfego malicioso e legítimo. Assim que o ataque termina, regista-se de novo um aumento das estimativas do parâmetro de Hurst, que decresce logo que o tráfego relativo ao ataque *deixa* por completo a janela de observação.

Ficou demonstrado que a presença de um ataque de rede intenso não resulta *necessariamente* na destruição da propriedade da auto-semelhança. Na verdade, a destruição desta propriedade depende da expressividade da perca de estacionariedade, que por sua vez depende da intensidade do ataque, e da carga da rede. Por exemplo, a presença de um ataque com expressão relativamente modesta resulta apenas no aumento da auto-semelhança. Independentemente da perca ou preservação da estrutura fractal, os valores devolvidos pelos dois estimadores utilizados aumentam durante a presença do ataque, assinalando apenas um *aparente* aumento do grau de auto-semelhança do tráfego.

Aliados à baixa complexidade computacional dos estimadores utilizados, os resultados aqui contidos sugerem que a análise da auto-semelhança pode ser eficientemente utilizada para detectar anomalias, cuja natureza *pode* estar relacionada com ataques de rede. O método que elabora nesta análise pode ser posicionado imediatamente após os mais simples colectores de dados de tráfego, levantando alertas que podem despoletar novas investigações. Contudo, e também devido à especificidade das métricas em que se baseia, o seu juízo solitário não pode decidir a legitimidade ou ilegitimidade do tráfego.

Agradecimentos

Os autores gostariam de agradecer o apoio financeiro da *Fundação para a Ciência e Tecnologia, Portugal* (formalizado pelo contrato no. SFRH/BDE/15592/2006), da *Nokia Siemens Networks Portugal S.A.* e do projecto PTD-C/EIA/73072/2006 TRAMANET: *Traffic and TrustManagement in Peer-to-Peer Networks*. Estão igualmente gratos a João Gomes, por criticar construtivamente este trabalho.

Referências

[1] Michael Patrick Allen, *Understanding regression analysis*, Humanities, Social Sciences and Law, ch. The coefficient of determination in multiple regression, pp. 91–95, Springer US, Novembro 2007, Free Preview.

[2] W.H. Allen and G.A. Marin, *The LoSS Technique for Detecting New Denial of Service Attacks*, SoutheastCon, 2004, Florida Institute of Technology;, IEEE, Março 2004, pp. 302–309.

[3] Ton Dieker, *Simulation of fractional Brownian motion*, Master's thesis, University of Twente, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, 2004.

[4] Fengmin Gong, *Deciphering Detection Techniques: Part III Denial of Service Detection*, White paper, McAfee Network Security Technologies Group, Janeiro 2003.

[5] Mohd Yazid Idris, Abdul Hanan Abdullah, and Mohd Aizaini Maarof, *Iterative Window Size Estimation on Self Similarity Measurement for Network Traffic Anomaly Detection*, Int. Journal of Computing and Information Sciences (IJCIS) **4** (2005), no. 4, 88–91.

[6] Pedro R. M. Inácio, Mário M. Freire, Manuela Pereira, and Paulo P. Monteiro, *Analysis of the Impact of Intensive Attacks on the Self-Similarity Degree of the Network Traffic*, The Second International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2008) (Cap Esterel, France), 2008, pp. 107–113.

[7] OD Jones and Y. Shen, *Estimating the Hurst index of a self-similar process via the crossing tree*, Signal Processing Letters, IEEE **11** (2004), no. 4, 416–419.

[8] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, *On the self-similar nature of ethernet traffic (extended version)*, Networking, IEEE/ACM Transactions on **2** (1994), no. 1, 1–15.

[9] Ming Li, *Change trend of averaged Hurst parameter of traffic under DDOS flood attacks*, Computers & Security (2006), no. 3, 213–220.

[10] NLANR, *NLANR - National Laboratory for Applied Network Research - Internet measurement, Internet analysis*, 2005, Acedido a 29 de Março de 2008.

[11] I. Norros, *Studies on a Model for Connectionless Traffic, Based on Fractional Brownian Motion*, COST24TD(92)041, 1992.

[12] Paul Wessel, *Critical Values for the Two-sample Kolmogorov-Smirnov test (2-sided)*, acedido a 27 de Agosto de 2008.

[13] Vern Paxson, *An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks*, ACM Computer Communications Review (CCR) **31** (2001), no. 3.

[14] Mohd Fo`ad Rohani, Mohd Aizaini Maarof, Ali Selamat, and Houssain Kettani, *Uncovering Anomaly Traffic Based on Loss of Self-Similarity Behavior Using Second Order Statistical Model*, IJCSNS International Journal of Computer Science and Network Security **7** (2007), no. 9.

[15] Bennett Todd, *Distributed Denial of Service Attacks*, February 2000, Acedido a 30 de Março de 2008.

[16] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson, *Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level*, IEEE/ACM Transactions on Networking **5** (1997), no. 1, 71–86.

Índice de Autores

ISBN 978-989-96001-0-2

Organização:

CISUC

Apoio:

IPN lis
CERT-IPN COMPUTER SECURITY INCIDENT RESPONSE TEAM

Patrocinadores:

Microsoft

ANACOM
AUTORIDADE NACIONAL DE COMUNICAÇÕES