AutoStrap: Bootstrapping Seguro de Redes 6LoWPAN para Prevenir Ataques Vampiro em Ambientes da Internet das Coisas

Tiago Diogo e Miguel Pardal

Instituto Superior Técnico, Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal,
{tiago.diogo,miguel.pardal}@tecnico.ulisboa.pt

Resumo A Internet das Coisas (IdC) e a sua visão de ligar dispositivos entre si e à Internet apresenta-se como uma oportunidade para criar grandes redes de partilha de informação. No entanto, intrusos podem capturar e debilitar estar redes tomando vantagem dos seus limitados recursos. No nosso trabalho propomos AutoStrap – um método de *bootstrapping* seguro capaz de assegurar ambientes da IdC baseados em redes 6LoWPAN – que visa prevenir ataques de esgotamento de bateria ("vampiros") com o âmbito de colocar a rede *offline*. Para atingir este objetivo, realizamos uma análise extensiva aos protocolos, gamas de ataques e estratégias de mitigação existentes, combinando essa informação no nosso sistema proposto de gestão de redes para espaços inteligentes. Ademais, conduzimos medições ao nível do espaço utilizado e recursos consumidos para entender quais os recursos físicos necessários para este tipo de aplicações.

Keywords: Internet das Coisas; Bootstrapping Seguro; Ataques Vampiro; CoAP; 6LoWPAN; RPL; IEEE 802.15.4

1 Introdução

A Internet das Coisas pode ser vista como uma teia de dispositivos interligados entre si que vão desde *wearables* até redes de sensores de gama empresarial. Apesar da enorme variedade e diferenças entre estes dispositivos, algo que todos têm em comum é a sua natureza restrita de recursos. A secção 2 aborda este tópico por analisar o tipo de redes e cenários em consideração. Dada esta variedade de ambientes, uma brecha na segurança destas redes pode implicar um vazamento de informação confidencial ou prover informações sobre as escolhas e paradeiro de um largo número de indivíduos constituindo uma violação de privacidade [1]. Nessa medida um estudo extensivo de ataques dirigidos a dispositivos restritos de recursos foi conduzido na secção 3 e uma estratégia comum de mitigação apresentada na secção 4. Esta estratégia provê garantias de segurança em troca de um aumento da complexidade da infraestrutura de apoio, o que por sua vez levou à proposta do AutoStrap e de uma estação de gestão da rede. A secção 5 dá uma visão mais detalhada destes dois sistemas. Para definir com precisão

o tipo e quantidade de recursos necessários ao funcionamento de uma rede com estas garantias de segurança, conduzimos experiências de ocupação de espaço e consumo energético em dispositivos físicos, bem como testes de usabilidade ao nosso sistema. Os resultados estão expostos na secção 6. Por fim a secção 7 apresenta as nossas conclusões e oportunidades para trabalho futuro.

2 Visão Geral da Rede

Existem diversos domínios de aplicabilidade e métodos diferentes para criar redes IdC. Alguns proveem comunicação direta entre os nós da rede e a Internet, enquanto outros proveem uma *gateway* para interagir com redes externas. No nosso trabalho focamo-nos em cenários onde os nós da rede não estão diretamente ligados à Internet e necessitam de uma infraestrutura adicional de acesso e recolha de dados tal como apresentado na Figura 1.

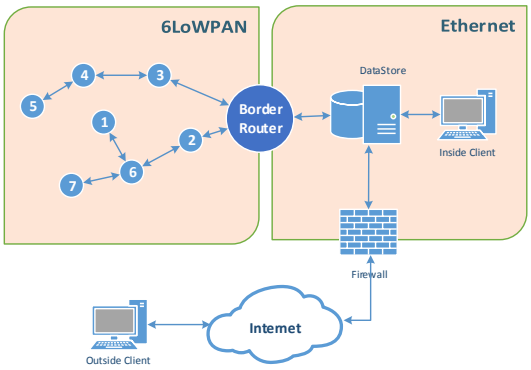

Figura 1: Visão Geral das Redes IdC Abordadas

Neste tipo de arquiteturas, os nós sensores ou atuadores pertencem a uma rede com recursos muito restritos que usa protocolos desenhados para essas necessidades e mecanismos de compressão de cabeçalhos para reduzir o tamanho dos pacotes, necessitando assim de um dispositivo de interface – o *border router* – de forma a comunicar com redes exteriores. Após atingir a rede exterior, as mensagens são processadas para converter dados dos sensores em informação útil que é depois por sua vez armazenada ou utilizada para despoletar eventos. Esta informação pode ser normalmente acedida quer por clientes dentro da rede ou através de pedidos via Internet.

Este tipo de arquiteturas pode ser usado, por exemplo, em sistemas de intrusão domésticos ou em sistemas de monitorização fabril. No cenário do sistema

de intrusão, os nós da rede colaborariam entre só para criar uma rede de sensores que propaga eventos em caso de intrusão, e a infraestrutura adicional estaria encarregue de receber esses eventos e notificar as autoridades. No cenário de monitorização fabril, os nós da rede estariam permanentemente a reportar leituras atualizadas de valores de controlo da maquinaria como por exemplo: temperatura, pressão e consumo instantâneo. A infraestrutura adicional estaria encarregue de fornecer esta informação a um painel de controlo monitorizado pelos operadores fabris. Se um atacante conseguisse desativar estes sistemas, poderia causar um encerramento de emergência da maquinaria fabril devido à falta de controlo sobre as condições de trabalho. Estas são preocupações reais apoiadas por uma gama de ataques que se foca em desativar redes IdC por colocar os nós *offline*. Tais ataques são de seguida analisados e documentados.

3 Análise de Ataques

Em ambientes IdC existe uma vasta gama de ataques que podem ser conduzidos em qualquer uma das camadas Open Systems Interconnection (OSI), desde ruído ao nível físico a ataques Denial of Service (DoS) ao nível aplicacional. No entanto, dadas as características deste dispositivos existe uma grupo de ataques ao nível da camada de rede com especial interesse e importância: ataques de esgotamento de baterias, também conhecidos como ataques "vampiro". Estes, focam-se em drenar as baterias – "vida" – dos dispositivos, trabalhando ao longo do tempo para desativar completamente a rede, daí serem chamados ataques "vampiro". Alguns destes ataques focam-se em implementações especificas enquanto outros são agnósticos ao protocolo utilizado [2][3]. Nos próximos parágrafos apresentamos alguns destes ataques em diferentes soluções de *routing* para demonstrar o seu funcionamento.

3.1 Protocolos sem Estado

Em sistemas que usam este tipo de *routing*, o nó de origem especifica todo o caminho a percorrer até ao destino no cabeçalho do pacote. Isto significa que os nós intermediários não fazem decisões em relação ao próximo nó, limitando-se a seguir as instruções dadas na origem. Usando este esquema de transmissão, um atacante pode especificar caminhos pela rede que estão longe do caminho ótimo entre dois pontos, gastando energia desnecessária no processo. O Ataque *Carousel* é um exemplo desses ataques. O seu objetivo é conduzir o pacote numa série de círculos ao longo da rede aproveitando-se das verificações limitadas aos cabeçalhos do pacote por parte dos nós intermediários. A Figura 2 mostra um exemplo em que um nó "vampiro" especificou um caminho composto por círculos ao longo da rede quando poderia ter chegado ao seu destino logo após a primeira passagem pelo nó D.

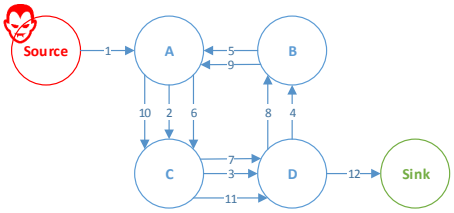

Figura 2: Ataque Carousel

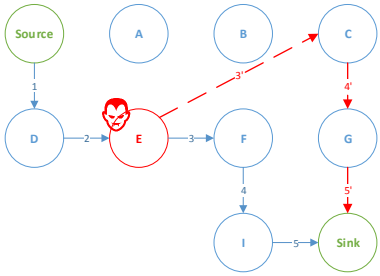

Figura 3: Ataque Antena Direcional

3.2 Protocolos com Estado

Em sistemas que usam este tipo de *routing*, os nós estão cientes da topologia da rede e são capazes de tomar decisões locais em relação à direção que um determinado pacote irá tomar. Neste caso, os atacantes podem tomar vantagem deste facto e enviar pacotes para nós da rede que não estão no caminho ótimo para o destino, ou enviar o mesmo pacote para diferentes locais da rede. O Ataque Antena Direcional é um exemplo desses ataques. Nele, o atacante toma o papel de intermediário e deposita o pacote numa localização longínqua da rede, com a ajuda, ou não, de uma antena direcionada ou até mesmo usando mais potência do que os restantes nós. Isto faz com que apenas um pacote possa ser enviado para múltiplas localizações distantes na rede e faça múltiplas travessias até atingir o seu destino. A Figura 3 mostra um exemplo em que um nó "vampiro" depositou um pacote num local distante na rede ao mesmo tempo que o enviou pelo trajeto normal obrigando o pacote a tomar dois caminhos até ao seu destino.

Embora seja verdade que existem estratégias de mitigação para estes ataques [2], essas estratégias implicam verificações e validações adicionais em cada nó e para cada pacote. Por exemplo, os nós poderiam verificar os cabeçalhos dos pacotes para detetar ciclos no trajeto da mensagem e invalidar o pacote. Ou caso vários pacotes repetidos cheguem ao destino, poder-se ia analisar o caminho feito por cada pacote e o ponto de convergência no trajeto indicaria o nó que enviou múltiplos pacotes revelando assim o atacante. Infelizmente estas estratégias de mitigação estão a colocar um fardo pesado sobre estes nós com recursos muito limitados. Para cada ataque adicional que quiséssemos mitigar mais validações teriam de ser empregues, chegando a um ponto em que os recursos despendidos em validações poder-se-iam equiparar a um ataque aos recursos desse elemento. Para resolver este problema, propomos que um *bootstrapping* seguro seja efetuado para cada novo dispositivo da rede. A secção seguinte detalha o que é um processo de *bootstrapping*, como é que ele é efetuado e algumas soluções existentes que seguem esta abordagem.

4 Bootstrapping Seguro

O termo *bootstrapping* é aplicado ao processo em que um novo dispositivo é adicionado a uma rede existente. Para atingir um *boostrapping* seguro, é necessário fornecer identificadores e/ou credenciais de segurança que permitam ao novo dispositivo identificar-se perante a rede ou uma forma para esse dispositivo obter essas credenciais. Dado que os ataques apresentados são executados por intrusos capazes de interagir com a rede a partir de um nó comprometido, se conseguíssemos garantir um *bootstrapping* seguro, na medida em que cada novo nó seria autenticado antes de se tornar um membro ativo da rede, então esses ataques já não poderiam ser executados. Métodos de *bootstrapping* seguro e sistemas de admissão já foram previamente propostos. No entanto, o desenvolvimento e otimização de protocolos ao nível aplicacional bem como novos esquemas de *routing* permitem novas abordagens mais focadas na natureza dos dispositivos usados na IdC. Bergman et al.[4] propôs uma técnica tri-faseada de *bootstrapping* seguro para nós de uma rede que usa o protocolo Constrained Application Protocol (CoAP). No entanto a distribuição da chave de rede usada para cifrar as comunicações daí em diante pode ser intercetada e com isso um atacante pode inserir os seus próprios nós maliciosos na rede. Os autores propõem reduzir a potência do radio emissor durante esta fase mas não é certo que isso seja impedimento para um atacante com elevados recursos de observação. Oliveira et al. [5] propôs uma solução de controlo de admissão para redes IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) baseado em aprovação administrativa. Cada novo dispositivo manifestaria a sua presença através da rede, e um administrador recebendo essa notificação daria acesso ou não ao dispositivo com base no seu identificador. Esta solução embora não necessite de hardware ou infraestruturas adicionais também não consegue garantir um processo seguro pois podem ser efetuados ataques de *man-in-the-middle* comprometendo a aprovação de novos dispositivos. Para além destes esforços de trabalho relacionado, existem técnicas adicionais de *bootstrapping* seguro [6] que recorrem a *tokens* e passwords de uso único para cifrar as primeiras comunicações de um nó na rede e permitir a receção de credenciais adicionais. Existem também abordagens em que os fabricantes de *hardware* carregam previamente os dispositivos com as credenciais de segurança necessárias para a fase operacional durante a fase de fabrico. No entanto estas abordagens necessitam de *hardware* adicional ou confiança nas credenciais inseridas pelos fabricantes. No nosso trabalho propomos um sistema de *bootstrapping* seguro – AutoStrap – onde:

- Não é necessário hardware adicional durante a instalação no terreno;
- Não é necessário obter credenciais adicionais após instalação no terreno;
- Não é necessário recorrer a terceiros para gerar ou instalar credenciais;
- Todas as mensagens enviadas na rede são cifradas e autenticadas desde o primeiro momento

Chamámos à nossa solução AutoStrap porque se destina a permitir um processo de *bootstrapping* seguro e eficiente, que tenha o mínimo de interação necessária com os operadores do sistema e não necessite de conhecimento interno

do sistema para ser usado. Na secção seguinte, a nossa infraestrutura proposta é analisada em termos de requisitos e objetivos, arquitetura e detalhes de implementação.

5 Infraestrutura Proposta

De forma a demonstrar as capacidades e domínio de aplicabilidade da nossa infraestrutura e sistema de *bootstrapping*, iremos usar um cenário de Campus Universitário Inteligente. Nos parágrafos seguintes serão aplicadas as informações recolhidas em termos de ataques e estratégias de mitigação com o intuito de selecionar protocolos de comunicação adequados e definir os objetivos e requerimentos do sistema.

5.1 Objetivos e Requerimentos

Uma necessidade premente entre aplicações IdC é que as suas comunicações sejam eficientes, neste sentido é nosso objetivo que toda a rede consuma o mínimo de recursos possível. Além disso, é necessário que o sistema seja utilizado por operadores sem conhecimento do seu funcionamento interno e que o processo de *bootstrapping* seja automatizado e eficiente. É ainda necessário garantir um modelo de comunicação seguro, pelo que é critico que as pacotes propagados na rede tenham garantias de confidencialidade, integridade e autenticidade.

5.2 Stack de Protocolos

Uma análise detalhada das diversas alternativas existentes em termos de protocolos para ambientes IdC [7] foi conduzida para revelar os pontos fortes e fracos de cada candidato. É de seguida apresentado um resumo dos protocolos selecionados para fazem parte do *stack* do modelo de comunicação.

5.2.1 Camada Física e Data-Link Dado que a maioria dos dispositivos IdC necessitam de rádios *wireless*, estes devem visar simplicidade e baixos consumos. Uma vez que o foco na cobertura de grandes distâncias faz o Wi-Fi[8] ter valores elevados de consumo energético, o protocolo o protocolo IEEE 802.15.4 [9] foi selecionado devido às suas especificações de baixo consumo energético, baixa taxa de transmissão de dados e alto rendimento de propagação de pacotes.

5.2.2 Camada de Rede A visão da IdC e a sua massificação só podem ser atingida com o uso do IPv6 [10] devido à necessidade de muitos mais endereços IP. Dadas as limitações no tamanho dos pacotes das redes IEEE 802.15.4, um grupo de trabalho da Internet Engineering Task Force (IETF) desenvolveu o mecanismo de compressão de cabeçalhos 6LoWPAN [11] que permite remover uma larga porção dos *overheads* encontrados no uso de pacotes IPv6 [12], tornando assim possível o uso deste esquema de endereçamento. Existindo a possibilidade de

alterações frequentes na topologia de rede associadas à instabilidade dos rádios, o protocolo Routing Protocol for Low-Power and Lossy Networks (RPL) [13] foi escolhido devido à sua capacidade de criação e reparação eficiente da topologia de rede em caso de falha ou adição de novos elementos.

5.2.3 Camada Aplicacional O CoAP [14] é um protocolo baseado na arquitetura REpresentational State Transfer (REST) com o objetivo de permitir interações entre clientes e servidores através de pedidos encontrados no protocolo HTTP tais como *Get*, *Post*, *Put* e *Delete*. Uma vez que usa User Datagram Protocol (UDP) em vez de Transmission Control Protocol (TCP) não adiciona os fardos das garantias de retransmissão e controlo de fluxo providenciadas pelo TCP. Apesar de ser um protocolo leve, o CoAP providencia importantes funcionalidades, tais como:

- Observação de Recursos - Monitorização de recursos através de um mecanismo *publish/subscribe*;
- Descoberta de Recursos - Os servidores CoAP mantêm uma lista dos seus recursos usando Universal Resource Identifiers (URIs) bem conhecidos pelos clientes descrevendo o seu tipo e objetivo;
- Interoperabilidade - Dado que o CoAP é baseado na arquitetura REST, um simples *proximal* é capaz de mapear pedidos CoAP a pedidos Hypertext Transfer Protocol (HTTP).

5.3 AutoStrap

O principio de funcionamento do AutoStrap é a adição de um novo componente na arquitetura da infraestrutura – o *bootstrapper* – que é responsável pela escrita para o dispositivo de todos os identificadores e credenciais de segurança necessários para uma operação que responda aos objetivos e requerimentos do sistema, sem necessidade de obter credenciais após o inicio da fase de operação ou de confiar em credenciais criadas por terceiros. As credenciais utilizadas são um identificador único e uma chave de 128bits usada pelo protocolo Advanced Encryption Standard (AES) [15] no modo Counter with CBC-MAC (CCM) estrela [16]. O uso desta chave e protocolo permite que os pacotes tenham o seu conteúdo cifrado e o seu cabeçalho autenticado com um Message Integrity Code (MIC) também gerado a partir dessa mesma chave. Isto assegura que todos os pacotes que se propagam na rede são confidenciais, íntegros e autenticos. Desta forma, os "vampiros" não serão capazes de se introduzir na topologia, frustrando assim as suas tentativas de conduzir ataques de esgotamento de bateria. Este processo é iniciado por um operador do sistema, com a ligação do dispositivo ao bootstrapper, e após selecionar o tipo de hardware numa interface gráfica embutida, todo o restante processo é realizado automaticamente em segundo plano tal como apresentado no diagrama de sequência presente na Figura 4.

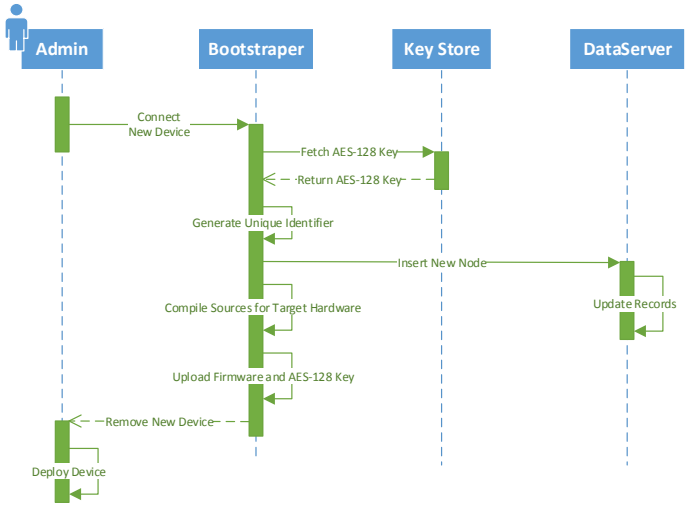

Figura 4: Processo de Bootstrapping

5.4 Arquitetura do Sistema

A arquitetura do nosso sistema assenta em parte nos modelos apresentados na secção 2, onde redes 6LoWPAN com recursos limitados comunicam com redes externas através de um *border router*. Para melhor visualizar e entender as potencialidades do sistema numa rede de maior dimensão, a Figura 5 apresenta um cenário de um Campus Inteligente onde cada pavilhão tema sua própria rede 6LoWPAN e reporta dados a um sistema central através de diversos *border routers*. É também verificada a possibilidade de obter informação do sistema através de pedidos dentro da rede ou fora dela pela Internet.

Em relação à arquitetura do sistema central de gestão, visível em maior detalhe na Figura 6, os componentes *Proxy* e *Data Server*, responsáveis por mapear pedidos vindos de fora da rede e por armazenar leituras e eventos vindos da rede com recursos limitados, já seria expectável serem encontrados em outros sistemas. A inovação contida neste proposta advém dos outros dois componentes do sistema, o *CoAP Client Observer*, e o *Bootstrapper* que faz por sua vez uso de um *Key Store*. O *KeyStore* é responsável por guardar a chave de rede utilizada no processo de *bootstrapping*. O *Bootstrapper* é responsável por conduzir o processo AutoStrap mapeando o novo dispositivo no sistema e fornecendo-lhe as credenciais de segurança necessárias. Por fim o *CoAP Client Observer* atua como o único cliente da rede. Em vez de termos utilizadores a requisitar novas leituras diretamente aos sensores da rede com recursos limitados, este cliente regista-se junto dos dispositivos existentes e é notificado de cada novo valor ou evento despoletado pelos dispositivos. Embora fosse possível ter vários utiliza-

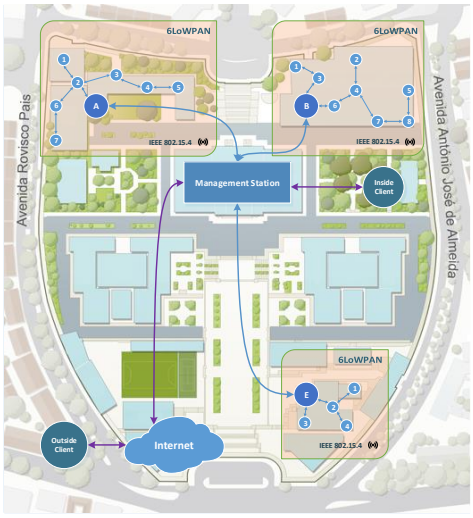

Figura 5: Arquitetura Global

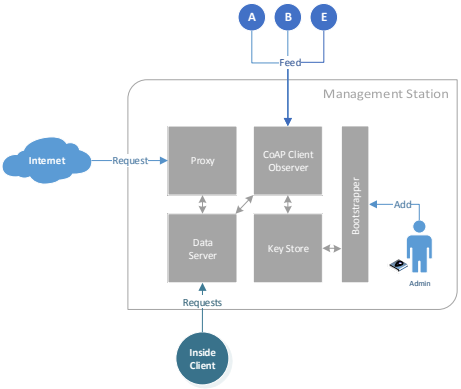

Figura 6: Arquitetura Sistema Central

dores a requisitar diretamente estes valores, isso implicaria um gasto adicional de recursos para processar todos esses pedidos. Com a utilização de um único cliente, apenas uma mensagem necessita de atravessar a rede para cada novo evento, sendo esse evento guardado no *Data Server* para consulta por parte dos utilizadores.

6 Avaliação

A nossa avaliação divide-se em duas componentes. Primeiro, medimos e documentamos os recursos físicos necessários para suportar os protocolos e estratégias de mitigação escolhidas de forma a entender se são adequados ao hardware usado na IdC. Segundo, conduzimos testes de usabilidade ao nosso sistema de gestão de redes para analisar a sua eficiência e facilidade de uso.

6.1 Aplicabilidade ao Hardware

De forma a avaliar o espaço real ocupado pelos protocolos e estratégias de mitigação usados, foi utilizada a ferramenta *msp430-size*[1] que mede o tamanho do firmware inserido nos dispositivos da rede. Esta ferramenta permite obter separadamente a quantidade de memoria flash e memória Random-Access Memory (RAM) ocupada. Os resultados da análise feita, tanto a um firmware base sem mecanismos de mitigação, como a um firmware que introduz segurança nas

[1] http://www.ti.com/tool/msp430-gcc-opensource

comunicações são apresentados na Tabela 1. Analisando os resultados podemos concluir que a introdução destes mecanismos representa um aumento de 3.02% na utilização de memória flash e 1.02% na utilização de memória RAM.

Tabela 1: Memória Utilizada

Segurança	Flash(KB)	RAM(KB)
No-Sec	59.56	13.54
LLSec	61.36	13.80

Tabela 2: Memória Existente

Dispositivo	Flash(KB)	RAM(KB)
Zolertia RE-Mote	512	32
Arago Wismote	128	16
TI CC2538DK	512	32

Para entender se o tamanho de firmware usado pelo nosso sistema é adequado aos dispositivos da IdC foram analisadas três placas de desenvolvimento (Zolertia RE-Mote[2], Arago Systems Wismote[3] and the Texas Instruments CC2538DK[4]) e as suas respetivas características sumarizadas na Tabela 2, sendo possível concluir que dado os tamanhos de firmware observados serem compatíveis com o hardware analisado, e este hardware ser por usa vez utilizado em aplicações IdC, a nossa solução é aplicável às redes IdC.

6.2 Consumo Energético

Embora seja necessário o uso de mecanismos adicionais para garantir a segurança da informação nas redes IdC, caso estes mecanismos obriguem a um elevado consumo energético não poderão ser empregues pois atuarão como atacantes no sentido em que drenarão as baterias dos dispositivos. Desta forma, uma análise ao consumo energético, usando hardware real[5] foi efetuada com o cuidado de observar tanto momentos de atividade como momentos de silêncio rádio. A partir dos resultados obtidos foi calculado o consumo energético sobre a forma de Potência(W) a partir da fórmula $P = IV$, onde I é a corrente em Amperes(A) e V a voltagem em Volts(V) e os resultados sumarizados na Tabela 3. Sabendo que o nosso firmware emprega um protocolo de Radio Duty Cycling (RDC) capaz de manter o rádio desligado durante cerca de 99% do tempo de utilização [17], os dados obtidos validam que a nossa solução mantém um baixo consumo energético podendo assim ser utilizada em ambientes IdC.

6.3 Processo de Bootstrapping

Tendo em conta a necessidade de que o processo de bootstrapping seja rápido e acessível a funcionários não familiarizados com o funcionamento interno do

[2]http://zolertia.io/product/hardware/re-mote

[3]http://www.wismote.com

[4]http://http://www.ti.com/tool/cc2538dk

[5]Foi utilizada a placa Zolertia RE-Mote dado o seu foco na redução de consumos e utilização de um criptoprocessador integrado.

Tabela 3: Consumos Energéticos

Modo	V	mA	W
Radio ON	9.0	17	0.15
Radio OFF	9.0	5.2	0.05

Tabela 4: Tempos de Bootstrapping

Operação	Tempo(s)
Inserir Novo Dispositivo	2
Abrir Interface	5
Selecionar Hardware	4
Compilar Código Fonte	13
Eliminar Firmware Existente	5
Upload Novo Firmware	3
Remover Novo Dispositivo	2

sistema, conduzimos experiências que permitiram observar o tempo e passos necessários para completar um processo de bootstrapping como medidas da sua usabilidade. A lista de passos e tempo observado é apresentada na Tabela 4. Dado que para além da inserção e remoção física do dispositivo, tudo o que o operador necessita de indicar é o tipo de hardware utilizado é assim cumprido o requisito de não existir necessidade de conhecimento adicional. Embora o tempo total aparente de bootstrapping seja 34 segundos, a sua eficiência é especialmente importante num processo em série. Nesse caso, após a primeira sequência todo o processo de abertura da interface, seleção de hardware e compilação já está realizado, representando um ganho de 22 segundos resultando num tempo real de 12 segundos. Desta forma podemos concluir que a nossa solução é prática e adequada aos cenários IdC.

7 Conclusões

Dadas as limitações dos dispositivos utilizados na redes IdC, obter comunicações seguras não é uma tarefa fácil. Para atingir este objetivo realizámos uma análise extensiva aos protocolos e estratégias de mitigação de ataques "vampiros" existentes estabelecendo uma *stack* de protocolos que usámos nos dispositivos com recursos limitados. Experiências em *hardware* real confirmam que o *stack* encontrado tem tanto um tamanho como um consumo energético compatível com os dispositivos IdC. Face às exigências adicionais em termos de infraestrutura que o nosso modelo de comunicação necessita, propusemos uma arquitetura central que vai de encontro às necessidades correntes dos ambientes IdC abordados e emprega mecanismos ao nível aplicacional para reduzir o número de pacotes que fluem na rede. Esta infraestrutura possui também os componentes adicionais de baixo impacto em termos de recursos necessários que permitem a execução do processo de *bootstrapping* seguro proposto – AutoStrap. Processo este automático, que não necessita de conhecimento interno por parte do operador e com métricas de usabilidade que o tornam adequado aos ambientes IdC.

Como trabalho futuro, salientamos a necessidade de proteção da memória dos dispositivos para impedir o roubo de credenciais de segurança e consequente clonagem dos dispositivos. Deixamos como sugestão o uso de circuitos integrados

com mecanismos de impedimento de leitura ou o bloqueio via software de certas regiões de memória dos dispositivos onde residem as chaves de rede.

Referências

1. Ukil, A., Bandyopadhyay, S., Pal, A.: Privacy for IoT: Involuntary privacy enablement for smart energy systems. 2015 IEEE International Conference on Communications (ICC) (2015) 536–541
2. Vasserman, E.Y., Hopper, N.: Vampire attacks: Draining life from wireless ad Hoc sensor networks. IEEE Transactions on Mobile Computing **12**(2) (2013) 318–332
3. Pongle, P., Chavan, G.: A survey: Attacks on RPL and 6LoWPAN in IoT. 2015 International Conference on Pervasive Computing (ICPC) **00**(c) (2015) 1–6
4. Bergmann, O., Gerdes, S., Schafer, S., Junge, F., Bormann, C.: Secure bootstrapping of nodes in a CoAP network. 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW) (2012) 220–225
5. Oliveira, L.M., Rodrigues, J.J., Neto, C., de Sousa, A.F.: Network Admission Control Solution for 6LoWPAN Networks. Proceedings of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (2013) 472–477
6. Fischer, K., Geßner, J., Fries, S.: Secure Identifiers and Initial Credential Bootstrapping for IoT@Work. 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (2012) 781–786
7. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Communications Surveys & Tutorials **PP**(99) (2015) 1–1
8. IEEE: IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Volume 2012. (2012)
9. IEEE Computer Society: Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). Volume 2011. (2011)
10. Pickard, J., Patrick, A.Y., Robinson, A.: Analysis of enterprise IPv6 readiness. SoutheastCon 2015 (2015) 1–7
11. Shelby, Z., Chakrabarti, S., Nordmark, E., Systems, C., Bormann, C., Ericsson: Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). https://tools.ietf.org/html/rfc6775 (2012)
12. Hui, J., Culler, D.: Extending IP to low-power, wireless personal area networks. IEEE Internet Computing **12**(4) (2008) 37–45
13. Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., Alexander, R.: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. https://tools.ietf.org/html/rfc6775 (2012)
14. Shelby, Z., Hartke, K., Bormann, C.: The Constrained Application Protocol (CoAP). https://tools.ietf.org/html/rfc7252 (2014)
15. Fips, N.: 197: Announcing the advanced encryption standard (AES). ... Technology Laboratory, National Institute of Standards ... **2009** (2001) 8–12
16. Corp, C.: Formal Specification of the CCM * Mode of Operation René Struik Voice : Fax : Re : Abstract as well as some (informational) design rationale . This document is an edited Purpose Notice. (2005) 1–19
17. Dunkels, A.: The ContikiMAC Radio Duty Cycling Protocol. SICS Technical Report T2011:13 , ISSN 1100-3154 (2011) 1–11