https://eventos.utfpr.edu.br//sicite/sicite2020

MQTTree: modelo de comando e controle de *botnet* com uso de *brokers* MQTT abertos

MQTTree: botnet command and control model using open MQTT brokers

Franco Barpp Gomes
francog@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná - Câmpus Curitiba, Curitiba, Paraná, Brasil

Daniel Fernando Pigatto
pigatto@utfpr.edu.br
Universidade Tecnológica Federal do Paraná - Câmpus Curitiba, Curitiba, Paraná, Brasil

RESUMO

A Internet das Coisas (IoT) vem ficando cada vez mais presente no cotidiano global. Porém, nesse meio composto de muitos dispositivos e redes que exigem leveza e rapidez de comunicação, a segurança acaba, muitas vezes, ficando em segundo plano. Uma das formas de se explorar a área ficou bastante evidente após o advento da *botnet* Mirai, que invadiu principalmente dispositivos IoT simples para construir uma rede com poder extremamente alto de ataques DDoS (*Distributed Denial-of-Service*). Nesse sentido, o presente trabalho busca alertar sobre a possibilidade de se ir além – não só usar o meio IoT como fonte de *bots*, mas usá-lo também como fonte de infraestrutura para comando e controle, atividade que normalmente envolve considerável investimento. Com o modelo proposto, seria possível, através do uso de numerosos *brokers* MQTT abertos como *proxies*, controlar uma grande *botnet* com pouco ou nenhum recurso de infraestrutura própria, ainda obtendo alta portabilidade e anonimidade.

PALAVRAS-CHAVE: Redes de computadores. Internet das coisas. Computadores - Medidas de segurança.

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

ABSTRACT

The Internet of Things (IoT) is continuously becoming more embedded in our day-to-day lives. However, in this environment composed of a variety of devices and networks that require communication lightweightness and speed, the security often ends up being underestimated. One of the ways to exploit this environment became clear after the advent of the botnet Mirai, which invaded mainly simple IoT devices to establish a network with extremely high DDoS (Distributed Denial-of-Service) attack power. In this sense, this research aims to alert about the possibility of going beyond – not just using the IoT environment as a source of bots, also using it as a source of C&C (Command and Control) infrastructure, which normally involves significant investment. With the proposed model, through the use of numerous open MQTT brokers, it would be possible to control a large botnet with little to no self-mantained infrastructure while still obtaining high portability and anonimity.

KEYWORDS: Computer networks. Internet of things. Computer security.

INTRODUÇÃO

Existem diversas tecnologias que foram criadas e popularizadas com o advento da Internet das Coisas (IoT). Uma delas foi o protocolo MQTT (*MQ Telemetry Transport*), um protocolo de arquitetura *publish-subscribe* (LIGHT, 2017). Ele é notável por sua portabilidade – comum em casos de uso como monitoramento de sensores.

O funcionamento desse protocolo depende da existência de um *broker*, que organiza o fluxo de informação entre tópicos e, assim, redireciona as mensagens dos publicadores (*publishers*) do tópico aos seus respectivos assinantes (*subscribers*). Por questões de praticidade, para facilitar esse processo de envio e recebimento de dados, muitos *brokers* são configurados sem senha. Na página Shodan, uma pesquisa de dispositivos por "MQTT Connection Code: 0", que significa que a conexão MQTT seria bem-sucedida sem quaisquer credenciais, retorna cerca de 73 mil endereços IP de *brokers* (SHODAN, 2020).

O presente trabalho busca alertar e propor um modelo que mostra que, mesmo sem de fato invadir esses distribuidores de informação, ainda é possível usar seus recursos em uma *botnet* para estabelecer uma infraestrutura de comando e controle. Considerando que o custo de infraestrutura de comando e controle potencialmente desestimula a criação de *botnets* numerosas, a possibilidade de manter uma *botnet* com custo grandemente menor é preocupante.

MATERIAL E MÉTODOS

A partir da idealização da proposta, foi utilizado o método de pesquisa exploratória para, através da leitura de trabalhos científicos ligados à área de Arquitetura de Redes, avaliar sua viabilidade de aplicação.

Em seguida, foram estudados casos existentes de *botnets* e outros trabalhos teóricos da área, de forma a levar esse conhecimento em conta para o estabelecimento de uma proposta mais concreta sobre o modelo de comando e controle de *botnet* concebido. Esse processo levou a adaptações no modelo inicialmente idealizado, trazendo melhorias em pontos como flexibilidade e eficiência.

A PROPOSTA

Os objetivos do modelo de comando e controle de *botnet* proposto são os seguintes:

a) explorar a existência de uma grande quantidade de *brokers* MQTT abertamente acessíveis;

b) manter a perspectiva de custo de operação baixa;

c) ser compatível com dispositivos IoT simples, isto é, ter alta portabilidade.

O primeiro objetivo é, de fato, o princípio mais importante desde projeto. Porém, b) e c) são também relevantes, já que um *design* simples e barato são pontos que favoreceriam uma aplicação real – o que busca-se evitar.

BROKERS COMO *PROXIES*

A primeira meta da proposta, isto é, a de usar *brokers* MQTT abertamente acessíveis, é o ponto inicial; esses tratam-se de recursos pouco vistos e altamente disponíveis. Também é, de certa forma, uma perspectiva diferente da normal – não é necessário invadir esses *brokers* para conseguir uma parte de seu poder computacional.

Isso fica claro quando considerado que esses *brokers* não precisam ser partes ativas da rede, eles podem ser apenas *proxies* de comandos. Além disso, considerando sua característica *publish-subscribe*, eles têm a capacidade serem usados para não só diretamente repassar, mas também multiplicar o tráfego que redirecionam (EUGSTER, 2003).

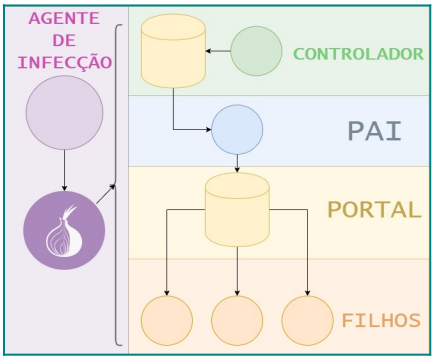
Essa capacidade poderia ser especialmente explorada levando em conta a base de uma topologia de *botnet* específica: a topologia hierárquica (OLLMANN. 2009). Nessa topologia, os dispositivos da rede são organizados hierarquicamente, de forma que cada *bot* se comunica com múltiplos outros, esses pertencentes a um nível hierárquico inferior ao primeiro.

CLASSES DE NÓS

São, no total, 5 tipos de nós: **Portal**, **Pai**, **Filho**, **Controlador** e o **Agente de Infecção**. O Portal é um servidor MQTT vulnerável encontrado pela rede, enquanto os Pais e Filhos são dispositivos genéricos. O Agente de Infecção é um servidor externo que se comunica de forma anônima com o Controlador; ele infecta alvos e gerencia a árvore de dispositivos. O Controlador, por sua vez, é quem efetivamente comanda a rede.

Essas classes são mostradas na Figura 1, em uma estrutura claramente de base hierárquica.

Figura 1 – Classes de nós



Fonte: Autoria própria.

FUNCIONAMENTO

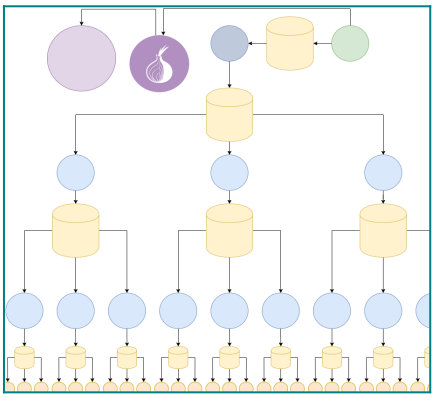
Dentro de um tópico de um Portal, um Pai se comunica com múltiplos Filhos, ordenando certas tarefas preestabelecidas. Quem ordena ações do Pai de maior ordem é o Controlador, da mesma forma que um Pai se comunica com seus Filhos, mas por um servidor MQTT próprio. Ou seja, os Pais atuam como *bots* de

forma normal, mas também redirecionam seus comandos recebidos ao seu Portal, que então redireciona-os aos respectivos Filhos.

O Agente de Infecção somente gerencia a rede e seu estado após a infecção, corrigindo alguns problemas se necessário, usando a rede Tor como camada de anonimidade (MCCOY, 2008). Como seu contato não é tão intenso, a latência característica da rede Tor não se torna um grande problema.

Considerando que o Pai de maior ordem poderá ter múltiplos Filhos que, por sua vez, também podem ser Pais. A rede da Figura 1, então, pode ser escalada para a rede representada na Figura 2.

Figura 2 – Topologia ampliada



Fonte: Autoria própria.

Note-se que cada portal se comunica com 3 nós Filhos com o objetivo de facilitar a visualização: na prática, poderiam ser dezenas. Assim, ficam evidentes as proporções que podem ser obtidas com somente alguns *brokers* MQTT e um servidor de comando simples.

ANONIMIZAÇÃO DO CONTROLADOR

Assume-se que o objetivo seja encontrar o Controlador de uma rede dessa espécie que esteja desempenhando um ataque DDoS (*Distributed Denial-of-Service*) (KOLIAS, 2017). Encontrar um nó ativo que esteja ativamente participando do ataque, de fato, não é nada difícil. Como encontrar o Controlador é efetivamente encontrar o maior Pai, o processo é encontrar um nó qualquer e, assumindo-o como Filho ou controlado, buscar seu Pai, recursivamente, até chegar ao topo da árvore, o Controlador.

Porém, o primeiro detalhe: em uma probabilidade uniforme, dentre todos os dispositivos atacantes, seria muito mais fácil encontrar um nó Filho de alta ordem, pois o número desses é muito maior que os de pequena ordem; visível como as camadas de uma pirâmide. Ou seja, provavelmente será um grande caminho pela rede até que se chegue ao Pai controlado. Isso não seria, no entanto, um problema, se obter o Pai de um Filho fosse simples.

Obter o Pai de um Filho, por sua vez, definitivamente não é simples. O principal problema disso é, na verdade, que quem se comunica diretamente com os Filhos não é o Pai: é o *broker* MQTT, o Portal. Com isso, o primeiro passo para

esse fim seria interceptar a comunicação entre o servidor MQTT e o Filho, descobrindo o Portal.

Em seguida, o problema é ainda mais complexo. Como, no protocolo MQTT, os *subscribers* (Filhos) não sabem quem enviou determinada mensagem, seria necessário obter o relatório do servidor para descobrir o *publisher*. Esse tipo de informação não é fácil de conseguir sem a ajuda direta dos gerentes da rede, que pode, realmente, demorar dias dentre contato e autorização.

Ou seja, o esforço de obter somente um Pai já é bem considerável. Achar o Controlador é fazer isso recursivamente, de um ponto que provavelmente é, em uma rede suficientemente grande, algumas dezenas de Pais de distância do Controlador. É uma atividade possível, mas que demanda grande esforço. Para ainda assim mitigar seus riscos para rede, pode ser obtida mais flexibilidade na organização da rede por meio do Agente de Infecção.

ORGANIZAÇÃO DINÂMICA DA REDE

Como o Agente de Infecção tem controle sobre a organização dos nós da rede em sua plenitude, pode-se facilmente fazer com que ele consiga editar funções de cada dispositivo e, assim, os níveis de hierarquia da rede. Isso é simples em termos de implementação, mesmo que seja um processo provavelmente demorado em execução: se trata somente de mudar canais de comunicação e informações sobre nós Pais e Filhos de cada nó, tratando o sistema todo como uma árvore.

Um dos pontos é que, por fazer isso, ele pode mudar, até mesmo, quem é o nó controlado diretamente pelo Controlador. Transitando esse nó específico entre um conjunto de nós, provavelmente os nós Pais de maior ordem, o acesso ao Controlador fica mais difícil ainda. Para maior segurança, esse conjunto de nós poderia ser de posse do *botmaster*, considerando que sua função é mais crítica em comparação às dos outros nós.

Além disso, seria possível mudar a organização geral da rede a cada quantidade pré-determinada de tempo ou após cada ataque. Adicionalmente, isso poderia ser feito de forma a manter as camadas superiores intactas ou como um setor de randomização separado, mitigando o risco de que um nó acabe tendo uma mudança muito substancial de ordem como Pai na rede. Essa mudança substancial seria prejudicial já que tornaria relativamente viável a estratégia de periodicamente monitorar alguns nós e esperar que, por sorte, esses acabem próximos do Controlador.

Outras mudanças possíveis seriam, por exemplo, a mudança somente do Portal em que os Filhos e o Pai se encontram, ou até a remoção de uma sub-árvore completa se ela representar algum perigo à rede. Tudo isso seria facilmente gerenciável pelo Controlador.

REMANEJAMENTO DE NÓS COM PORTAIS INATIVOS

Uma outra possibilidade são Portais pararem de funcionar. Isso seria potencialmente devido ao administrador do *broker* ter percebido a infecção e desconectado os dispositivos. Isso é problemático no sentido que,

essencialmente, o Controlador não tem como receber uma mensagem direta de um nó específico, pois nem ele nem o nó se conhecem.

A solução, nesse caso, seria que o Agente de Infecção periodicamente verificasse a disponibilidade dos servidores MQTT e manualmente remanejasse Pais e Filhos com Portais inativos para um novo Portal. Isso poderia ser feito usando um nó aleatório da rede como intermediário, para então verificar a existência e classificar o canal de comunicação do *broker* MQTT.

O Agente de Infecção pode anonimamente testar o funcionamento de uma rede MQTT com um dispositivo qualquer da rede. Fazendo o processo através da rede Tor, essa comunicação seria feita de forma anônima; não haveria desvantagem alguma na anonimidade do Agente de Infecção no processo.

RESULTADOS E DISCUSSÃO

O presente trabalho faz tão somente a condução de uma proposta; não há aplicação prática concreta no momento. Por esse motivo, não há resultados de utilização a serem apresentados. Porém, os resultados da pesquisa podem ser discutidos em meios teóricos.

A ideia, em si, é algo que aparentemente nunca foi discutido: uma *botnet* de topologia hierárquica com servidores MQTT intermediários entre cada camada. A topologia hierárquica tradicional é usada normalmente de forma que dispositivos infectados possam agir como *proxies* diretos de forma não especializada. Isso, além de limitar os dispositivos que atuam como ser *proxies*, estabelece uma conexão direta entre esses dispositivos vulneráveis e seus comandados, o que é negativo para a anonimidade da rede.

Além disso, os resultados potencialmente atingíveis sobre a dificuldade de rastreamento do Controlador pelo uso desses servidores intermediários mostram uma forma de obter boa anonimidade sem necessariamente usar ferramentas mais complexas e limitadas.

CONCLUSÃO

É, de qualquer forma, uma grande possibilidade que ainda existam falhas não resolvidas no modelo apresentado e que seriam descobertas mais evidentemente em uma aplicação funcional desse. Alguns pontos mais problemáticos foram tratados especificamente, mostrando possíveis soluções que poderiam se aplicar ou não dependendo de usos de caso específicos. Mas, acima disso, o modelo tem visíveis inovações que, mesmo se não forem aplicadas diretamente pelas formas aqui citadas, podem inspirar novos modelos.

Finalmente, se torne claro que o presente trabalho não tem como objetivo incentivar atos maliciosos; somente busca alertar para possíveis novas ideias no contexto de comando e controle de *botnets* e, assim, incentivar a busca efetiva de formas de impedir o desenvolvimento dessas estratégias antes que elas sejam realmente praticadas.

AGRADECIMENTOS

Ao Prof. Dr. Daniel Fernando Pigatto e à Profa. Dra. Ana Cristina Kochem Vendramin, que desempenharam o papel de, respectivamente, orientador e coorientadora do projeto.

REFERÊNCIAS

EUGSTER, Patrick T. et al. The many faces of publish/subscribe. **ACM computing surveys (CSUR)**, v. 35, n. 2, p. 114-131, 2003.

KOLIAS, Constantinos et al. DDoS in the IoT: Mirai and other botnets. **Computer**, v. 50, n. 7, p. 80-84, 2017.

LIGHT, Roger A. Mosquitto: server and client implementation of the MQTT protocol. **Journal of Open Source Software**, v. 2, n. 13, p. 265, 2017.

MCCOY, Damon et al. Shining light in dark places: Understanding the Tor network. In: **International symposium on privacy enhancing technologies symposium**. Springer, Berlin, Heidelberg, 2008. p. 63-76.

OLLMANN, Gunter. Botnet communication topologies. **Retrieved September**, v. 30, p. 2009, 2009.

SHODAN. Shodan, 2020. Disponível em: https://www.shodan.io. Acesso em: 28 ago. 2020.