SEGURANÇA OFENSIVA NA EXPLORAÇÃO DE VULNERABILIDADES EM DISPOSITIVOS PARA INTERNET DAS COISAS ATRAVÉS DE TÉCNICAS DE COMANDO E CONTROLE

Palavras-Chave: DISPOSITIVOS IOT, FIRMWARE, C2

Autores:

NATALIA EMBOAVA, FT – UNICAMP

Prof. Dr. ANDRÉ LEON SAMPAIO GRADVOHL (orientador), FT - UNICAMP

INTRODUÇÃO:

Nos últimos anos, o crescente número de dispositivos com baixo poder computacional e capacidade para captar dados do ambiente através de sensores trouxe um novo significado para o conceito de Internet das Coisas (IoT), além da simples transmissão e recepção de dados de objetos conectados a uma rede [1]. Com esse crescimento, a segurança desses dispositivos passou por diversas mudanças e deve ser constantemente atualizada e revisada, já que a quantidade de possíveis vetores de ataque também aumentou.

Os Firmwares, que são *softwares* essenciais para o funcionamento de dispositivos IoT, estabelecem a comunicação entre o *hardware* e o *software* desses dispositivos e são comumente armazenados na memória do dispositivo. Eles são vitais para seu funcionamento. Através desses componentes, é possível extrair informações sobre o funcionamento interno de um dispositivo ou de um ecossistema IoT, o que pode ser explorado em potenciais ataques [2]. Um dos principais objetivos da análise de *firmwares* [3] é a aplicação da engenharia reversa em arquivos binários, o que pode revelar uma grande quantidade de informações importantes e dados sensíveis relacionados à segurança do dispositivo.

Por outro lado, o conceito de comando e controle (C2) é utilizado em ataques para a comunicação com dispositivos que foram comprometidos. Esse tipo de ataque consiste no uso de um servidor que envia comandos e recebe dados dos dispositivos comprometidos [4]. Um servidor C2 é utilizado para que os atacantes possam controlar remotamente os dispositivos comprometidos.

Este projeto visa à exploração de vulnerabilidades nos sistemas de dispositivos IoT com base em conhecimentos de segurança ofensiva. Inicialmente, os *firmwares* desses dispositivos serão analisados para obter informações e partes essenciais para o funcionamento do dispositivo. Em seguida, com o auxílio da ferramenta *Metasploit* [5], será criada uma conexão persistente com o dispositivo alvo para fins de comando e controle, baseando-se no *framework MITRE ATT&CK* [6][7]. Por fim, realizamos a exploração de possíveis vulnerabilidades presentes no sistema desse dispositivo.

METODOLOGIA:

A pesquisa consistiu na utilização da técnica de “Software de Acesso Remoto” [8] do *MITRE ATT&CK* como base para a exploração de vulnerabilidades em roteadores. Inicialmente, para simular um ambiente de teste controlado, utilizamos uma máquina virtual, *VMWare Workstation Player 17*, com o sistema operacional Ubuntu 22.04, como máquina hospedeira. Esse ambiente permitiu emular o roteador, criar o servidor C2 e realizar os testes subsequentes. O principal componente utilizado para as emulações dos dispositivos foram *firmwares*. Especificamente, utilizamos o sistema de arquivos contido nestes.

A Figura 1 ilustra o processo que realizamos. A primeira etapa do processo envolveu a extração do sistema de arquivos a partir do *firmware* do roteador. Este sistema de arquivos foi então montado na máquina hospedeira para criar um ambiente de emulação. Em seguida, configuramos um servidor C2 na máquina hospedeira para controlar remotamente o roteador emulado.

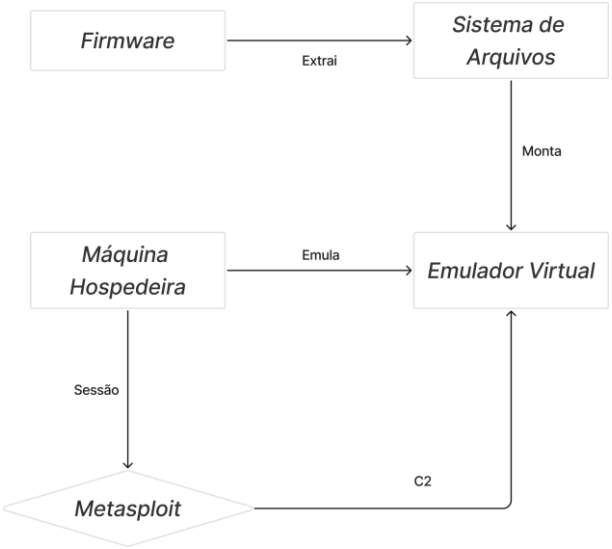

Figura 1: Esquema do sistema completo. Fonte: Os autores.

Para emular o roteador, utilizamos o *software* QEMU, que auxilia na virtualização de um sistema similar ao roteador real, replicando sua arquitetura e sistema operacional [9]. Identificar a arquitetura do *firmware*, que neste caso é do tipo MIPS, foi essencial para que tudo fosse configurado corretamente. Também foi necessário configurar a rede da máquina hospedeira adequadamente para que a comunicação entre o emulador virtual criado com o QEMU e a máquina hospedeira pudesse simular sistemas distintos de maneira eficaz.

Em seguida, o sistema de arquivos extraído foi transferido para o emulador virtual, onde o montamos novamente para tornar esses arquivos executáveis, como mostra a Figura 2.

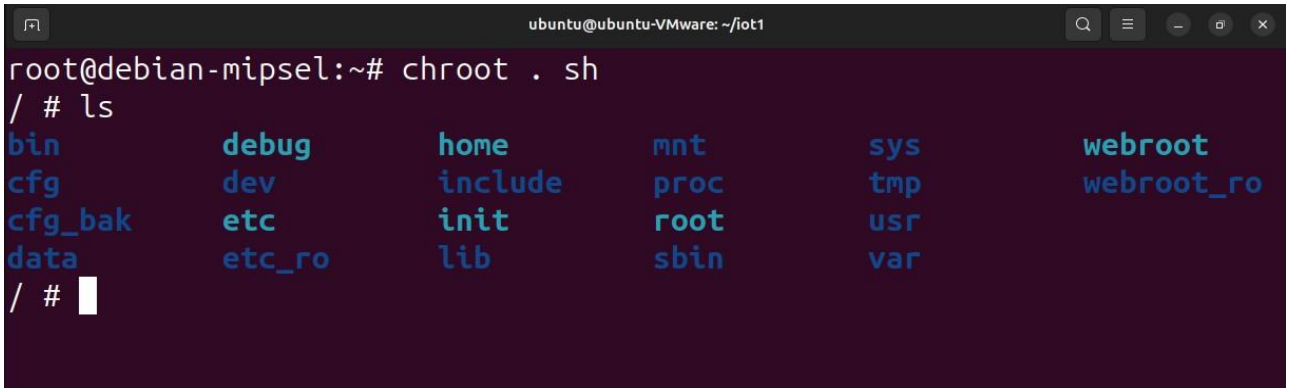

Figura 2: Emulador virtual com o sistema de arquivos do roteador montado. Fonte: Os autores.

Posteriormente, utilizamos o software *Metasploit* para criar um canal de comando e controle (C2), aplicando a técnica específica de uso de softwares para esta finalidade. O *Metasploit*, uma ferramenta amplamente utilizada para testes de penetração e exploração de vulnerabilidades, oferece várias funcionalidades, incluindo a capacidade de estabelecer um canal de comando e controle. Com isso, foi possível executar comandos no sistema alvo, facilitando a exploração de possíveis vulnerabilidades.

Para essa etapa do projeto, precisamos configurar um servidor para estabelecer uma conexão inicial de C2. Nesta etapa, a própria máquina local atuou como servidor, recebendo a conexão de retorno do emulador virtual que simula o roteador e enviando comandos para ele. Precisamos configurar um *listener*, que permanece aguardando por conexões de retorno (*reverse shell*) do dispositivo comprometido.

Em seguida, foi gerado um código a ser executado no sistema alvo, denominado *payload*. Através do próprio *Metasploit*, o *payload* foi enviado para iniciar uma sessão chamada *Meterpreter*. Essa sessão permite que o dispositivo comprometido, o emulador virtual criado com o QEMU, seja acessado a partir do terminal da máquina hospedeira, possibilitando o envio de comandos como se fosse um usuário do sistema. Assim, foi possível navegar pelo sistema de arquivos presente no sistema emulado, manipulando todos os arquivos contidos nele.

A partir desta etapa, iniciamos a exploração por vulnerabilidade no sistema por meio da *shell* interativa. Por se tratar de um roteador, bons pontos de partida incluem arquivos de servidores web, pois geralmente controlam funções como inicialização do dispositivo, autenticação de usuários, gerenciamento de configuração, entre outros. O arquivo escolhido com base no sistema mostrado nas figuras, foi o binário **httpd**, localizado no caminho `/bin/httpd`. Esse binário geralmente é responsável por executar o servidor web embutido no roteador.

Em seguida, copiamos o arquivo para a máquina hospedeira, que nos permitiu realizar uma análise mais profunda do código-fonte, utilizando a ferramenta Ghidra [10]. Essa ferramenta permitiu a descompilação e a análise manual detalhada do código, facilitando a identificação de potenciais falhas de segurança. Durante essa análise, procuramos por usos inadequados de funções de manipulação de *strings*, como 'strcpy, 'sscanf', 'gets', que são frequentemente suscetíveis a vulnerabilidades do tipo *buffer overflow*.

RESULTADOS E DISCUSSÃO:

Nesta pesquisa, realizamos um estudo sobre o uso de técnicas específicas de controle remoto do sistema combinadas com métodos de segurança ofensiva para a identificação de vulnerabilidades em sistemas-alvo, especificamente em roteadores.

Os resultados obtidos foram positivos, pois o método proposto mostrou algumas falhas que podem ser exploradas por atacantes, como demonstramos no parágrafo anterior. Assim, através da utilização de um servidor C2, conseguimos estabelecer uma conexão remota com o sistema alvo, demonstrando a viabilidade de acessar e manipular o dispositivo de forma semelhante aos softwares de acesso remoto descritos na técnica do *MITRE ATT&CK*. O *Metasploit*, uma ferramenta amplamente utilizada em segurança ofensiva, foi fundamental nesse processo. Esse software permitiu a execução de comandos e a exploração de vulnerabilidades no ambiente emulado.

BIBLIOGRAFIA

[1] R. Yu, X. Zhang and M. Zhang, **"Smart Home Security Analysis System Based on The Internet of Things,"** 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Nanchang, China, 2021, pp. 596-599, doi: 10.1109/ICBAIE52039.2021.9389849.

[2] S. J, A. E, N. Ramesh and V. Vijayakumar, **"Comparative Analysis of Firmware Security: A Proactive Paradigm for Enhancing Efficiency and Adaptability through Anomaly**

Detection," 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, 2023, pp. 842-847, doi: 10.1109/ICRTAC59277.2023.10480848.

[3] X. Feng, X. Zhu, Q. -L. Han, W. Zhou, S. Wen and Y. Xiang, **"Detecting Vulnerability on IoT Device Firmware: A Survey**," in *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 25-41, January 2023, doi: 10.1109/JAS.2022.105860.

[4] CrowdStrike. **Command and Control**. Disponível em: https://www.crowdstrike.com/cybersecurity-101/cyberattacks/command-and-control/. Acesso em: 20 jun. 2024.

[5] Rapid7. **Metasploit Documentation**. Disponível em: https://docs.metasploit.com/. Acesso em: 15 jul. 2024.

[6] MITRE**. MITRE ATT&CK: A Knowledge Base of Adversary Tactics and Techniques.** Disponível em**:** https://attack.mitre.org/. Acesso em**:** 1 ago. 2024.

[7] W. Chorfa, N. B. Youssef and A. Jemai, **"Threat Modeling with Mitre ATT&CK Framework Mapping for SD-IOT Security Assessment and Mitigations,"** 2023 IEEE Symposium on Computers and Communications (ISCC), Gammarth, Tunisia, 2023, pp. 1323-1326, doi: 10.1109/ISCC58397.2023.10217945.

[8] MITRE Corporation. **Remote Services**. Disponível em: https://attack.mitre.org/techniques/T1219/. Acesso em: 21 jun. 2024.

[9] M. Li, Y. Yu, Y. Zou and Y. Luo, **"Design of Dynamic Firmware Security Detection System Based on Virtual Simulation Technology,"** *2023 7th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE)*, Xi'an, China, 2023, pp. 218-221, doi: 10.1109/ICEMCE60359.2023.10491029.

[10] GHIDRA**. Software reverse engineering**. Disponível em: https://ghidra-sre.org/. Acesso em**:** 1 ago. 2024.